Display Techniques in Information-Rich Virtual Environments

Nicholas Fearing Polys

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

Doctor of Philosophy
In
Computer Science

Dr. Doug A. Bowman
Dr. Chris North
Dr. Scott McCrickard
Dr. Ken Livingston
Dr. Don Brutzman

June 2, 2006
Blacksburg, Virginia

Keywords:
Visual Design, Information Visualization, Virtual Environments,
Information-Rich Virtual Environments,
3D User Interfaces, Usability Testing and Evaluation

Display Techniques in Information-Rich Virtual Environments

Nicholas Fearing Polys

ABSTRACT

Across domains, researchers, engineers, and designers are faced with large volumes of data that are heterogeneous in nature - including spatial, abstract, and temporal information. There are numerous design and technical challenges when considering the unification, management, and presentation of these information types. Most research and applications have focused on display techniques for each of the information types individually, but much less in known about how to represent the relationships between information types. This research explores the perceptual and usability impacts of data representations and layout algorithms for the next-generation of integrated information spaces.

We propose Information-Rich Virtual Environments (IRVEs) as a solution to challenges of integrated information spaces. In this presentation, we will demonstrate the application requirements and foundational technology of IRVEs and articulate crucial tradeoffs in IRVE information design. We will present a design space and evaluation methodology to explore the usability effects of these tradeoffs. Experimental results will be presented for a series of empirical usability evaluations that increase our understanding of how these tradeoffs can be resolved to improve user performance. Finally, we interpret the results though the models of Information Theory and Human Information Processing to derive new conclusions regarding the role of perceptual cues in determining user performance in IRVEs. These lessons are posed as a set of design guidelines to aid developers of new IRVE interfaces and specifications.

Display Techniques in Information-Rich Virtual Environments (IRVEs)

Contents

APPENDICES

Figures, Tables, and Equations

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Acknowledgements

"The world of reality has its limits; the world of imagination is boundless."

- Jean-Jacques Rousseau

"Meaning is at once the mundane foundation of the mind's trivial pursuits and the inspiration for our most intimate, creative, and spiritual quests."

- Erik Davis

First, thanks are due to my advisors Doug Bowman and Christopher North for their continued support for and attention to this work. Without their encouragement and discrimination, many of the deeper questions addressed here may not have been posed. I am grateful for the opportunity to thrive on their powers of curiosity and to contribute to such a new and compelling research area.

For my committee, I am thankful for the particular inspiration of Scott McCrickard, waving the banner of Usability high on the ramparts and proclaiming the real-world demands for and application of our research. Thanks to Ken Livingston for continuing to inspire excellence in the quest to understand this earthly carriage and its driver. Last, and certainly not least, I would like to thank Don Brutzman for his enthusiasm and commitment to open standards and open source data and tools. His vision of networked, multi-dimensional information space sharable over the web has kept me thinking its all possible; notwithstanding the obvious naval metaphor, “A rising tide lifts all boats!”

Many people have contributed to the research and publications described here: Andrew Ray, Jian Chen, Lauren Shupp, Dustin Arendt, James Volpe, Ulmer Yilmaz, Vladimir Glina and Seonho Kim. I would like to thank my fellow members of the 3D Interaction Research Group at VT, who have helped with so many explorations, discussions, and insights over the years, especially: Chad Wingrave, Dheva Raja, Scott Preddy, and others. Also, my other fellow grad students who know how to laugh: Pardha Pyla, Beth Yost, Rob Capra, Jonathan Howarth, Robert Ball, Con Rodi, Kibum Kim, and Jamika Burge.

Thanks especially to the University Visualization and Animation Group (UVAG) at Virginia Tech for the use of their facilities and systems such as the CAVE and file systems. Particularly, Ron Kriz, Patrick Shinpaugh, and John Kelso for the use of their knowledge, good graces, and working software. Also, much of this work is the product of extensive design discussions with domain researchers such as the PathSim research team at The Virginia Bioinformatics Institute: Karen Duca, Reinhard Laubenbacher, Mike Shapiro, Kichol Lee, John McGee, Dustin Potter, Purvi Saraiya, and David Thorley-Lawson.

Thanks are due to Alan Hudson and Justin Couch from Yumetech and Bradley Vender from North Dakota State University and for their assistance in integrating the XJ3D browser as a Snap component. It is also important to recognize the entire X3D Specification and Source Working Groups, especially Dick Puk, Leonard Daly, Tony Parisi, Joe Williams, and Keith Victor. The process of technical specification dialogue has usually been professional and productive… if not fun in a twisted sort of way.

All of the academic and professional support would not have amounted to a hill of bits without the support of my family. They have always encouraged me to find my bliss. David, Daphne, Josh, Sam, Mackensie, Jesse, Killhours, Polysians, Mills, all, Thank you!

Finally, most thanks are due to my dearest friend, advocate, and wife Kat Mills.

1. Introduction

1.1 Motivation

Across a wide variety of domains, analysts and designers are faced with complex systems that include spatial objects, their attributes and their dynamic behaviors. In order to study and understand these systems, users require a unified environment to explore the complex relationships between their heterogeneous data types- ***integrated information spaces***. The problem of integrated information spaces arises from the fact that users must comprehend the nature and relationships within each data type individually as well as the relationships between the data types. In other words, they require optimized visualization configurations to achieve the most accurate and complete mental model of the data.

Virtual environments (VEs) excel at providing users a greater comprehension of spatial objects, their perceptual properties and their spatial relations. Perceptual information includes 3D spaces that represent physical or virtual objects and phenomena including geometry, lighting, colors, and textures. As users navigate within such a rich virtual environment, they may need access to the information about the world and objects in the space (such as name, function, attributes, etc.). Presenting this related information is the domain of Information Visualization, which is concerned with improving how users perceive, understand, and interact with visual representations of abstract information [Card, 1999]. This abstract (or symbolic) information could include text, links, numbers, graphical plots, and audio/video annotations [Bolter, 1995], [Bowman, 1999]. Both spatial and abstract information may change over time reflecting temporal aspects. Unfortunately, few systems allow users flexible exploration and examination of dynamic abstract information in conjunction with a dynamic VE.

This work examines what integrated visualization capabilities are necessary for users to gain a full understanding of complex relationships in their heterogeneous data, and to create advantageous research, design, and decision-support applications. Visual Analytics refers to this science of facilitating analytical reasoning through interactive visual interfaces [Thomas, 2006]. However it is still not clear what constitutes a 'good' or effective design in information-rich applications. The challenge is to provide a set of organized, multi-dimensional representations that aid users to quickly form accurate concepts about and mental models of the system they are studying.

This research program leverages methods from Virtual Environments (VEs), Information Visualization (InfoVis), and the Psychology of Perception to develop '*Information-Rich Virtual Environments*' (IRVEs) as a solution to the challenges of integrated information spaces. Information-Rich Virtual Environments start with realistic perceptual and spatial information and enhance it with abstract and temporal information. In this way, IRVEs provide a context for the methods of VEs and InfoVis to be combined and towards a unified interface for exploring space and information. Next generation digital tools must address this need for the integration and presentation of spatial, abstract, and temporal information, and the following scenarios exemplify the nature and requirements of integrated information spaces.

1.2 Problem Scenarios

The volume and variety of data facing computer users present new opportunities and challenges. The bottleneck is not in data collection, but in the lack of appropriate frameworks and tools for managing and presenting diverse knowledge to the analyst. In order to generate concepts, hypotheses and decisions from heterogeneous information types, users must be able to identify relations between the information types in an easy and meaningful way. The following scenarios illustrate the problem of integrated information spaces with examples from 4 different domains: architecture, aeronautical engineering, cheminformatics, and biological modeling and simulation.

1.2.1 Architecture

The design challenges of integrated information spaces are particularly relevant to Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). Take for example computational environments in building construction. Architects design and review complex plans for the construction of a building, such as a home. Good plans must take into account the spatial layout of the home as well as other information such as materials, costs, and schedule (Figure 1.1). In this domain, the perceptual and abstract information are tightly interrelated, and must be considered and understood together by the architect.

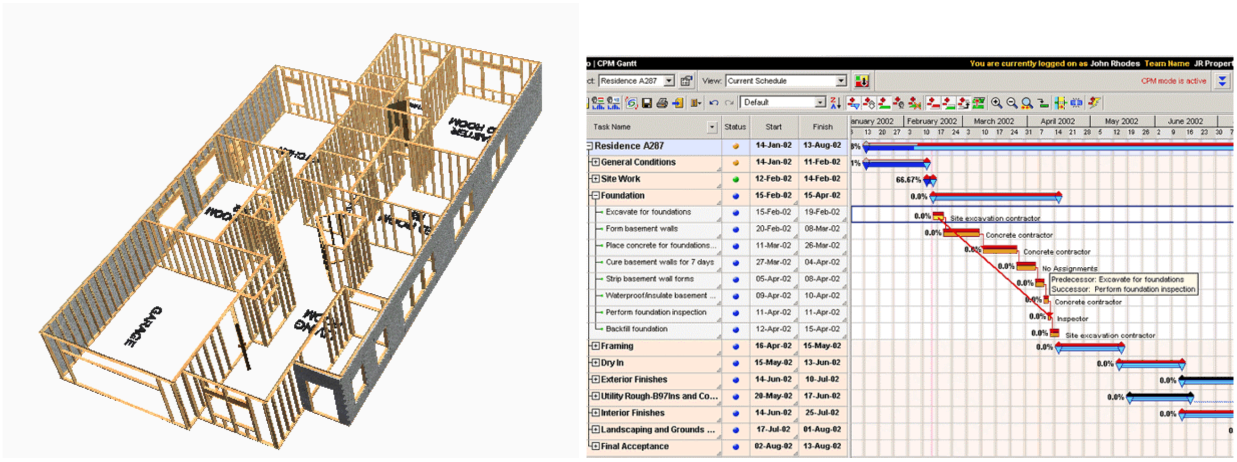


Figure 1.1: Perceptual (left) and abstract (right) information associated with a home's construction

For example, in a typical architectural design and review process, the project team may build a physical mockup and walk through it with blueprints, cost sheets, and timelines to discuss and note issues for its eventual construction. In this process, participants might say: "Let's go to the 3rd floor bedroom; how can we reduce the cost of this room? Which items are most costly? Are these items aesthetic or essential for load bearing support? Let's attach a note for the clients that this wall could be thinned to reduce cost." To complete such a task, participants need to cognitively integrate information from any number of separate representations, which is inefficient and error-prone. The current methods do not reflect the integrated nature of the data or support a unified interface for query, visualization and modification.

1.2.2 Aeronautical Engineering

Consider an engineer, '*E*', working on a complex aerospace craft such as the Space Shuttle. The craft is aging and the tolerances of the original parts are suspect. The engineer is tasked with designing a new gear assembly for the tailfin brake. *E* builds a 3D geometric model of the assembly in a CAD program, specifying its dimensions and the materials for its construction. *E* must then test the assembly design and its parts for physical tolerances using a meshing program and finite-element simulator. *E* must specify the kinetic forces of the assembly such as gears, locks, fulcri etc. After the simulator is run, *E* analyzes the results looking for weak points and insuring that all parts are within physical requirements (i.e. Figure 1.2).

E repeats this process a number of times to satisfy the specified design constraints of material's weights and stress limits. When he is satisfied, he saves the candidate model and simulation results into a database that represents the craft in its entirety. But *E* is not done; he must also confirm how his design affects the whole craft in flight and damage scenarios. *E*'s new part is then evaluated in the context of the other shuttle systems. Each scenario is linked to prioritized causal chains in a knowledgebase that infers mission consequences due to the particular flight/damage scenario.

How many applications did *E* use? How many times was he required to switch between a spatial view (e.g. a design application), a temporal view (e.g. a simulator application), and an abstract view (e.g. a information visualization application)? Was any model data lost, added, or made redundant between the applications? Was he able to view the impacts of his design changes simultaneously within one environment? On the same machine? This scenario illustrates the problem of integrated information spaces- current data models, applications, and presentations are fragmented and inefficient for users.

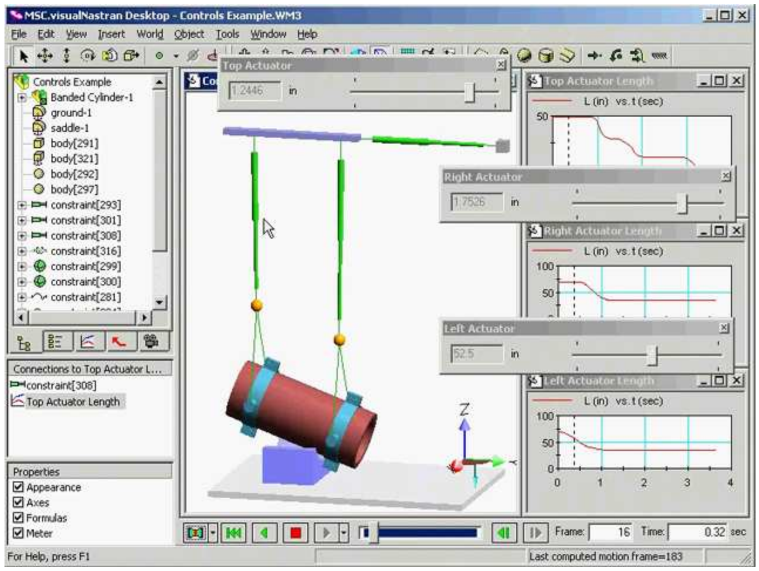


Figure 1.2: 4D engineering design tool (MSC software, MSC.visualNastran 4D)

1.2.3 Cheminformatics

Managing chemical information also exemplifies the combination of perceptual (spatial) and abstract information types. In chemistry, there is a growing body of data on molecular compounds: their physical structure and composition, their physico-chemical properties, and the properties of their constituent parts. For any molecular compound for example, there a number of kinds of information about it including: its physical atomic structure and element makeup, its atomic weight, water solubility, melting point, plus ultra-violet, infrared, and mass spectra. When considering information *about* a molecule or its parts, established visualization techniques can be used. For example, names and annotations can be displayed as textual tables, spectral data with multi-dimensional plots, and classifications with tree diagrams. In these abstract contexts, however, inherently spatial attributes of the data (such as the physical structural of atoms and bonds in a molecule) are difficult to understand.

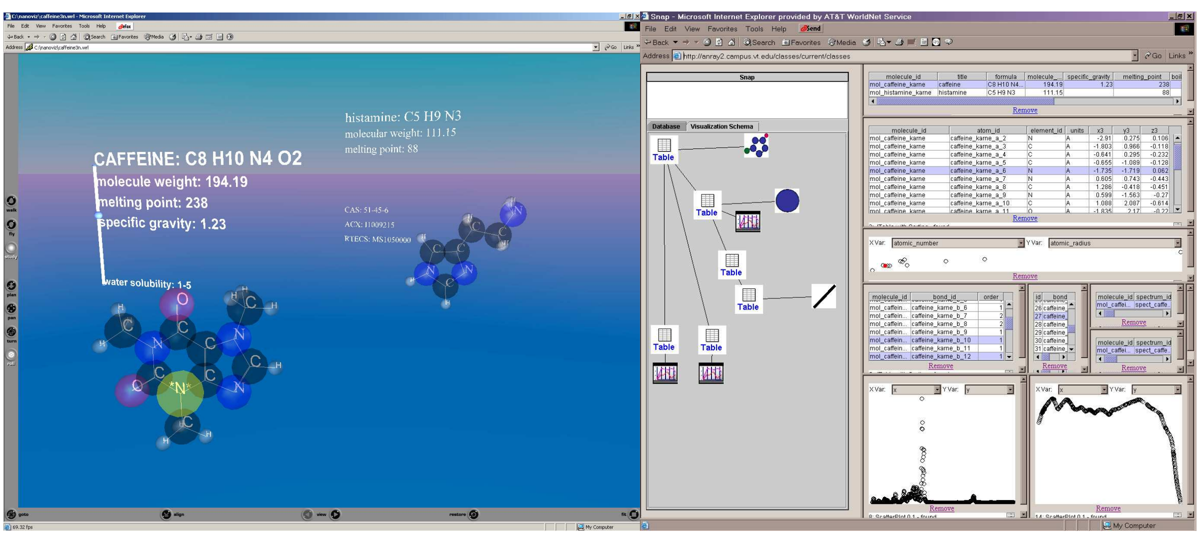


Figure 1.3: Linked multiple views of Chemical Markup Language (CML) data

In these applications, users frequently need to locate, relate, and understand this information across the abstract and perceptual visualizations (i.e. Figure 1.3). Users may need to index into the perceptual information through the abstract information and vice versa. For example, What are some structural or geometric features of *this* molecule? What are some characteristics of *this* molecule (boiling point, melting point, etc)? How about *this* atom (radius, weight, number)? What are the similarities and differences between *this* molecule and *that* molecule (size, molecular weight, and shape)? In order to optimally support such user queries, techniques for interactive visualization must be evaluated and improved.

1.2.4 Biological Modeling and Simulation

In the biomedical field, many researchers understand the spatial context of anatomy and its influence on real or simulated systems. Providing intuitive interfaces to multi-dimensional, spatially-registered time-series data is a great challenge. For example, future bioinformatics modeling and simulation applications require the researchers and students to visualize the results of one or more simulation runs, compare them to experimental results, specify the parameters for new simulations, navigate through multiple scales, and explore associations between spatial, abstract, and temporal data.

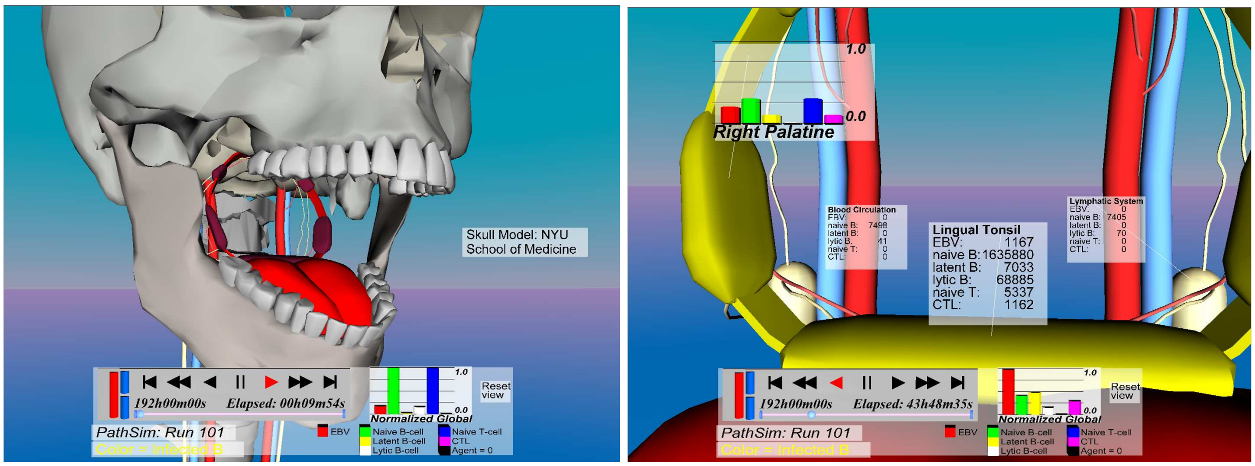


Figure 1.4: Embedded visualizations of an immunology simulation (PathSim)

Consider a user who decides to examine the effect of titer on the course of an infection. With the majority of interfaces, the user would type in the number of viruses (and perhaps their locations) at an input screen, wait for the simulation to finish, and finally extract the output from the simulation database as a table or spreadsheet. In an integrated information space, however, the user would deposit virions at the physical sites where infection occurs. After the simulation commences, the user might revisit the information space to view signaling events initiated by virus deposition at the molecular level. She would see the cytokines diffusing from the site of production into the surrounding tissue.

Later, the user might examine how fast the virus is spreading from the original site, killing cells, or recruiting immune cells to the vicinity (e.g. Figure 1.4). In examining the effect of titer, a group of researchers will likely run the simulation for a range of initial viral concentrations. They may then decide to view only the differences between several runs. This would be particularly interesting, as it would rapidly reveal whether titer is irrelevant or critical for development of clinical disease, rate of clearance, long-term prognosis, etc.

1.3 Challenges of Integrated Information Spaces

The scenarios described in the previous section highlight some crucial aspects of integrated information spaces. First, in many systems, dynamic properties are best modeled with their spatial and structural aspects - spatial relationships are important when structure, location, and function are related. Indeed, this is the case in many domains where physical systems are designed, simulated, or analyzed (i.e.[Subramaniam, 2003]). IRVEs aim to render clear views of such complex systems in order that users can develop an understanding of the relationships between the spatial representations and the abstract representations in the information environment.

Second, the environment cannot simply be a movie playing back – users require navigation among multiple information types. Abstract and temporal information must be retrieved and presented for appropriate regions and objects, and users are free to navigate to any viewing position. Ideally in IRVEs, all information should be accessible and coherent - without requiring the user to import or export data, or switch between applications for example.

Third, integrated information spaces need to effectively manage user attention and mental workload. For example, too much information or ambiguous layouts and cues can render the user lost, confused, and frustrated. IRVEs seek to reduce the cognitive distance between the user and the system and increase the information throughput between the user and the system.

There are many challenging problems for Computer Science in the development of integrated information spaces regardless of the domain. From a digital library perspective, there is the problem of the storage and retrieval of this data. Database, query and filter system architectures are well-established for abstract information, but spatial databases and heterogeneous representations have not achieved the same level of success.

From a network programming perspective, the variety and volume of data needs to be published or shared among collaborators (delivered over the web for example). Not only are network protocols and quality-of-service issues of interest, but also web standards, services, and shared ontologies. All of these concerns situate integrated information spaces in the ecology of the World Wide Web.

Finally, from an HCI perspective, the presentation of and interaction with massive heterogeneous data is the great challenge. Information display and layout as well as interaction design are not well understood for this class of problems. It is imperative that next-generation interfaces leverage the strengths of the human operator to create useful and economical tools for analysis and decision-making. This requires research into the how users perceive and process the variety of information in the space

While all of these challenges must be addressed to some degree, we believe the greatest progress toward next-generation information-rich applications will come from systematic research and development of IRVEs. There are few guidelines for designers to follow when considering the usability impact of their visualization and interaction choices for the ‘last mile’. Ideally, we want to understand how to leverage the human’s abilities for pattern-recognition, creative reasoning, and insight. Through research into IRVE display techniques (layout algorithms), we hope to answer this need by providing design taxonomies, prototypes, and empirical evidence supporting high-throughput, low-workload visual configurations.

1.4 Problem Statement

At the intersection of virtual environments and information visualization, interface designers aim to supply users with user with relevant information at a minimum perceptual, cognitive, and execution load. In order to “amplify cognition” as Card, Mackinlay, and Shneiderman [Card, 1999] suggest (pg. 7), designers are motivated to efficiently employ human perception and cognition in order to economically design and communicate perceptual substrates for accurate interpretation and use. Many useful analytic and visualization tools have been developed to enable detailed analysis of various data types, and these tools are important. However, in order to solve complex problems of design, engineering, and scientific research, better tools for overview, analysis, and synthesis of heterogeneous datatypes are required.

The canonical virtual environment (VE) application is the walkthrough – a static three-dimensional world made up of geometry, colors, textures, and lighting. Walkthroughs contain purely *perceptual* information – that is, they match the appearance of a physical, spatial environment as closely as possible. Many other VE applications, even those that are dynamic and present multi-sensory output, also have this property. In traditional information visualization (InfoVis) applications, on the other hand, the environment contains purely *abstract* information that has been mapped to perceptual dimensions such as position, shape, size, or color. In this research, we intend to develop a new research area at the intersection of traditional VEs and traditional InfoVis. *Information-rich virtual environments* (IRVEs) start with realistic perceptual information and enhance it with related abstract information.

We propose Information-Rich Virtual Environments (IRVEs), the combination of information visualization and virtual environments, as a solution to the requirements of integrated information spaces. An IRVE allows navigation within a perceptual environment that is enhanced with the display of and interaction with related abstract information. The methods of VEs and InfoVis are combined in a concerted way to enable a unified approach to exploring the space and its manifest information. Enhancing information to virtual environments can include: nominal, categorical, ordinal, and quantitative attributes, time-series data, hyperlinks, or audio or video resources. Some of this information may already be effectively visualized with established techniques and applications. Information visualizations for example, present abstract information using a perceptual (usually visual) form. In contrast however, IRVEs connect abstract information with a realistic 3D space. In this way, abstract and perceptual information are integrated in a single environment [Bolter, 1995].

Some specific types of IRVEs have been explored. For example, scientific visualizations that display abstract data attributes as a function of a 3D space using color encoding or glyphs can be classified as IRVEs. However, IRVE applications are a superset of scientific visualization applications and a much broader framework for the general case of IRVEs is needed. This framework must encompass the broader spectrum of abstract information types and structures, as well as different types of relationships between the perceptual and abstract information. It must integrate the display and interaction techniques of InfoVis with those of VEs, while emphasizing the fidelity and realism of the perceptual information.

1.5 Research Goals

The basic concept of IRVEs is not new; indeed many VE and visualization applications have included related abstract information. However, there has been a lack of precise definition, systematic research, and development tools for IRVEs. Our research generalizes from prior work to define this research area and produce principles and tools for the development of effective and usable IRVEs. Specifically, a systematic program is needed to:

- Understand how different IRVE layout algorithms (display techniques) effect user performance, and
- Develop a methodology to assess, design, and deliver appropriate displays.

1.6 Approach

For current and future Information-Rich Virtual Environment applications, it is imperative that developers inform their design through sound Human Computer Interaction (HCI) practice. This research will leverage work from cognitive psychology, information psychophysics, and visualization systems to provide guidelines of design practice for desktop and large screen IRVE systems. Through user-centered research, we aim to examine how the respective visualization techniques for perceptual and abstract information can be combined and balanced to enable stronger mental associations between physical and abstract information while preserving the models of each type of information.

It follows that now we can enumerate the goals of the research proposed here:

a) **Define a theoretical framework for Information-Rich Virtual Environments (IRVEs) as the solution to the problem of integrated information spaces**

b) **Enumerate the design space for IRVE tasks and display techniques**

c) **Describe IRVE display configurations in an XML DTD and Schema**

d) **Prototype information-rich application interfaces to identify problems and generate hypotheses regarding optimal IRVE information designs**

e) **Identify tradeoffs and guidelines for the IRVE display design space using prototype interfaces, usability evaluations, and metrics for individual cognitive differences**

The goal is to understand what makes effective information display techniques for IRVEs. We propose to accomplish this through design and evaluation of IRVE display techniques. However, design and evaluation activities should not be done in an ad hoc fashion; rather, they should be based on **a theoretical framework**. We need to understand precisely what IRVEs are, what tasks users will want to perform in them, what techniques are possible for information display and interaction, and how those techniques might affect user task performance and usability. Thus, our first research objective is to specify the contents and boundaries of this research area.

Our theoretical framework includes a notion of the design space for IRVE information display techniques. In other words, what are the possible ways that abstract information can be included within a VE? This includes issues such as the type of abstract information (e.g. text, audio, graphics), where the information should appear (e.g. in a Heads-Up Display, in a spatial location attached to an object), how links among information should be represented, and so on.

The **IRVE information design** enterprise focuses on the representation and layout of enhancing abstract information in perceptually realistic virtual environments. There are many possible representations for a given dataset, such as a table, a plot or graph, a text description, or an equation. Placing this information (layout) within the 3D VE is not a trivial issue - there are a number of possibilities as to how abstract information is related to the perceptual information in an IRVE. The displayed information needs to be visible for the user; it should not occlude important objects in the world; and it should not interfere with other displayed information. We will also develop techniques for managing the layout of information within a VE.

The third research goal is to use the dimensions of the information design space to define a configuration language for IRVE displays. There are two reasons for this. First, the World Wide Web Consortium's [W3C] codification of the meta-language XML has opened new and powerful opportunities for information visualization, as a host of structured data can now be transformed and/or repurposed for multiple presentation formats and interaction venues. XML is a textual format for the interchange of structured data between applications. The great advantage of XML is that it provides a structured data representation built for the purpose of separating content from presentation. This allows the advantage of manipulating and transforming content independently of its display. It also dramatically reduces development and maintenance costs by allowing easy integration of legacy data to a **single data representation which can be presented in multiple contexts or forms** depending on the needs of the viewer (a.k.a. the client). Data thus becomes 'portable' and different formats such as X3D and VRML may be delivered and presented (styled) according to the application's needs [Polys, 2003].

In addition, a configuration language for IRVE displays will leverage another important aspect of XML - the tools that it provides as in the DTD and Schema. The DTD, or Document Type Definition, defines 'valid' or 'legal' document structure according to the syntax and hierarchy of its elements. The Schema specifies data types and allowable expressions for elements' content and attributes; they describe the document's semantics. Using any combination of these, **high-level markup** tags may be defined by application developers and integration managers. This allows customized and compliant content to be built for use by authors and domain specialists. These tags could describe prototyped user-interface elements, information objects, interaction behaviors, and display venues. Using the linguistic structure of XML, developers can describe valid display interfaces for their application by using a DTD and Schema. The content model can then be published as a validating resource over the web, and even be standardized amongst the community. An XML language will be used to describe IRVE display components and instantiate them in the runtime of an application or testbed.

In order to expand the theoretical foundation of IRVEs, we will prototype a number of IRVE interfaces. **User Interface Prototypes** are tools employed to discover and examine usability issues [Wasserman, 1982]. Prototypes can be used to discover and refine user requirements, as well as develop and test design ideas in the usability engineering process [Rosson, 2002]. Through our applied work with the Virginia Bioinformatics Institute, graduate projects and testbed evaluations, we will explore the design space- iteratively developing IRVE interface components for evaluation and testing.

Finally, **usability evaluations and empirical tests** will be conducted to provide data on the performance of various display techniques for different kinds of data sets. IRVE evaluation requires an integrated approach to display and interaction, and there are many variables. Our test designs focus on the graphical integration of perceptual and abstract information in commodity computing systems. **Through our research, we examine two critical IRVE activities: Search and Comparison.**

In order to enumerate design guidelines that resolve the IRVE information design tradeoffs, we conduct a series of controlled experiments with our IRVE prototype testbed. This testbed serves as the control and condition environment to test various design features along the dimensions of the IRVE design space. We consider time and correctness in task completion, as well as user satisfaction, as dependent variables.

Empirical usability evaluations will be informative because performance for information perception and interpretation is required for the acquisition of conceptual and procedural knowledge. We aim to reduce the user's mental workload by leveraging pre-attentive perceptual processes in information display. The throughput between environment and user can be measured through objective performance measures as well as subjective measures. Qualitative subjective measures are important to understand how users react to an IRVE interface. In addition, experimental outcomes may be influenced by a participant's spatial and cognitive abilities, therefore we will assess these using standard protocols. The results of these experiments will allow us to identify advantageous design features, guidelines, and hopefully design patterns for IRVE display techniques.

1.7 Research Questions and Hypotheses

We believe that Information-Rich Virtual Environments (IRVEs), by enhancing perceptual information with abstract information, enable a semantic directness that can lead users to more accurate mental models of the data they are interpreting. We have stated that a major problem of integrated information spaces is for users to comprehend perceptual and abstract information both together and separately. In order for IRVEs to meet this requirement, we need to understand how perceptual and abstract data can be combined, what makes the combinations effective, what makes them usable, and how users think and act when using them. We hope these questions will lead us to discover a set of principles and guidelines for IRVE designers and developers.

With a theoretical framework and a set of tools for IRVEs in place, we will address specific issues of design. Recently, we have enumerated a host of research questions posed by IRVEs [Bowman, 2003a]. This research addresses one specifically related to information design and supporting users in identifying patterns and trends in their heterogeneous data sets. We are especially interested in the question of layout algorithms, display techniques that portray relationships between spatial, abstract, and temporal information.

Specifically:

Where and How should enhancing abstract information be displayed relative to its perceptual referent so that the respective information can be understood together and separately?

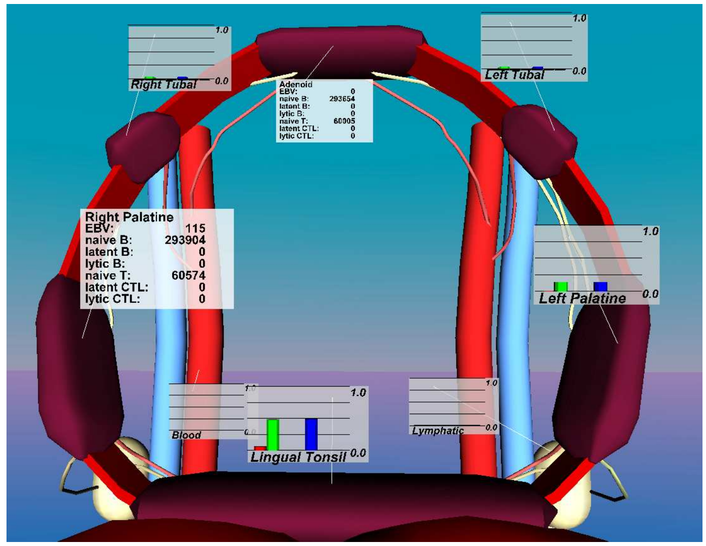
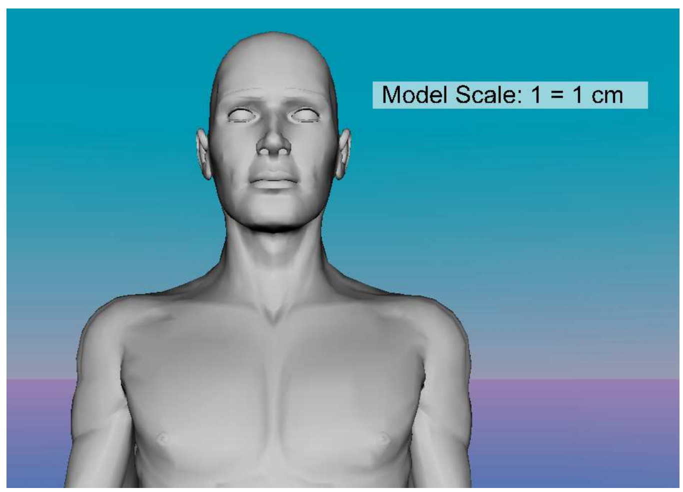
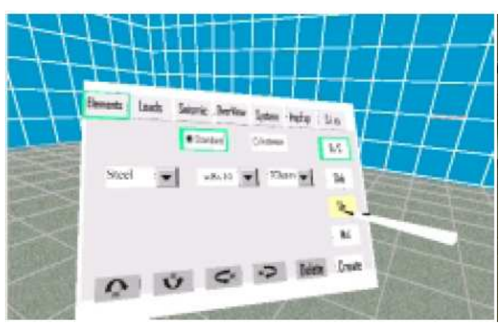
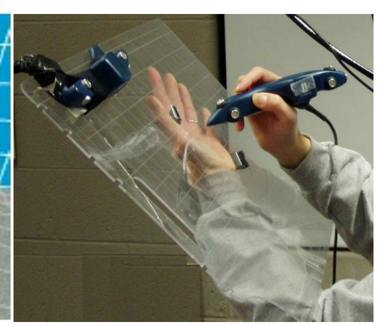
For IRVEs, we need to design information displays that enable accurate mental models of the relation between abstract and perceptual information while at the same time maintaining accurate models of each individual information type. IRVEs present a number of design challenges (Section 3.2) and we have developed a description of the dimensions of the space.

We originally proposed the IRVE design space in [Bowman, 2003a]. Here it has been revised to more fully capture the nuances regarding: in which coordinate system the abstract information is **located** and how it is graphically **associated** to the perceptual information (Table 1.1). There is a large range of design possibilities in this space and the capabilities of new database models and compositing techniques must be assessed from a user-centered design perspective. Therefore, we focus our investigation on these dimensions to understand the relative strengths of the combinations of Depth and Gestalt cues for IRVE information displays.

First, the Layout space dimension of IRVEs refers to the coordinate space in which the abstract information is located. We have adapted these distinctions from Augmented Reality (AR) to account for the variety of IRVEs and coordinate systems for abstract information layout (**Section 2.1.3**). Depending on the display technique used, annotations in these Layout spaces may provide a variety of Depth cues consistent with their referent in the virtual space. Display techniques may be implemented by one or many federated visualization applications. Object space represents one end of this dimension, where abstract information is represented with strong depth cues such as occlusion, motion parallax, relative size, and linear perspective. Display space is the other end of this dimension, where abstract information does not support strong depth cues such as occlusion, and visual interference between annotations and objects is minimal. Examples of these layout spaces are shown in Table 1.2.

Association / Layout Space	Common Region	Proximity	Connectedness	Common Fate	Similarity
Object	x	x	x	x	x
World	x	x	x	x	x
User	x	x	x	x	x
Viewport	x	x	x	x	x
Display	x	x	x	x	x

Table 1.1: Orthogonal Layout space and Association dimensions in IRVE design

Layout Space	***Example***
Object space Object space is relative to an object's location in the environment. Image from PathSim [Polys, 2004d]	
World space World space is relative to an area, region, or location in the environment. Image from PathSim [Polys, 2004d]	
User space User space is relative to the user's location but not their viewing angle. Image of tablet UI from Virtual SAP [Bowman, 2003b].	 

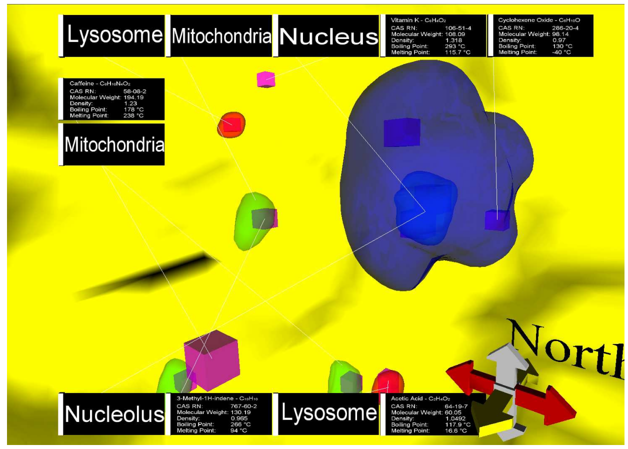
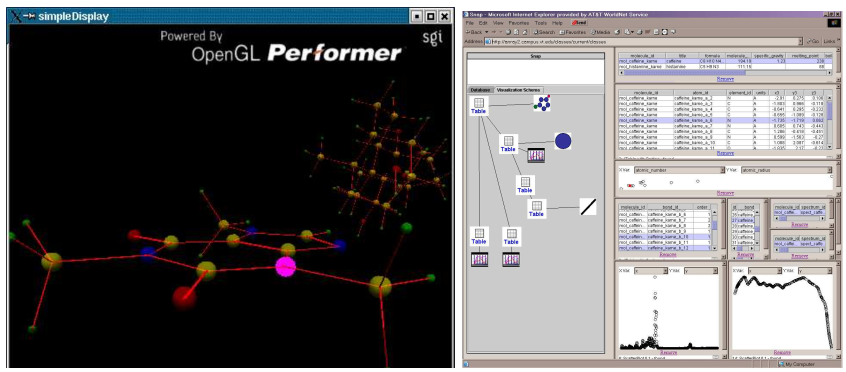
Viewport space Viewport space is the image plane where HUDs or overlays may be located. Viewport space is relative to the user's location *and* their viewing angle. Image from HUD condition from [Polys, 2005c]	
Display space Display layout space where abstract visualizations are located outside the rendered view of the virtual environment in some additional screen area. Image from Snap2Diverse [Polys, 2004a]	

Table 1.2: Examples of the Layout space dimension in IRVEs

Figure 1.5: Gestalt principles in the Association dimension

Second, the Association dimension refers to the visual configuration by which abstract information is related to spatial information. Here we invoke the Gestalt principles of 2D perception to describe how abstract information can be associated to perceptual information in the visual field of an IRVE (Figure 1.5). Common Region, Connectedness, Similarity all refer to visual properties of an image frame configuration; the common fate principle requires a temporal dimension to the environment such as an animation or feedback from a user interface event. How do the Gestalt cues effect performance in IRVEs where the visual field is dynamic and where Depth cues are also important?

For the combination of abstract and perceptual visualizations in IRVEs, association should be maximized and interference (such as occlusion and crowding) should be minimized. Based on our design matrix in Table 1.1, the guiding research question can be re-phrased as:

“What are the best ways to manage layout space and visual associations so that perceptual and abstract information can be understood together and separately?”

Choosing the best display techniques to fill this requirement depends on the nature of the data set and the nature of the user tasks - different user tasks may require different display techniques. In addition, IRVE designers have a range of display platforms and resolutions to contend with and this is certainly a primary consideration in invoking particular information design guidelines. In order to address the most common platform across industries, this research will focus on typical resolutions for commodity desktop and projection systems.

The general hypothesis behind this research is that human perception and cognition impose a structure on the IRVE information design space that can be discovered and leveraged for optimal design. This is a branch of Human Factors and Human Computer Interaction research known as Cognitive Ergonomics [Smith-Jackson, 2005]. Given an information-rich data set and a set of user tasks, we intend to substantiate guidelines for the appropriate mappings of data to display. Through the approach and method described below, we will address our guiding research question in two parts:

1. To understand the *tradeoffs* ***between*** *layout spaces per task*. We will examine Object space versus Viewport and Display space.
2. To understand the *tradeoffs* ***within*** *layout spaces per task*. We will examine Object and Viewport space.

1.8 Significance

‘Information-Rich Virtual Environments’ (IRVEs) are a strategic research area addressing the problem of integrated information spaces- virtual environments that enhance their perceptual (spatial) cues with abstract information about the environment and its constituent objects. In IRVEs, the perceptual and spatial data of the virtual environment is integrated with abstract data of information visualizations, giving users simultaneous visual and interactive access to the variety of data types. The IRVEs agenda provides a venue for the methods of Virtual Environments, Information Visualization, Cognitive Psychology, and Information Architecture to be combined in complementary ways. The lessons learned provide us a deeper understanding of the great potentials in rich media and the human mind.

IRVEs provide exciting opportunities for extending the use of VEs for more complex, information-demanding tasks in many domains. Upon completion of this research we will have generalized prior research and provided a theoretical framework for systematic visualization research in the field of IRVEs. We will have also presented a set of tools for the development and evaluation of IRVEs. The results of these evaluations are design guidelines for IRVE information displays and undoubtedly more research questions for this burgeoning area.

By exploring across and within dimensions of the IRVE design space, we hope to understand how various display techniques support efficient perceptual access to and cognitive integration of different data types. IRVE display techniques can provide a range of perceptual cues that bind abstract information with spatial information and there are tradeoffs to these cues. As such, this research also contributes to the models of Human Information Processing by demonstrating the relative powers and interactions of Depth and Gestalt perceptual cues for common tasks and task-information mappings. Our results have a direct impact on the future design of Virtual Environments, Information Visualization, and Augmented and Mixed Reality interfaces.

This research program will leverage standard formats for interactive 3D media such as Extensible 3D (X3D) and Virtual Reality Modeling Language (VRML). X3D is the successor to VRML as a full-featured ISO international standard for creating and delivering real-time 3D content. Through our prior work in the development of the X3D standard, we have the opportunity to impact future versions of this standard. The scenegraph behaviors that comprise IRVE interfaces require a number of custom objects that may be candidates for standardization as X3D Components. Briefly, these include the annotation and meta-information components, rendering and overlay components, as well as improving text support and level-of-detail components. We hope to improve the standard through the results of our research, and work with the specification groups of the Web3D Consortium to specify this functionality for reusable, native implementations.

In the remainder of this document, we detail related work, definitions, and our approach, experimental method and results. This work provides a deeper understanding of human performance in IRVEs information displays by grounding the visual and data modeling issues in cognitive ergonomics. Ultimately, we hope that this research lays the foundation for an entirely new set of powerful and easy-to-use VE applications.

1.9 Summary of This Work

Chapter 2 provides a comprehensive and multi-disciplinary literature review on issues relevant for the study of IRVEs. It begins with Psychophysics and fundamental theories of Signal Detection, Information Theory, and the models of Attention, and Perception. We continue to examine the phenomena of information integration and comprehension through the human user's sensory apparatus via the existing work of Information Visualization, Multimedia, and Virtual Environments. In addition, we connect the problem of integrated information spaces to the literature in Cognitive Psychology and Augmented Reality. Finally we discuss how existing methods of Usability Engineering and User-Centered Design are applied and enriched through this research into IRVEs.

In Chapter 3 we describe our proposed solution to the problem of integrated information spaces. We detail the definition of IRVEs and the variety of tasks and task mappings that are canonical for IRVEs. In this chapter we also detail the design dimensions of IRVEs and the display components architecture we have designed and implemented to meet the requirements of integrated information spaces.

Chapter 4 describes the technology and publication paradigms relevant to IRVEs. It presents a detailed treatment Information Architectures that support the transformation and delivery of data to IRVE runtimes. Chapter 5 details a case-study of applying IRVE techniques to a bioinformatics research problem. Spatially-registered timeseries data is especially challenging to systems biologists and immunologists who study infection through clinical and laboratory research as well as simulation. We use our work on the PathSim Visualizer to better understand IRVE requirements in the real world and to prototype IRVE display techniques. System and user interface features are described and related to current visualization and analysis tools.

Chapters 6 and 7 cover the range of IRVE display techniques and evaluations performed in this research program. We ran analytic and empirical usability studies with human subjects to understand design tradeoffs and user performance for IRVEs. Chapter 5 describes our comparative evaluations of display techniques between Layout Spaces (Object vs. Viewport and Object vs. Display). Chapter 6 describes our examination of display techniques within two common layout spaces: Object and Viewport. The results of these evaluations provide insight into the performance tradeoffs of IRVE techniques and illustrate the task-specificity of certain techniques.

Chapter 8 summarizes this IRVE research program and our derived design guidelines for IRVE information displays. In this final chapter, we relate our findings back to models of perception as well as the requirements of supporting information architectures. We conclude with general lessons learned, and goals and speculations for future work.

2. Review of the Literature

2.1 From Sensation to Perception

2.1.1 Signals, Channels and Cues

One important Human Information Processing model for perception is called 'Signal Detection Theory'. This model concerns the detection of a signal in 'noisy' conditions (i.e. under uncertainty) over some channel, in this case sensory modalities (i.e. visual or auditory channels). Whether the signal sensation is perceived depends on two factors: beta and d' ("d-prime"). beta is a term that describes the subjective level of certainty in the human operator. d' refers to the sensitivity of the sensory system. Both of these factors may be manipulated by design. For example, a visualization or display technology might change the salience of a stimuli to overcome a sensitivity threshold or a decision support tool might guide the operator to explicitly consider certain information or procedures in order to reduce a risky criterion or bias [Wickens, 2000]. The question of receiver bias and sensitivity to various visual cues is the subject of this research.

2.1.2 Attention and Pre-Attention

When humans acquire a skill, they are typically learning to perform a complex behavior or set of behaviors. As they learn the skill, some aspects of performance can be automatized to require less cognitive and attentional resources. Automatized processes reduce cogntive overhead as they do not involve conscious control or attentional resources; as such, they can usually be performed in parallel with other tasks and are usually obligatory ([Eysenck, 2000], pg 141). [Treisman, 1988] noted that there may be extensive processing of unattended sources of information and articulated a robust theory called 'Pre-attentive Processing Theory'.

The efficiency advantages of automatic processes make them a desirable target for certain aspects of training. However, some aspects of complex task performance should not be automatized in order to guarantee sensitivity and flexibility to novel situations. These aspects of performance should remain controlled and receive proper attentional resources. In contrast to automatized processes, controlled processes can be characterized as declarative, serial, and explicitly managed by trainable conscious, or 'top-down', strategies [Gopher, 1996]

Management of attentional resources can be determined by the environment and by user strategy. The striking effects of the contrast between automatic attentional processes and those guided by top-down or instructional processes is clear in Simons work on attentional capture and inattentional blindness [Simmons, 2000]. When instructed or given one kind of kind of stimuli, another unexpected type may go unnoticed. In retrospect or under different instruction, the same unexpected stimuli is obvious. The perceptual system can be high-jacked by top-down control of attention, sometimes resulting in the phenomena described as 'attentional blink' [Rensink, 2000]. The Human perceptual system can also be primed for detection of spatial and linguistic stimuli non-consciously (at a pre-semantic level) [Tulving, 1990].

Gopher (1996) has also examined the role of a control system in attentional performance for variable priorities and variable degrees of theoretical understanding of the system in dealing with 'mishaps' in the system. He found that executive attentional control is a strategic behavior and that users can increase performance given a proper conceptual model. As Green & Bavelier have shown [Green, 2003], a minimal training period of 10 X 1 hour sessions on first person, 3D-action video games (e.g. Medal of Honor) can significantly transfer and increase user performance in attentional enumeration as measured by tests of "Useful Field Of View" (UFOV) and attentional blink. Interestingly, this effect was not observed in subjects trained with Tetris, an exocentric and 2D spatial puzzle game.

Ekstrom et al's set of factor-referenced cognitive tests [Ekstrom, 1976] builds on an extensive effort to describe and measure fundamental aptitudes [Carroll, 1993]. It is an open question if these established paper tests would be predictive of IRVE performance or preference. Two test are most obviously related to IRVEs and we include them in our experimental data collection: Perceptual Speed (Number Comparison) and Closure Flexibility (Hidden Patterns); respectively:

- The Numbers Comparison test is a test of perceptual (P) “Speed in comparing figures or symbols, scanning to find figures or symbols, or carrying out other very simple tasks involving visual perception”. In the test, users are asked to inspect pairs of multi-digit numbers and indicate if they are the same or different.

- The Hidden Patterns is a test of the Closure Flexibility factor (CF), which is “The ability to hold a given precept or configuration in mind so as to disembed it from other well defined perceptual material.”. This test is to mark if a single given configuration is embedded in a given geometrical pattern.

2.2 From Perception to Information

2.2.1 Information Visualization

Card, Mackinlay, and Schneiderman have defined Information Visualization as “The use of computer-supported, interactive, visual representations of abstract data to amplify cognition” ([Card, 1999], pg. 7). This definition provides us with a clear starting point to describe visualization techniques for X3D as it distinguishes abstract data from other types of data that directly describe physical reality or are inherently spatial (i.e. anatomy or molecular structure). Abstract data includes things like financial reports, collections of documents, or web traffic records. Abstract data does not have obvious spatial mappings or visible forms and thus the challenge is to determine effective visual representations and interaction schemes for human analysis, decision-making, and discovery.

The nature of visual perception is obviously a crucial factor in the design of effective graphics. The challenge is to understand human perceptual dimensions and map data to its display in order that dependent variables can be instantly perceived and processed pre-consciously and in parallel [Friedhoff, 2000]. Such properties of the visual system have been described (ie sensitivity to texture, color, motion, depth) and graphical presentation models have been formulated to exploit these properties, such as pre-attentive processing ([Pickett, 1995], [Treisman, 1988] and visual cues and perception [Keller, 1993].

Primary factors in visualization design concern both the data (its dimensionality, type, scale, range, and attributes of interest) and human factors (the user’s purpose and expertise). Different data types and tasks require different representation and interaction techniques. How users construct knowledge about what a graphic ‘means’ is also of inherent interest to visualization applications. For users to understand and interpret images, higher-level cognitive processes are usually needed. A number of authors have enumerated design strategies and representation parameters for rendering signifieds in graphics ([Bertin, 1983], [Tufte, 1983; Tufte, 1990]) and there are effects from both the kind of data and the kind of task [Shneiderman, 1996].

Ben Shneiderman [1996] outlined a task and data type taxonomy for interactive information visualization, which is also useful for our description of techniques for IRVEs. Top-level tasks for navigating and comprehending abstract information are enumerated as: Overview, Zoom, Filter, Detail-on-demand, Relate, History, and Extract. Overview refers to a top-level or global view of the information space. Zoom, Filter, and Details-on-demand refer to the capability to ‘drill down’ to items of interest and inspect more details (of their attributes). History refers to the ‘undo’ capability (ie returning to a previous state or view) and Extract is visualizing sub-sets of the data. Enumerated data types are: 1-dimensional, 2-dimensional, 3-dimensional, Multidimensional, Temporal, Tree, and Network. Since each of these can be part of an IRVE, we will refer to these distinctions throughout the remainder of the proposal.

Card, Mackinlay, and Scheiderman [Card, 1999] have examined a variety of graphical forms and critically compared visual cues in: scatter-plot, cone-trees, hyperbolic trees, tree maps, point-of-interest, and perspective wall renderings. As we shall see, their work is important since any of these 2D visualizations may be embedded inside, or manifested as, a virtual environment. Interactive computer graphics present another level of complication for designers to convey meaning as they are responsive, dynamic and may take diverse forms. There are challenges both on the input medium for the user and the action medium for the user. These are known as the Gulf of Evaluation and the Gulf of Execution respectively [Norman, 1986]. Typically in the literature, visualizations are described and categorized per-user-task such as exploring, finding, comparing, and recognizing (patterns). These tasks are common in interactive 3D worlds as well. Information objects may need to be depicted with affordances for such actions.

Visual Markers

General types of data can be described as: Quantitative (numerical), Ordinal, and Nominal (or categorical). Visualization design requires the mapping of data attributes to 'visual markers' (the graphical representations of those attributes). Information mappings to visualizations must be computable (they must be able to be generated by a computer), and they must be comprehensible by the user (the user must understand the rules that govern the mapping in order to interpret the visualization). The employment of various visual markers can be defined by the visualization designer or defined by the user. Tools such as Spotfire [Ahlberg, 1995] and Snap [North, 2000] are good examples of this interactive user control over the display process. In addition, a set of 'modes' of interaction have been proposed for exploratory data visualizations which attempt to account for user feedback and control in a runtime display [Hibbard, 1995]. Table 2.1 summarizes the ordering of visual markers by accuracy for the general data types. These rankings lay a foundation for identifying parameters that increase the information bandwidth between visual stimuli and user.

Data Type	Quantitative	Ordinal	Nominal
Graphical Representation	position length angle / slope area volume color / density [Cleveland and McGill, 1984]	position density color texture connection containment length angle slope area volume [Mackinlay, 1986]	position color texture connection containment density shape length angle slope area volume [Mackinlay, 1986]

Table 2.1: Accuracy rankings for visual markers by general data type

Card, Mackinlay, and Schneiderman [Card, 1999] as well as Hibbard et al [Hibbard, 1995] have described a reference model for mapping data to visual forms that we can apply to our discussion. Beginning with raw data, which may be highly dimensional and heterogeneous, data transformations are applied to produce a 'Data Table' that encapsulates the records and attributes of interest. The data table also includes metadata which describes the respective axis of the data values. Visual mappings like those shown in Table 2.1 are then applied to the data table to produce the visual structures of the visualization. The final transformation stage involves defining the user views and navigational mechanisms to these visual structures. As the user interactively explores the data, this process is repeated. Ideally, the user has interactive control (feedback) on any step in the process (the data transformation, visual mappings, and view transformations); see Figure 2.1.

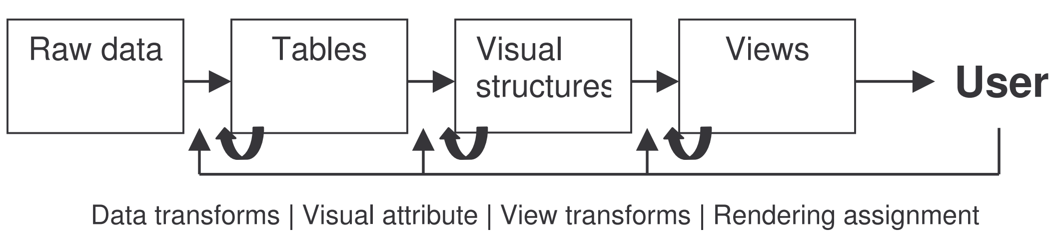


Figure 2.1: Processing in a typical visualization pipeline (from Card et al, 1999)

Multiple Views

A growing body of work is leveraging object-oriented software design to provide users with multiple linked views or renderings of data. [Roberts, 1999] and [Boukhelifa, 2003] have described additional models for coordinating multiple views for exploratory visualization including 2D and 3D views. In Roberts' Waltz system for example, multiform 2D and 3D visualizations of data are displayed and coordinated as users explore sets and subsets of the data.

[North, 2001] has also described a taxonomy of tightly-coupled views which reviews systems and experimental evidence of advantages including significant speed up on overview+detail tasks. These visualizations are coordinated by simple events such as*:* 1. Selecting items <-> Selecting items, 2. Navigating views <-> Navigating views, and 3. Selecting items <-> Navigating views, for example. The 'Visualization Schema' approach allows users to build their own coordinated visualizations [North, 2002]. We would like to extend this concept to virtual environment design so that embedded information and interfaces inside the environment can be customized and composed in a structured way.

Baldonado, Woodruff, and Kuchinsky [Baldonado, 2000] have proposed guidelines for building multiple view visualizations. They claim four criteria regarding how to choose multiple views: diversity, complementarity, parsimony, and decomposition. They also put forward four criteria for presentation and interaction design: space/time resource optimization, self-evidence, consistency, and attention management. Recent empirical research supports these guidelines [Convertino, 2003] and methodologies for designing multiple views should evaluate their design according to these criteria. While these guidelines are well-formulated for 2D media, none have been critically evaluated in the context of 3D worlds as we propose.

Zoom-able Interfaces

Bederson, et al [Bederson, 1996] have proposed that interface designers appeal to user's knowledge about the real world, i.e. that objects appear and behave differently depending on the scale of the view and their context. They propose a new "interface physics" called "semantic zooming" where both the content of the representation and the manipulation affordances it provides are naturally available to the user throughout the information space. The navigational interface physics used in their Pad++ system operates from the proximity and visibility of information objects in a 2.5 D space. Here, users pan and zoom and the elevation off the visualization canvas determines the choice of information representation. More recently, Bederson et al used a scenegraph to represent the visualization space as a "zoomable" surface [Bederson, 2000].

In building their Visual Information Density Adjuster System (VIDA), Woodruff et al [Woodruff, 1998a] adapted the idea of zoomable interface navigation. Instead of changing the representation based on the user's elevation, they change the user's elevation on the basis of representation (detail) the user wants to see- a process they call 'goal-directed zoom' [Woodruff, 1998b]. Users apply constraints on goal visual representations for subdivisions of the display and the system manages the overall display density to remain constant. This system seeks to limit cluttering and sparseness in the visualization display.

VIDA is an application of a cartographic principle that speaks to transitioning between representations- the 'Principle of Constant Information' ([Töpfer, 1966], [Frank, 1994]). This principle holds that that the amount of information (i.e. the number of objects in a viewing area) should remain constant across transitions even if the data is non-uniformly distributed. Conforming to the principle constitutes a 'well-formed' visualization. In VIDA, the principle is applied to a screen grid through 2 techniques: multi-scale representations and selective omission of information. Multi-scale representations modulate information density though techniques like changing object shape, changing object size, removing attribute associations, and changing an object's color. Selective omission refers to select, aggregate, and reclassifying data to reduce the density of the rendered visualization.

The information density metrics reported in Woodruff et al [Woodruff, 1998a; Woodruff, 1998b] are number of objects, or the number of vertices. While their system allow expert users to define their own density functions, the authors note that theirs is an example to demonstrate the capabilities of their system and it is not clear what constitutes a 'good' density metric. In addition, problems like real-time constraint solving cause flicker in the visualization. They note that their planned improvements are to move from a grid-based model to an object-based model.

While these works are informative for navigation of abstract information spaces, a new look at information density and zooming is required to treat IRVE perspectives. In IRVEs, there are typically more navigational degrees of freedom and fewer constraints on user motion relative to the space and objects of interest. In many cases, enhancing abstract information such as text or graphics may not be visible or legible either because of size, color, or occlusion by the referent or nearby objects in the space.

2.2.2 Multimedia

Generally, image information is best to display structure, detail, links of entities and groups; text is better for procedural information, logical conditions, abstract concepts [Ware, 2000]. From an information design perspective, an important tool is a 'Task-Knowledge Structure' analysis ([Sutcliffe, 1994], [Sutcliffe, 2003]), which concentrates on user task and resource analysis to formalize an entity-relationship model. This model enables the effective design of multimedia interfaces and information presentation – i.e. what media resources the user needs visual access to when. This is an important technique for IRVE design as it intends to formally identify items that need user attention and minimize perceptual overload and interference per task. Such an analysis can also help identify familiar chunks of information that can improve cognitive and therefore task efficiency.

Sutcliffe and Faraday published extensively on their work in multimedia comprehension. For example Faraday explored the how to evaluate multimedia interfaces for comprehension [Faraday, 1995]. They examined how eye-tracking data could provide guidelines on how to better integrate text labels and images for procedural knowledge [Faraday, 1998]. Co-references between text and images can drive user's attention and optical fixation points. Eye-tracking allowed them to measure the ordering of attention to different media types and subsequently improve animation timing in the presentation. Their work supports the claims of Chandler & Sweller [1990] that such co-references between text and images can improve the comprehension of complex instructional material. They also evaluated an instructional interface Etiology of Cancer CD Rom for memorization using these techniques [Chandler, 1991; Faraday, 1996; Faraday, 1997]. The integration of text and image information in multimedia presentations has resulted in better user recall than images only.

In addition, they summarize their work as guidelines for multimedia designers. The following are two of some fourteen listed in the publications:

- "Reading captions and labels will lock attention, preventing other elements being processed. Allow time for text elements to be processed: this may range from 1/5 second for a simple word to several seconds if the word is unfamiliar." (pg. 270)
- "Labeling an object will produce shifts of fixation between the object and the label; ensuring that an object and its label are both available together may aid this process. (pg. 270)

More recently, Mayer [Mayer, 2002] described how cognitive theory informs multimedia design principles for learning. To evaluate meaningful learning, they use transfer questions including troubleshooting, redesigning, and deriving principles. Using the dual-channel, limited capacity model of Baddeley (below), he suggests that users are actively selecting and organizing mental representations from pictoral and verbal channels and integrating them into a coherent model and relating that integrated model to other knowledge in long-term memory. Because on-screen text can interfere with the visuospatial channel, speech narration is generally better that text. These benefits shown from these investigations are promising for IRVEs goal of integrating perceptual and abstract information as well.

2.2.3 Virtual Environments

A virtual environment (VE) is a synthetic, three or four-dimensional world rendered in real time in response to user input. The first three dimensions are spatial and typically described in Euclidean coordinates x, y, and z. The fourth dimension is time; objects in the VE may change properties over time for example animating position or size according to some clock or timeline.

Munro et al [Munro, 2002] outlined the cognitive processing issues in virtual environments by the type of information they convey (Table 2.2). In reviewing VE presentations and tutoring systems, the authors note that VEs are especially appropriate for: navigation and locomotion in complex environments, manipulation of complex objects and devices in 3D space, learning abstract concepts with spatial characteristics, complex data analysis, and decision making. Our investigation into IRVE display techniques is especially concerned with improving the latter three.

Location Knowledge	Structural Knowledge
• Relative position • Navigation • 'How to view' (an object) • 'How to use' (an object access & manipulation affordances e.g. a door)	• Part-whole • Support-depend (i.e. gravity) • Containment
Behavioral Knowledge	**Procedural Knowledge**
• Cause-and-effect • Function • Systemic behavior	• Task prerequisite • Goal hierarchy • Action sequence

Table 2.2: Taxonomy of knowledge types for VE presentations (per Munro et al, 2002).

Virtual environments can be run across a wide range of hardware. An important distinction is the difference between immersive VEs and desktop VEs. Desktop systems typically use commercial off-the-shelf components such as monitors, keyboards, and mice and are the most widespread platform to support VEs. Immersive systems on the other hand use head tracking, tracked input devices, and alternative display technologies such as projection walls (such as a CAVE) or head-mounted displays (HMD). An immersive VE appears to surround the user in space, and can lead to the sensation of presence (the feeling of "being there"). This contrasts with desktop systems where the graphical display is does not fill the user's field of view. This property has brought desktop VEs to also be called 'Fishtank Virtual Reality'.

While both immersive and desktop platforms may render at different resolutions and may provide stereoscopy, the common setup is that desktops are mono-scopic and can support higher resolutions. A general research thrust of our 3D Interaction group is to understand the differences between VE platforms and what design parameters should be changed when migrating content and applications. In general, interaction in VEs consists of three activities: Navigation, Selection, and Manipulation [Bowman, 2001a] and desktop and immersive systems require different sets of interaction techniques [Bowman, 2004]. For example, desktop input devices are not tracked and have fewer degrees of freedom than those typically used in immersive settings.

Design principles for interaction techniques in VEs have been described in terms of performance and naturalism [Bowman, 2002]. In spatial navigation for example, the travel technique should impose minimal cognitive load on the user and be learned easily so that it can be automatized and used 'second nature' [Bowman, 2004]. [Pierce, 1997] first leveraged user perspective and proprioception in demonstrating image plane interaction techniques for selection, manipulation, and navigation in immersive environments. The research proposed here will investigate design principles for IRVE information display techniques on desktop systems (monoscopic, no head-tracking). However, being that many IRVE display techniques can be expressed through common scenegraph languages, they may also be run on immersive systems.

Virtual Environment Applications

There are four principal areas that have seen the most successful application of Virtual Environment technology: entertainment, training, phobia treatment, and education. Typically, these applications rely on perceptual or experiential environments, while IRVEs are enhanced with related abstract information.

Perhaps the most well-known entertainment company, Disney Imagineering, has produced a number of location-based attractions for their theme-parks that use immersive VE technology including DisneyQuest and Aladdin [Pausch, 1996]. These systems have entertained thousands of visitors with immersive content from various Disney movies. DisneyQuest is a projection-based system modeled after a pirate ship's deck with props such as a helm and cannons. The Aladdin system uses HMDs and a motor-bike type motion platform for users to fly around an ancient city on a magic carpet and meet characters. These systems are successful for a number of reasons including their consideration of usability issues and the use of interactive 3D as a narrative medium. In both worlds, users have a background story in mind, and their degrees of freedom for interaction and navigation are tightly constrained to that story. Users tended to focus more on the environment, not the technology and showed good signs of presence such as involuntary ducking in response to the environment content.

Arguably, the most successful application of VEs technology on desktop systems is that of gaming. Computer games have consistently pushed the development of 3D graphics rendering technology to increase visual realism, resolution, and frame-rate. Many of the most popular games are played from a first-person perspective; users enter the space navigating, selecting, manipulating objects to achieve the primary goal of victory (collecting points, killing bad guys, maintaining health, etc.). These games are fast-paced, stimulus-rich environments that require players to divide attention among a number of tasks. These console or desktop gaming systems have developed performance-oriented techniques for travel interaction, mapping the 6 degree of freedom (DOF) camera to lower DOF devices such as mice, joysticks, and control key inputs. VRML and X3D also provide a number of navigation modes that allow users to control 2 DOF at a time. Such interaction techniques as implemented for desktop systems should be critically considered when applied to IRVEs.

In the field of training, VEs have been successfully used to train soldiers and agents on scenarios that would be too difficult, expensive, or dangerous to recreate in real life. One such application is the simulation of missions in hostile or unknown territory where a user navigates through the virtual world and attempts to achieve some operational goal. This wayfinding research has led to a number of guidelines as to how to convey route, survey, and location knowledge through VEs [Darken, 1996] & [Darken, 2002]. In addition, there have been a number of compelling large-scale, multi-user systems for simulating tactical operations such as SIMNET and NPSNET [Macedonia, 1997]. These types of systems support military scenarios where multiple users such as pilots, tank drivers, ground troops can all work to coordinate their actions to achieve more efficient or effective execution of an operational goal.

There has been important analytic work in the application of immersive VEs to the treatment of psychological phobias and anxiety [i.e. [Strickland, 1997], [Rothbaum, 1999]. Clinical data collected by controlled-study experiments have shown that graded exposure to anxiety-inducing situations in the virtual environment significantly reduced the users' subjective rating of discomfort to that situation. Physical props can increase treatment effectiveness. Patients of height anxiety and other phobias may benefit from the use of virtual environments in this regard: privacy and cost-effectiveness for i*n vivo* desensitization, and increased effectiveness of behavioral therapy through graded exposure especially when voluntary imagining techniques are not effective. Because this treatment relies on graded exposure to 'being there', these promising results may not transfer to desktop systems where the user is not surrounded by the environment.

The ScienceSpace Project showed that conceptual learning can be aided by features of immersive VEs such as: their spatial, 3-dimensional aspect, their support for users to change their frames of reference, and the inclusion of multi-sensory cues [Salzman, 1999]. The curriculum modules included learning about dynamics and interactions for the physics of Newton, Maxwell, and Pauling. It seems likely that this advantage would also transfer to desktop courseware and applications. Indeed, education researchers have shown improved student performance by augmenting science lectures with desktop virtual environments including the 'Virtual Cell' environment for biology and the processes of cellular respiration [McClean, 2001], [Saini-Eidukat, 1999; White, 1999].

These applications demonstrate the advantages of VE interfaces for the acquisition of knowledge. More work is needed however to determine how these advantages can be leveraged in the embedded and coordinated visualizations of IRVEs. Specifically, we want to investigate how abstract information can be displayed in an IRVE to maximize naturalism and performance.

Display: Sizes, Resolution

Both resolution and physical size of a display play an important role in determining how much information can or should be displayed on a screen [Wei, 2000]. Swaminathan & Sato [Swaminathan, 1997] examined the advantages and disadvantages of large displays with various interface settings and found that for applications where information needs to be carefully studied or modified, 'desktop' settings are useful, but for collaborative, shared view and non-sustained and non-detailed work, a 'distance' setting is more useful. This work orients our design claims and evaluation to the paradigm of the single-user desktop workstation.

Tan et al [Tan, 2003] found evidence that physically large displays aid user's performance due to increased visual immersion; Mackinlay & Heer [Mackinlay, 2004] proposed seam-aware techniques to perceptually compensate for the bezels between tiled monitors; our system rendered views naively, splitting images across monitors as though there were no bezels.

While a number of studies have examined the hardware and the display's (physical) Field of View (e.g. Dsharp display [Czerwinski, 2003]), less is known about the performance benefits related with the Software Field of View (SFOV) and virtual environments. However, Draper et al [Draper, 2001] studied the effects of the horizontal field of view ratios and simulator sickness in head-coupled virtual environments and found that 1:1 ratios were less disruptive than those that were far off. There is also a good body of work on SFOV in the information visualization literature, typically with the goal of overcoming the limitations of small 2D display spaces. Furnas, for example, introduced generalized Fish-Eye views [Furnas, 1981; Furnas, 1986] as technique that allows users to navigate data sets with 'Focus-plus-Context'. Gutwin's recent study [Gutwin, 2003] showed that fisheye views are better for large steering tasks even though they provide distortion at the periphery.

2.3 Integrated Information Spaces

2.3.1 Feature Binding and Working Memory

Despite its commonality and the abundance of demonstrated design guidelines, few details are known about the relation of perceptual and cognitive processes underlying user comprehension of rich graphical data. A survey of the literature implicates the components of Working Memory (WM) are crucial players in the apprehension and comprehension of graphical information [Tardieu, 2003]. While there are alternate models of the architecture of WM ([Baddeley, 1974; Baddeley and Logie, 1999; Baddeley, 2003], [Just, 1996], [Ericsson, 1995]), there is general agreement that there are capacity thresholds ([Miller, 1956], [Vogel, 2001]). The research proposed here seeks to determine how pre-attentive and Gestalt visualization designs can better utilize Visual WM capacity and in turn improve user performance in Search and Comparison tasks in IRVEs.

Models of Working Memory

Early models from cognitive psychology and human information processing postulated sensory, short--term memory, and long-term memory stores. The sensory store for vision has been described by [Kossyln, 1994], and includes the notion of a visual buffer that is selectively processed by an attention window. The attention window may be controlled by top-down or bottom-up (stimulus-based) processes and feeds both dorsal (what) and ventral (where) visual processing. Long Term Memory (LTM) is the system responsible for our persistent memories of procedural, episodic, and semantic information.

Subsequent evidence has led to the definition of a componentized 'Working Memory' (WM) to replace the single short-term memory (STM) store, and a number of prominent researchers have since weighed in with alternate models of the architecture of WM and its relation to the rest of the cognitive system. Because the WM construct involves both storage and processing features and seems to have a limited capacity, one point of contention for these models is how to account for the range of performance differences across individuals and ages.

Working Memory in this formulation (e.g. [Baddeley, 2003], Figure 2.2) consists of multiple fluid components including the visuospatial sketchpad, the episodic buffer, and the phonological loop. These 3 subcomponents are managed by a central executive component that manages attention (focusing, dividing, and switching) and connects the other working memory components to LTM. The central executive is a modality-free processing center integrating and shifting between information in the sensory storage buffers. Support for this model comes from a number of empirical results that demonstrate interference within but not between verbal and visual stimuli. While the precise nature of how these memory systems are composed and interact is still an open research question, the qualitative distinction is the basis for most memory research.

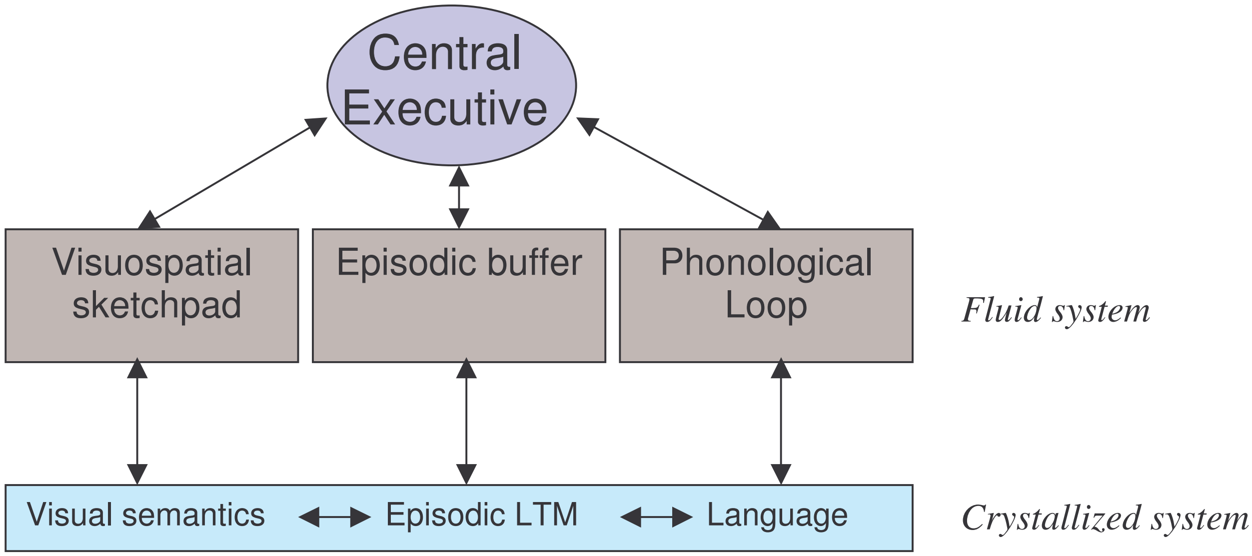


Figure 2.2: Revised Multi-Component Working Memory [Baddeley, 2003]

Collette & Van der Linden [Collete, 2002] examined a set of central executive functions in the brain with functional Magnetic Resonance Imaging (fMRI) controlling for storage. Manipulating a set of processing functions attributed to the central executive, they found increased activation across both prefrontal and parietal regions. While the specific cognitive WM functions for updating, inhibiting, shifting process, and coordinating dual-tasks were statistically separable, they shared some commonalities. The authors interpret their results to mean that executive functioning may "be better conceptualized in terms of interrelationships within a network of cerebral area rather than associations between one executive function and a few specific prefrontal cerebral areas" (pg. 122). In addition it suggests that a more rigorous computational model of the central executive's common and specific functions is needed.

Ericsson & Kintsch [1995] propose an alternate explanation of cognitive architecture that can account for 2 important facts that they claim Baddeley's modal model of WM cannot. First, experts and skilled performers have expanded memory capacities for complex tasks in their domain. This capacity allows then to efficiently access and use knowledge from their LTM. Second, skilled performers can stop working on a problem and resume it later without a major performance impact. Instead of simply replacing STM with WM (what they call STWM), Ericsson & Kintsch propose expanding the model with another store: Long-Term Working Memory (LTWM), which is a more stable storage medium and is accessible via cues in STWM. Through study and practice in some domain (including text comprehension), these cues can be structured to efficiently encode and retrieve information between STWM and LTM. They studied expert and domain-specific strategies in 5 task domains: mental abacus calculation, mental multiplication, dinner orders, medical diagnosis, and chess. In each, they found evidence of distinct retrieval structures being used to avoid interference and improve performance. This notion of mnemonic skill is consistent with Maguire et al's recent work [Macguire, 2003] of neuroimaging world-class memory performers. According to Ericsson [Ericsson, 2003], the different patterns of fMRI activation between experts and controls is attributable to the differences in memory strategy - experts have established retrieval structures using their imagery and 'method of loci' strategies, strategies the controls did not use. In turn, these expert strategies activated brain areas associated to spatial memory and navigation.

Just & Carpenter [1996] take a different view on WM and make a capacity argument to explain individual differences in task performance. In their theory, there are capacity constraints in WM and the degree of information encapsulation in WM determines the efficiency and power of the system. Thus, performance is determined by the management of the WM chunks within capacity and not by separate modular resources. In their work, they used a reading span test (attributed to phonological WM), dividing subjects into high, medium, and low span groups. Subjects read passages with complex sentences (e.g. subject and object relative clauses) and ambiguous or polysemius words. As predicted for the variety of sentence types, high-span individuals were better are comprehension correctness and seemed to be able to maintain ambiguous terms across long sentences to their semantic resolution. Some researchers have gone so far as to extended this capacity argument to implicate WM as being a principle factor in *g* or general fluid intelligence ([Conway, 2003], [Miyake, 2001]).

Since the early formulation of STM, alternate models of cognition and memory have emerged including WM, WM with capacity constraints, and the distinction between STWM and LTWM components. How each accounts for individual differences in cognitive and memory performance is at the center of the debate. Baddeley's approach is an architectural one that describes WM and divides subcomponents in order to replace the traditional STM. Ericsson & Kintsch maintain that Baddeley's model is insufficient to account for domain-specific expertise and add another processing module to encode and retrieve information from LTM. Contrasting Ericsson & Kintsch, Just & Carpenter claim that another module is not needed to account for expertise but rather that capacity in WM is the significant issue. Further research is required into the units and methods of chunking and encapsulation in WM components and the nature of central executive functions.

Capacity of Working Memory

The capacity for verbal information (numbers, letters, words) in the phonological loop is best known from Miller's number [1956]: 7 +/- 2 items. This capacity can be effectively increased through 'subitizing' or 'chunking' strategies that aggregate items of information. This construct has specific importance to the notion of expertise and the use of chunking strategies to avoid perceptual and cognitive overload in the user.

Vogel et al [2001] found evidence that visual WM capacity should be defined in terms of integrated objects rather than individual features. In a series of 16 experiments, they attempted to control for extraneous factors such as verbal WM, perceptual encoding, and WM encoding variability. They showed users a sample set of objects such as colored, oriented, or conjoined shapes; after an inter-stimulus delay interval they presented a test set and asked users to indicate if the two sets were identical or different. The delay intervals were intended to eliminate effects from the iconic (perceptual) store and accuracy was stressed rather than speed.

They found that 3-4 objects could be stored concurrently in naïve subject's WM and that objects with multiple features do not require more capacity than single-feature objects. They examined conjunctions of 2 and 4 features with no significant difference in performance below the capacity threshold. This is not to say that 3 or 4 items of high-fidelity are necessarily represented WM; in fact they admit that WM may contain more items of low-fidelity representation. In addition, they note that individual differences and experience may lead to variability on the capacity they report. Finally, Vogel et al propose a mechanism of synchronized neural firings that can account for their results.

It has been demonstrated that a maximum of about 4 items could be simultaneously compared during a visual search task. However the definition of what constitutes a visual 'item' in the visuospatial sketchpad is not clear and there are alternate accounts of what constitutes a perceptual unit for vision. For example, the Gestalt principles (recently summarized by Ware [2000]) address the criteria for the perception of unity and grouping of figures. These principles are crucial when considering the design of multimedia environments, and the possibility of 'chunking' visual items and features. While the importance of these principles to design is not debated, metrics for quantitative assessment are elusive and difficult to apply. Algorithms inspired from the work of David Marr (2 ½ D sketch, [Marr, 1982]) and machine vision are computationally intensive and still in their early stages. [Biederman, 1987] proposed 'geons' as fundamental units of shape in object perception; and more recently Biederman & Gerhardstein [Biederman, 1993] found evidence for this in viewpoint invariance during shape recognition. In addition, Irani and Ware [Irani, 2000] showed that using such 3D primitives to display information structures was significantly better for substructure identification than in displays using 2D boxes and lines.

[Saiki, 2003] however, has found evidence that object representations may not be the functional unit of WM. He used another testing paradigm to demonstrate that when 4-6 objects are dynamically occluded they are not represented as coherent objects. This paradigm is termed 'multiple object permanence tracking'; users are required to detect changes of features (e.g. color, shape) while in the objects are in motion and pass behind occluders. In a set of 4 experiments, users were required to detect feature switches of 4-6 objects in motion at a variety of speeds. Even when the motion was slow and predictable, Saiki found that users failed to maintain features and conjunctions of features. While this result might be interpreted as supporting the limited capacity of visual WM, it also indicates that integrated object representations may not be the basis for WM units. Saiki proposes that feature-location binding is much more transient than previously supposed and that what is tracked are only visual indices for 'just-in-time' processing. In addition, Saiki claims that these results also support evidence from change blindness [Rensink, 2000; Simmons, 2000].

Individual differences may also exist in how well users can interpret dynamic 3D spatial stimuli. Given the results of Saiki [2003] mentioned above, there may be additional cognitive factor tests that can assess performance for dynamic spatial stimuli. This must be investigated further (i.e. [Bradshaw, 2003]). In addition, this research will help formulate further hypotheses as to how abilities such as Closure Speed and Closure Flexibility are implicated in the comprehension of rich, multi-dimensional graphic information.

Structure of Working Memory

In order to investigate Baddeley's model of Working Memory, Miyake et al [2001] undertook a latent-variable analysis of spatial abilities and the functions of visuospatial working memory and central executive. Specifically, they were looking to establish an empirical and theoretical relation between these two components as opposed to the verbal (phonological) domain. They found that cognitive factors (i.e. [Carroll, 1993]) such as Spatial Visualization and Spatial Relations significantly loaded the central executive component and not the visuospatial component. In addition, the Perceptual Speed factor loaded both the central executive and visuospatial working memory with a slight emphasis toward the central executive. They used this result to speculate that the visuospatial component of Working Memory is strongly tied to the central executive and this asymmetry (to phonological processing) may be due to the need to manage task goals and resist interference from external stimuli.

Of specific interest to graphics and visualization, is the contents and capacity of the visuospatial sketchpad. Despite the fact that in most cases visual and spatial information are tightly interlinked, there is evidence from dual-task studies, brain damaged patients, and PET studies [Baddeley, 1980; Farah, 1988; Smith, 1997] respectively) that there are separate visual and spatial sub-systems to WM. This aligns with [Logie, 1995]'s division into: the 'visual cache' for form and color, and the 'inner scribe' for spatial and movement information.

The nature of these structural divisions will be further explored through this research using information-rich virtual environments because spatial relations and navigation may operate independently of other perceptual and WM processes such as feature-binding. Some important questions raised by this work are*:* how do individual differences affect performance on the interpretation of IRVE stimuli?, and does this performance tell us anything about the implied relation between the visuospatial sketchpad and the central executive?

Computation in Working Memory

In the tradition of models of human-information processing such as [Card, 1983] and Anderson's ACT* program [Anderson, 1983], [Lohse, 1991] developed an explanatory and predictive model for the cognitive processing required to interpret graphical depictions of abstract information. The model, named UCIE (Understanding Cognitive Information Engineering), is similar and complementary to GOMS (Goals, Operators, Methods, and Selection rules). It is similar in that it contains structural and quantitative models of performance. It is complementary in that it considers scanning (saccade), perception (focus), and discrimination (interpretation and semantic lookup) processes in more detail. The model uses schema for each kind of graphical presentation and this schema directs how the system scans and decomposes the stimuli.

Lohse used the model to evaluate line graph, bar graph, and tabular depictions of a data set for 3 types of questions: read, compare, and identify trends. Using regression analysis with 561 observations from 28 subjects on the 3 questions, the UCIE model predicted 58% of the variation in reaction times (r=.759). The discrimination time alone (predicted by UCIE) explained 36% of the variation. UCIE predicted that grid lines on bar and line graphs would facilitate answering read questions, which they did. The model as published could not account for grid lines on tables however. The UCIE model bears on this research in two ways. First it provides a computational description of perceptual and cognitive processes underlying graph comprehension. Second, it opens the possibility that individual differences could be explained in part by the nature of the user's 'graph schema' – the knowledge a user uses to encode the graph into WM and interpret it.

Working Memory Summary

This research on Working Memory has direct implications for the design of IRVEs. However it falls short of answering many important questions. For example, by the Gestalt principle of Connectedness, are embedded representations chunked as one item? How do Gestalt principles rank with each other in interactive, rich-media environments – what becomes perceived as a unit figure? How many figures can be apprehended and compared at once? If the view perspective may be interactively controlled (in 3D), will performance be reduced or improved?

[Zhang, 1994] found evidence to suggest that users employ the visual display as an external working memory store. Users tend to favor the economical approach of keeping knowledge in the world rather than in the head. This effectively lowers any storage overhead, but can also reduce the number of metal operations and hence introduced errors. The cognitive value of explicit diagrammatic representations over linguistic representations was also described by [Larkin, 1987]. It is an open question how many more words an IRVE is worth.

2.3.2 Augmented Reality

It is important to note that IRVEs share a great deal in common with augmented reality (AR) [e.g. Hoellerer, Feiner et al., 1999]. Prior work in AR has included research on information display and layout [Bell, 2001], and user interaction [Prince, 2002], similar to the research we propose. AR applications enhance the physical world with additional information, much of it abstract, while IRVEs enhance the virtual world with abstract information. The key difference between IRVEs and AR then, is that IRVEs are purely synthetic, which gives them much more flexibility - information objects in an IRVE can be perfectly registered with the world, and realistic objects in an IRVE can be animated, moved, scaled, and manipulated by the user at a distance, for example. Thus, while prior work in AR provides a good starting point (and we have used some of this work in the tools described below), the design of IRVEs should be studied separately.

Feiner et al [Feiner, 1993] enumerated locations for the display of enhancing information in AR. They divided display locations into 'surround-fixed', 'display-fixed', and 'world-fixed' categories. Surround-fixed annotations do not have a specific relationship to the physical world, but are displayed at a fixed position in the surround. In AR terms, the surround is a spherical display space that envelopes the user regardless of their position. Annotations displayed here are rendered orthogonal to the users head angle or camera orientation. Display-fixed annotations retain position relative to the display no matter where the camera or user's head moves. World-fixed information is displayed relative to objects or locations in the physical world.

The display locations described by Feiner et al. were organized by AR implementation considerations and are thus semantic technicalities for AR developers, but few others. For IRVEs, we must adapt the terminology to incorporate synthetic environments on desktop and immersive devices. We will characterize display locations according to a user's perspective and what coordinate space the information resides in: abstract information may be located in *object space*, *world space*, *user space*, *viewport space*, or *display space*.

First, we subdivide the idea of world-fixed into Object Space and World Space. Annotations in object space are located relative to the object they refer to; if the user perspective or the object moves, the annotation is moved accordingly for tight spatial coupling. World Space annotations are located relative to world coordinates and do not directly refer to objects but rather to areas and regions of space.

Second, we redefine the notions of surround-fixed and display-fixed to accommodate the variety of IRVEs. One important distinction to recognize is the difference between to User Space and Viewport Space. User Space is 3-dimensional and surrounds the user - annotation locations are relative to the user's position. This space moves with the user and can include virtual toolbelts or spherical or planar layout spaces around the body. Head tracking or alternative displays such as PDAs or tablets are essential to make this a viable layout space. The Viewport Space is the 2D image plane of the rendered VE in either desktop or immersive contexts. Display space now becomes a definition for desktop and windowing systems specifically: windows, frames, and pop-up that are drawn externally to the VE's rendering.

There is one additional subtlety to these new definitions. If an annotation is defined in User Space *and* it is also constrained to the head or camera orientation, it is perceptually equivalent to a Viewport Space annotation. In and VRML and X3D worlds, a Heads Up Display (HUD) must be implemented this way. While this setup achieves the same end as an over-layed HUD in Viewport Space, it is awkward in that authors must know the Viewpoint's fieldOfView (FOV) and the dimensions or aspect ratio of the rendered projection, but have no guarantee of the distance to the near-clipping plane.

Bell et al [2001] developed a useful strategy for dynamically labeling environment features in mobile AR. Their method involved an image-plane management algorithm that measured. They used a Binary Space Partition tree (BSP) to determine visibility order of arbitrary projections of the scene. From visible surface data for each frame, a view-plane representation is managed that contains each visible object's rectangular extent. The algorithm identifies then empty spaces in the view-plane and draws object annotation (such as labels or images) in the empty spaces by depth order priority. The results of this approach are impressive.

2.3.3 Information-Rich Virtual Environments (IRVEs)

Parts of the IRVE story exist in the literature. For example, [Dykstra, 1994] was the first to demonstrate how X11 windows could be embedded and rendered within virtual environments; this is an enabling technique for IRVEs. Plumlee and Ware [Plumlee, 2003] have used multiple embedded views and frames of reference for the navigation of large-scale virtual environments, but their augmenting views only provide additional spatial cues or alternative views of the perceptual world. To our knowledge, the first descriptions of the challenges for integrating symbolic and perceptual information in VEs was Bolter et al, [1995].

Bowman et al implemented and evaluated Information-Rich Virtual Environments through the Virtual Venue and the Virtual Habitat [Bowman, 1998]. The Virtual Venue aimed to provide perceptual and abstract information in the context of an athletic venue (the swimming and diving arena constructed for the 1996 Atlanta Olympic Games). It included a 3D model of the venue, text information about the Olympic sports of swimming and diving, audio "labels" for various components of the building, spatial hyperlinks between text and locations, a slideshow of images related to the environment, and an animated diver that could perform several types of dives. An evaluation of this application showed that the most effective information presentation techniques were those that were "tightly coupled" to the environment.

The Virtual Habitat application is an IRVE for environmental design education. It consists of a 3D model of a gorilla habitat at Zoo Atlanta, enhanced with abstract information including textual signs, audio clips, and overview maps. Based on the principle of tight coupling, the abstract information was embedded within the environment at relevant locations. An evaluation of student learning in this application showed a trend towards increased understanding of environmental design principles.

These two early applications showed the enormous potential of IRVEs to allow users to form mental associations between perceptual and abstract information. They also demonstrate that the success of the applications depended on the display and layout of the abstract information, and the interaction techniques used to navigate and access the information. Furthermore, these applications used only simple abstract information (text or audio). In our research, we will design techniques and perform studies that will lead to a more thorough understanding of the issues of information display and interaction in IRVEs, based on a theoretical framework, and including complex visualizations of abstract information.

Parallel Graphics' Virtual Manuals solution demonstrates the integration of abstract information (e.g. names, part numbers) within the spatial world and with external windows for training applications in operation and maintenance. Temporal information is rendered through animated sequences of assembly and dis-assembly for example. This approach is consistent with HCI research in comprehension and user's mental models. For example, users gained improved situational awareness, understanding, and recall through multimedia presentations integrating these features [Sutcliffe, 1994; Faraday, 1996].

The trend toward IRVEs can also be seen in Sheppard's recent survey [Sheppard, 2004] of construction-related 4D applications. Applications such as PM-Vision and ConstructSim for example, provide an integrated workspace for construction project managers to relate various costs and timelines to their models. Users can switch back and forth between views to examine and change parameters and scenarios. In addition, Domos Software's 4D Builder Suite, integrates CAD geometries and project planning softwares to give planners an integrated view of various material and scheduling choices; the suite uses XML to describe the relations between VRML object identifiers and the planning and profile definitions [Domos, 2004].

Yost et al [Yost, 2006] have formulated terminology that is consistent with the theory of IRVEs. They have described a taxonomy for multiple-view visualizations and the role of context on desktop and large-screen displays. In their terms, a virtual environment can be described as a 'Structure-centric' visualization – a view where spatial/perceptual information is depicted. 'Attribute-centric' visualizations are any depiction of abstract information. We note the parallel of their research and hope for cross-validation of any guidelines derived in this work.

A recent study in our group examined exploration and search tasks in immersive IRVEs using Head-Mounted Displays (HMDs) [Chen, 2004]. The study compared combinations of two navigation techniques and two layout spaces for textual information. The two spaces for annotation labels were Viewport Space and Object Space; the two navigation techniques were HOMER and Go-Go. For naïve search (Table 2.3), the Viewport Space was significantly better for both navigation types and a significant advantage to a combination of HUD and Go-Go navigation was demonstrated. While there were some confounding variables in that study (for example label orientation in the Object Space condition as well as association parameters), it underscores the fact that IRVE navigation and display techniques should be considered together.

Type	Tasks
1	Search for abstract information and then search for perceptual information.
2	Search for perceptual information and then search for abstract information.
3	Search for perceptual information, followed by additional perceptual information, and then abstract information
4	Search for abstract information, followed by perceptual information, and then abstract information

Table 2.3. IRVE Search Task types used in Chen et al, 2004.

Chen has continued to innovate IRVE interfaces involving immersive technologies. In a recent project [Chen, 2005], she compared the costs of context switching between multiple platforms and displays. Chen analyzed search task performance between a desktop display (Display Space) and a tablet PC + CAVE wall (User space). We note the strong complement of her research and hope for cross-validation of any guidelines derived in this work.

These applications are beginning to demonstrate the power of IRVEs to provide a unified environment for visual analysis. In building and construction, this can improve efficiency by identifying and minimizing potential conflicts and costs in the planning stage. In Chapter 5, we will examine additional IRVE systems we have implemented and evaluated in the domain of biomedicine.

2.4 Usability Engineering & Design Guidelines

To design and deploy integrated information spaces that meet user requirements, developers face a number of challenges. Across application domains, these requirements involve display and interaction techniques to relate heterogeneous data types to one another and to find spatial patterns and trends in the data. To date, interface features and software architectures have been ad hoc and primarily domain-specific. In order to support IRVE functionality, a rational design approach is required. The data and presentation problem of integrated information spaces is common to simulation and design applications in a number of domains.

Constructing optimal IRVEs will require development of new information access interfaces, efficient database storage and integration, and open software architecture for mapping data to real-time graphical displays. From the end-user perspective, the IRVE must be intuitive and easy to use, facilitate insight generation from massive and complex databases, drive data retrieval, and support perceptual similarity to the domain area. While we must grapple with many of these implementation issues, our research here will consider those secondary to the fundamental issues of perception and information design.

The development of our theoretical framework and interface components of IRVEs will be informed by prototype systems built through user-centered design and the usability engineering process [Rosson, 2002]. User-centered design refers to a product design process that considers the user population and its demographics, requirements, work environment, and tasks as the primary driving force. Many researchers have shown that a user-centered design process produces systems that are more usable, more accepted, and less problematic than systems designed with the focus on features. User-centered design is a part of the overall process known as usability engineering, which is based on an iterative design-evaluate-redesign cycle.

Usability engineering is an approach to software development in which target levels of system usability are specified in advance, and the system is engineered toward these measures. Designers work through a process of identifying the user **activities** the system must support, the **information** that is required for users to understand the system and task state, and the **interaction**s required to support those tasks. For each feature in a design, usability engineers identify the tradeoffs of that feature and then analyze their claims that the feature resolves the tradeoff. Usability evaluation of VEs and IRVEs presents issues above and beyond 2D interfaces.

We will design and develop our IRVE interface prototypes using the VE-specific usability engineering process described by Gabbard et al. [Gabbard, Hix et al., 1999] and later Bowman et al [Bowman, 2002]. This process includes:

- *User analysis*: a detailed description of the user population, including demographics, technical expertise, typical workflows, typical work environment, collaboration patterns, etc.
- *Task analysis*: a detailed assessment of the goals of the user and the tasks used to achieve those goals, including inputs and outputs, constraints, sequencing of tasks, etc.
- *Scenario development*: a set of typical usage scenarios for the 3D tools, based on the user and task analyses, used as test cases for the various types of evaluation.
- *Heuristic evaluation*: evaluation of predicted performance and usability based on interface experts' application of heuristics and guidelines for 3D interfaces.
- *Formative evaluation*: task-based evaluations by members of the user population in order to inform the redesign of interaction techniques or interface elements.
- *Summative evaluation*: the comparative evaluation of various alternatives for interaction techniques or interface elements based on the results of the iterative design/evaluation process.

The usability engineering and prototyping process will enable us to develop IRVE feature sets and claims about their implications. At this point, the claims must be tested and the inferfaces evaluated for usability. There are generally two methods for usability evaluations: Analytic and Empirical [Rosson and Carroll, 2002]. Analytic methods require expertise to apply [Doubleday, 1997] and can raise validity concerns [Gray, 1998]; however, they may be used early and often on the development process. Empirical methods can provide better validity, but field study and laboratory testing can be expensive.

The research proposed here will use Empirical evaluations in order identify benefitial display techniques for various IRVEs. Consequently, this program will produce methods and heuristics that can be applied to Analytic evaluations of IRVE. We will use the testbed evaluation method [Bowman, 2001b] to set up experiments and gather data. A testbed is a software platform that can support the systematic manipulation and empirical evaluation of IRVE display and interaction parameters. In order to test the significant parameters of the IRVE design space, we will develop a canonical IRVE testbed that will allow us to compose runtimes for IRVE display spaces and systematically vary design variables. Testbed content for the studies are typically generic or of some simple domain.

Heuristic evaluation is an analytic usability evaluation method that does not require end-user participation and yields qualitative results as to the usability of the system or interface in question [Bowman, 2002]. Evaluators examine the system in terms of a set of heuristics or 'rules of thumb' that help to identify and rank usability problems. Heuristic evaluation has principally been applied to 2D GUI interfaces and has been shown to be effective in identifying the majority of usability problems when three to five evaluators are used [Nielsen and Molich, 1992]. While many of these heuristics apply to computer interfaces generally (such as undo/redo, good error messages, system status, consistent vocabulary, and help), analogous rules of thumb have not been formulated for virtual environments specifically.

In order to use heuristics to evaluate 3D virtual environments, the system and application requirements should be clearly stated and formulated through usability engineering processes such as Scenario Based Design [Rosson and Carroll, 2002] or Critical Parameters [Newman, 1997]. Here is sample set addressing design issues and categories of user activities in VEs:

Appropriate Veridicality

Veridicality is the correspondence of interaction techniques to natural, real-world actions. Natural metaphors can increase learnability but may limit the use of 'magic' techniques that could make tasks more efficient [Rosson & Carroll, 2002]. Consider the task and user audience first. If a VE system has been designed for soldier training, teleportation would violate this heuristic.

Appropriate Device Modalities

The use of multimodal input and display devices can provide varied interactions and compelling cues; however excessive sensory and attentional requirements may be distracting or cause fatigue and simulator sickness [Rosson & Carroll, 2002]. Use multimodal devices and cues sparingly and only when the benefits for the user are clearly defined. If a VE system has been designed for collaborative visualization, speech input and audio cues could seriously interfere with the primary task. If a VE system has been designed for physical manipulation tasks, a haptic device would be preferable to a wand.

Presence & Performance

Situational awareness, or user presence in a VE, can be essential to the success of certain applications. System or situational events that break presence should be avoided. If a VE system has been designed for entertainment or simulation, discontinuous frames or ghosting of objects can break the immersive quality or confuse users about the worlds' state.

User Control and Freedom of Navigation

User disorientation is a major usability problem in VEs. Navigation types should be appropriate and constrained to the application requirements. Give users only as much control as needed. Preset viewpoints and camera animation paths or invisible walls are useful in guiding users to relevant locations in the world. If flying and rotational abilities are unconstrained, provide affordances to reset a user's view orientation. In large-scale VEs where wayfinding may be a problem, provide 'landmarks' and/or maps to leverage user's natural spatial abilities. If a VE system has been designed for architectural walkthroughs and the users are clients who are new to VE travel metaphors and techniques, unconstrained navigation can quickly lead to frustration and confusion.

User Control and Object Selection

Provide feedback (such as highlighting) on objects that are identified by pointing or are candidates for selection. This allows users more direct understanding as to the result of their action(s). Use selection techniques appropriate to the scale and distance that users will operate in. At close distances the problem may not be so pronounced, but at large distances, accuracy may suffer. If a VE system has been designed for working with small objects that are densely packed, a large selection unit (ie 'cursor') will be unusable.

User Control and Freedom of Manipulation

The degrees of freedom that input hardware provides should be appropriate to application requirements- not more, not less. In addition, the directness, sensitivity, and constraints of user control should be tuned appropriately. These considerations can prevent user errors and increase satisfaction. If a VE system has been designed for design of nanoscale models, manipulation values in the micron scale should not be used. If a VE system has been designed for machine maintenance training, manipulation should be constrained to the physics of the parts involved.

Aesthetic and Minimalist Design

Irrelevant or unused functionality options should not be included in the interface; they may compete with relevant options for space and visibility. In addition, unnecessary use of animations should be avoided: add content kinematics and details sparingly in order to avoid distraction and performance penalties. If a VE system has been designed for middle school students, affordances and task options that a college student or professional researcher would expect should not be included. If a system is designed for both novice and expert use, provide a controlled 'exposure of power' to system functionality through the use of 'accelerators' for common tasks.

This is a preliminary sketch of some general usability heuristics for VEs. The results of our research will generate a similar list for IRVEs. Heuristics are more general than guidelines as guidelines provide the designer more concrete advice on how to resolve a particular tradeoff. For example with further substantiation, the results of Chen et al [2004] mentioned could be positioned as guidelines:

- "HUD is a better display technique for naïve search tasks related to abstract information in densely packed environments. This is because the user can directly access the abstract information without the need to locate the actual position of the perceptual object in a crowded world." And

- ‘Go-Go interaction technique is better suited for navigation in environments that require easy and more flexible movements. In applications like architectural walkthroughs where density is minimal or occlusion is high, the Go-Go technique performs better because there is no explicit target selection needed.’

With empirical and analytic evidence from task performance IRVEs, this research provides substantiated guidelines to resolve fundamental tradeoffs in IRVEs designs for Search and Comparison tasks. For example how to mitigate the Association – Occlusion tradeoff and the Legibility-Relative Size tradeoff, which are described in the next chapter.

3. Information-Rich Virtual Environments

In this chapter, we detail the nature of Information-Rich Virtual Environments (IRVEs) and position them as a strategic solution to the problems of integrated information spaces. The first half of the chapter provides a more formal basis to discuss the nature of IRVEs: we provide a set of definitions about IRVEs and their properties, we describe the types of activities and tasks that IRVEs must support, and we enumerate a set of specific design goals for IRVEs.

In the second half of the chapter, we provide examples of IRVE display techniques and detail the display components we have developed in this research. We also provide an XML-based description of these IRVE components. This is more than simply convenience or exposition – the XML formalisms enable validation of the syntactic and semantic content of an IRVE, guaranteeing that it is composed and delivered in a reliable manner. In the federated world of the WWW, IRVEs are often data-driven or composed and delivered dynamically; it is crucial that IRVEs inter-operate with and leverage foundational web technologies such as XML.

3.1 Definitions

The first crucial step towards a more complete understanding of IRVEs is a precise definition of the term. Previously, Bowman wrote that IRVEs "…consist not only of three-dimensional graphics and other spatial data, but also include information of an abstract or symbolic nature that is related to the space," and that IRVEs "embed symbolic information within a realistic 3D environment" [Bowman, 1999]. These statements convey the sense of what we mean by IRVE, but they leave significant room for interpretation. What is meant by "spatial," "abstract," and "symbolic" information? What makes a VE "realistic?" The definitions given below serve to disambiguate these terms.

We begin with a set of definitions of terms that will be used to define an IRVE:

(from [Bowman, 2003a])

1. A **virtual environment (VE)** is a synthetic, spatial (usually 3-dimensional) world seen from a first-person point of view. The view in a VE is under the real-time control of the user. Typically, VEs contain purely perceptual information (geometry, lighting, textures, etc.). VEs, as a spatial visualization domain can be described as 'Structure-centric'.

2. **Abstract information** is information that is not normally directly perceptible in the physical world. For example, information about the visual appearance or surface texture of a table is directly perceptible, while information about its date and place of manufacture is not (this information is thus abstract). Nominal, categorical, ordinal, and most quantitative data are considered abstract. Taken together, the abstract information can form abstract structures distinct from the sensory or spatial structure of the VE. Shneiderman [Shneiderman, 1996] defines a taxonomy of such abstract structures including temporal, 1D, 2D, 3D, multi-dimensional, tree, and network. Information visualization techniques. Card et al [1999] provide methods for the display of such structures. Information Visualizations can be described as 'Attribute-centric'.

3. A VE is said to be **realistic** if its perceptible components represent components that would normally be perceptible in the physical world. If a VE's components represent abstract information (see #2) then the VE is not realistic, but abstract. For example, a virtual Greek temple (existed in the past), Statue of Liberty (exists in the present), DNA molecule (exists at an unfamiliar scale), or city of Atlantis (exists in fantasy) could all be considered realistic. On the other hand, a VE displaying spheres at various points in 3D space to represent three parameters of the items in a library's collection would be abstract.

These three terms allow us to define IRVEs

4. An **information-rich virtual environment (IRVE)** is a realistic VE that is enhanced through the addition of related abstract information.

We also further define the space of IRVEs:

5. IRVEs exist along a continuum that measures the *fidelity of the perceptual information mapping*. In other words, how faithfully does the IRVE represent the perceptual information from the physical world in the virtual world? In some cases, perceptual information will be *changed* to show some abstract information about a location or object. In other cases, new information/objects will be *added* to the environment without changing the perceptual information of the original environment. The two extremes of this continuum are "Pure scientific visualization," which changes perceptual information (e.g. the color of a wall) to represent some abstract information (e.g. the air pressure at each point on the wall), and "Information-enhanced VEs," which represent the physical environment with as much perceptual fidelity as possible, and add additional abstract information in the form of text, audio, video, graphs, etc.

6. When we describe IRVEs, we will consider any attribute-centric (abstract information) visualization an 'Annotation'. As we have mentioned, annotations may consist of a variety of information and may take many forms. We will use the term 'Referent Object' to refer to the spatial/perceptual object in the (virtual) world that is annotated.

7. Other dimensions in the space of IRVEs include the *variety of abstract information types* present in the environment, and the *density of abstract information* in the environment. Density will be very hard to define quantitatively, but it could still be useful as a qualitative measure.

8. "Pure information visualization" (e.g. a 3D scatterplot of census data) is not an IRVE because the VE is abstract, not realistic. All of the information in the environment is abstract information that has been mapped to a perceptual form. IRVEs, on the other hand, add information visualization to realistic VEs to provide richness.

To make this definition more concrete, consider the example of VEs for design and review in building construction presented in Chapter 1. Good plans take into account the spatial layout of the home as well as other information such as cost, materials, and schedule. The spatial and abstract information are tightly interrelated, and must be considered and understood together by the architect or contractor. Users of such an application need a) virtual space with perceptual fidelity, b) access to related abstract information, and c) an understanding of the relationships between the perceptual and abstract information.

3.2 IRVE Activities and Tasks

Activities are high-level descriptions of user goals that involve artifacts and their context of use. In a usability engineering approach for example, workflow and social implications of design decisions can be assessed using ethnography and participatory design. Activity theory [Nardi, 1992] is another tool that can be used to formulate requirements and designs through methods such as checklists [Kaptelinin, 1997]. The activities users need to perform when using the system drive the subsequent processes of Information and Interaction design [Rosson and Carroll, 2003].

Tasks are the next level of decomposition in application design and may be formalized through Hierarchical Task Analysis [Diaper, 1989] or Task Action Grammars [Payne, 1986] for example. This phase binds tasks to interface features and enumerates the sub-tasks and artifacts required for each to be accomplished. A **Task-Knowledge Structure** [Sutcliffe, 1994; Sutcliffe, 2003] extends this analysis with an entity-relation diagram that structures what information users need access to in order to complete the task. Such an analysis of user tasks can provide essential requirements for the visual interface.

For IRVEs, we have identified 4 categories of tasks: exploration, search, comparison, and pattern finding. Exploration and search require navigating the world space and the information space and recognizing that a perceptual object has abstract information associated with it. Comparison requires examining abstract and perceptual data that are interrelated but may have unique structures. Finding patterns and trends refers to the recognition and extraction of higher-level forms in perceptual and abstract information.

In IRVEs, these activities will be performed both *within and between* the spatial, abstract, and temporal information. As we have noted, there are fields whose particular interest is examining performance within these respective types (i.e. Virtual Environments and Information Visualization). For example, [Shneiderman, 1996] enumerated a set of generic information visualization tasks that provide a structure to design interface features. For Virtual Environments, [Bowman, 2004] proposes a set of generic tasks whose requirements we design for. Figure 3.1 shows the combinatory relationship of these tasks for IRVEs. However, even the union of these task sets not adequate to capture the richness of IRVE use.

Consider that in some situations, users want to examine abstract information according to its context in the space. For example in our architectural scenario, the architect is designing a model in a VE and wants to use cost information in deciding where to locate a door. By moving the door in the model, the cost view is updated. Similarly he or she might wonder 'how much does this second floor bath fixture cost?'; by selecting it in the VE, the proper item is highlighted in the cost view.

In other situations, users may also use abstract information as an index into space. For example, from a display of the construction production schedule, elements to be completed by a certain date are selected by the architect. The VE responds by highlighting those elements in the 3D architectural plan, or temporarily filtering other elements from the plan. The architect could similarly view the location of all building components that are greater than a certain cost, etc.

For the problem of integrated information spaces, it is crucial that we examine how these activities can be supported *between* information types. Therefore, we introduce the notion of a **task-information mapping**:

- **Spatial Information to Abstract Information:** In this case, the user wishes to use the VE as an index into the abstract information. An example task is details-on-demand, in which the user desires to retrieve abstract information related to a given element in the VE space.
- **Abstract Information to Spatial Information:** In this case, the user wishes to proceed from abstract to VE. This can enable the user to control the VE through abstract information. For example, by selecting data in a separate abstract display, users could highlight desired objects in the VE, filter uninteresting objects, or automatically travel to a related location.

In order to test how our IRVE layout techniques impact usability for search and comparison, we subdivide the tasks by their task-information mapping. For our purposes, the task-information mappings are denoted by the following convention:

[IRVE_TaskType : informationCriteria -> informationTarget]

IRVE Search Tasks [S:*] require subjects to either:

- Find a piece of abstract information (A) based on some perceptual/spatial criteria (S). Task example [S:S->A]: 'What molecule is just outside of the nucleolus?'

- Find a piece of perceptual/spatial information (S) based on some abstract criteria (A). Task example [S:A->S]: 'Where in the cell is the Pyruvic Acid molecule?'

IRVE Comparison Tasks [C:*] require subjects to either:

- Determine an abstract attribute (S) after comparing by some spatial criteria (A). Task example [C:S->A]: 'Find the lysosome that is closest to a mitochondria. What is the melting point of the molecule in the lysosome?'
- Determine a spatial attribute (S) after comparing by some abstract criteria (A). Task example [C:A->S]: 'Where in the cell is the molecule with the lowest melting point?'

<table>
<tr><td>IRVE Activities</td><td>Search: query, goal-directed navigation

Exploration: browsing; opportunism; non-goal directed navigation</td><td>Comparison: describe the similarities and differences of two or more items

Patterns and trend finding: apprehend systems or rules manifest in the arrangements of objects and properties</td></tr>
<tr><td>Information Visualization
Task set
Shniederman [1996]</td><td>Overview: Gain an overview of the entire collection.
Zoom : Zoom in on items of interest
Filter: filter out uninteresting items.
Details-on-demand: Select an item or group and get details when needed.
History: Keep a history of actions to support undo, replay, and progressive refinement.
Extract: Allow extraction of sub-collections and of the query parameters.</td><td>Relate: View relationships among items</td></tr>
<tr><td>Virtual Environment
Task set
[Bowman et al 2004]</td><td colspan="2">Selection
Manipulation
Navigation</td></tr>
</table>

Table 3.1: IRVE activities overlayed on Information Visualization and Virtual Environment Tasks

3.3 IRVE Design Goals

In [Polys, 2004b; Polys, 2004c] we enumerated the scope of design challenges and options for the display of abstract information in desktop virtual environments and demonstrated an object-oriented approach to encapsulating a variety of display behaviors. The techniques we described address a number of fundamental challenges for information design across display locations. IRVE designers must tackle a number of visual design challenges. These include visibility, legibility, association, occlusion, aggregation and screen size. These challenges are non-trivial in that they relate and tradeoff with each other.

Visibility

Foremost, annotation panels should be *visible* to the user. This means that our first spatial layout consideration is the size of the annotation. If the annotation panel is object-fixed and the object is within the viewing frustum, the panel should not be located behind its referent object. Conversely, the annotation should not block the user's view of the referent object by being located directly in front of the object (between the user and the referent). One tradeoff along these lines arises in the case that the object is sufficiently large or near that it consumes the user's field of view. In such a case, the panel should at least not block the user's view of important features of the object. At a distance, the panel should be sufficiently large that it is noticeable, but not so large that it dominates the visual field and becomes perceived as the referent itself rather than an attribute of the object.

Legibility

A crucial consideration in the case of supplemental text or numeric information is *legibility*. If an annotation (such as text) is to be displayed and legible, it must be of sufficient size and clarity that users can read its letters and numbers. Font attributes (such as family, style, language, and anti-aliasing) and the variability of acuity in human vision can impact a design and its legibility and these are important considerations. In addition, there is the issue of sizing and layout behaviors for annotations. For example, in the case of object or world-fixed annotations, scaling of size can be a function of user proximity to the object. In the case of user or display-fixed annotations, legible font size may be a function of screen resolution.

Annotation panels that contain text, graphs, or images also have a natural 'up' direction. Since users may navigate by flying in 3D spaces and their orientation may not be constrained, object-fixed annotations should be true 3D Billboards. Another consideration for legibility is color and contrast. If the font color of a text annotation is the same as the environment background or its referent object (in the case of object-fixed), the characters may blend in with their background. One solution to this problem is to include a rectangular plane of a contrasting color behind the textual annotation. These background panels may be semi-transparent to minimize occlusion of other objects in the scene.

Association

Associating an annotation with its referent object is a crucial issue in Information-Rich Virtual Environments. Users must be able to perceive the referential relations of abstract and spatial information with minimal cognitive overhead. The laws of Gestalt perception (most recently summarized in [Ware, 2003] include in no order:

- Connectedness
- Proximity
- Common Fate
- Common Region
- Similarity
- Good Continuity
- Symmetry

The association may be depicted explicitly by way of a line between the panel and a point on the object (Connectedness). Relation may also be depicted implicitly in a number of ways. For example, the annotation being 'near enough' to the object that the relation is perceived (Proximity, Common Region), or the annotation being rendered is the same color scheme as its referent object (Similarity). Common

Fate refers to the principle that objects that move together in similar trajectories or change color together are related.

While Ware [2000] gives primacy to connectedness, there is little evidence concerning how Gestalt principles rank against each other across the range of information rich visualizations that are dynamic and include depth. A crucial challenge for depicting both implicit or explicit Associations is to insure that the relation can be perceived and understood from any perspective, even if the referent object is oddly shaped.

Occlusion

When considering the design of object and user space annotation panels, there is also the issue of *occlusion*. In dense or crowded scenes with a large number of annotation panels, users can be quickly overwhelmed or confused as annotations consume the visual space. Occlusion is a strong depth cue, but hinders visibility of information in the environment. In IRVE information displays, the challenges of Occlusion and Association are also related. Consider that the stronger the Gestalt association between object and annotations the more of the scene is occluded by the annotation. We term this problem the *'Association-Occlusion Tradeoff'*. Management of occlusion can be accomplished either by a centralized manager class that knows the image-plane size and the span of 3D object's 2D projection (e.g. [Bell, 2000; Bell, 2001]) or by a distributed ruleset (i.e. constraints) that gives rise to emergent behaviors such as flocking [Reynolds, 1987].

Aggregation

The content(s) of an annotation may be of a variety of data types (i.e. nominal, categorical, ordinal, numeric), a variety of data structures, and of a range of volumes. Thus, another important consideration in the design of IRVE annotations is the geometric and abstract levels of detail depicted at a given time. We refer to the informational hierarchy as the *level-of-aggregation* which may or may not correspond one-to-one with the referent object's geometric level-of-detail. As a user drills down, iteratively requesting more detailed attributes, the content and the size of the annotation may change. Successive annotation details may become visible implicitly as a function of user proximity or explicitly as a result of user action such as mouse-over or selection. If the annotation metadata is of a variety of media types, designers may need to introduce additional affordances to the annotation such as hyperlinked menus and display logic.

Desktop and Large Screen Displays

The body of research on Human Computer Interaction across display sizes is growing and effective use of screen space is a crucial consideration for interface designers. In IRVEs, the details of information display techniques are likely to be task specific. Indeed they might be specific to different display sizes as well - a technique that works well on a desktop may be unusable in a large screen context and vice versa.

The variety of content and applications on the web using standard formats such as VRML and X3D are prime examples of how additional information can be integrated and presented to the user outside the viewing frustum allocated to the 3D scene (e.g. Figure 3.1).

In display space contexts, where multiple external frames and windows are viable display venues for annotation information and supplemental views, it is especially important that designers establish a perceptual correspondence between objects in the 3D view and items in other areas of the screen real estate. In Gestalt terminology, this correspondence may be established by shared visual attributes such as color (similarity) or by implicit or explicit user actions (common fate, such as synchronized highlighting). For example, if a user navigates to a nearby object in the 3D scene and a text area in another part of the screen changes to show the object's description, there is a referential relation established and the user will expect this navigation action to have a similar effect with other objects. Such correspondence between information types can also be achieved through interaction such as selection. For example, the Brushing and Linking technique for multiple views information visualization [Ahlberg, 1995; North, 2001] can render the association through shared state based on user interaction.

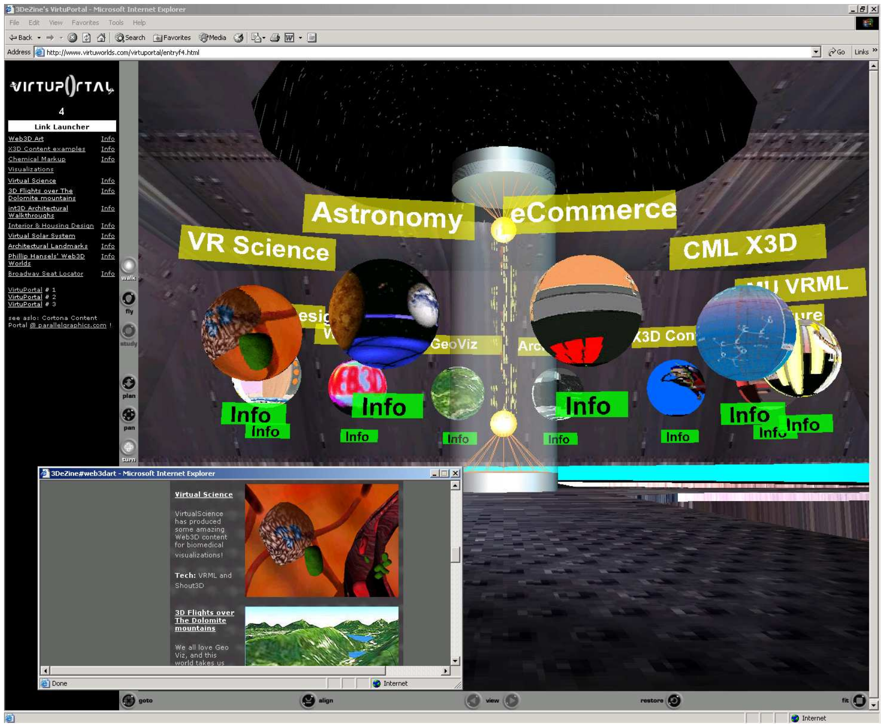


Figure 3.1: An IRVE Web Portal using frames and pop-up windows to manage virtual world content;this example shows Object space and Display space annotations

Web browser windows and embedded media objects (such as Web3D worlds) are usually sized in absolute pixels, while frames and tables can be sized by percentages or absolute pixels. Using web pages and hyperlinks, 3D, text, images, audio, and video resources can all be loaded into different windows and respond to interaction events. For VRML and X3D worlds embedded in a web page at a fixed size, the user perspective on the VE is specified by the `fieldOfView` field of the `ViewPoint` [Keller, 1993] node. This is a value in radians with the default value being π/4; if a VE is rendered at a fixed size, larger values create a fish-eye effect while smaller values create tunneled, telescoping effects. Naturally, with a larger *fieldOfView*, more of the VE is visible, but perspective can be distorted especially at the periphery. This is similar to the focus+context technique in information visualization, originally described by Furnas [1981, 1986]. Effectively managing screen size and projection distortion are important challenges for IRVE information design.

3.4 IRVE Design Space

Our theoretical framework includes a notion of the design space for IRVE information display techniques. In other words, what are the possible ways that abstract information should be included within a VE? This includes issues such as the type of abstract information (e.g. text, audio, graphics), where the information should appear (e.g. in a heads-up display, in a spatial location attached to an object), how links among information should be represented, and so on.

The IRVE information design problem focuses on the representation and layout of embedded abstract information in realistic virtual environments. There are many possible representations for a given dataset, such as a table, a plot or graph, a text description, or an equation. Placing this information (layout) within the 3D virtual environment is not a trivial issue. The displayed information needs to be visible for the user; it should not occlude important objects in the world; and it should not interfere with other displayed information. We made an initial proposal as to the dimensions of information design in IRVES [Bowman, 2003a]. This taxonomy was subsequently expanded in [Polys, 2004b; Polys, 2004c]. Our proposed design matrix is shown in Table 3.2 below.

Abstract information content in an IRVE may be a variety of media types such as text, numbers, images, audio, video, or hyperlinked resources. We can define this supplemental, enhancing information as *annotations* that refer to some perceptual data in the VE. Annotations may be simple labels, detailed attributes such as field-value pairs, 2D or 3D graphs, or other multimedia. Annotations may be associated with objects in the environment, the environment itself (or locations in the environment), or a temporal event in the environment. Annotations may be rendered as a result of implicit user action such as navigating closer to an object (proximity-based filtering), turning to examine the object (visibility-based filtering), or explicit user action such as selecting an object for details-on-demand.

Abstract information design parameter	Psychological process	Usability impact
Visual attributes: - color - fonts - size - background - transparency	**Perception**	- Legibility - Readability - Occlusion
Layout attributes: - layout space - association	**Interpretation, Feature-Binding**	- Relating abstract and perceptual information - Conceptual categories & abstractions - Occlusion
Aggregation: - level of information detail - type of visual representation	**Making Sense**	- Comparison & Pattern Recognition - Effectiveness - Satisfaction

Table 3.2: Updated IRVE design matrix for abstract information display

Based on prior work and the existing models of human information processing, we classify abstract information design attributes based on three aspects: *visual attributes*, *layout attributes,* and *level of aggregation*.

Visual Attributes

Watzman [Watzman, 2003] has examined usability guidelines and visual design principles as they relate to text typography and color usage. Text typography is the smallest definable part of a visual design and consists of typefaces, letterforms and sizing as well as readability and legibility factors. These include: contrast, combinations, and complexity; word spacing & justification, line spacing; highlighting with bold, italic, caps; and decorative type, color and background. Watzman details the relation of principles such as harmony, balance, and simplicity to text legibility and readability. These attributes determine the fundamental level of perceptual issues to be considered. They include ***font type, font size, color, brightness*****, and** ***transparency***. These factors can significantly affect the readability and legibility of the text information, and IRVE display components must be customizable in this dimension.

Layout Attributes

There is a wide range of possibilities for how annotations can be arranged in an IRVE. In our work, we introduced the dimension of **Layout Space** as the parameter for *where* an annotation is located. We have enumerated at least five possibilities for an annotation's location based on the coordinate system it is resident in: Object, World, Viewport, User, and Display space (defined above in Section 1.5 and shown in Figure 3.2). Layout spaces are distinguished by the annotation's location in the scene's transformation graph; annotation layout techniques therefore operate within these various coordinate systems.

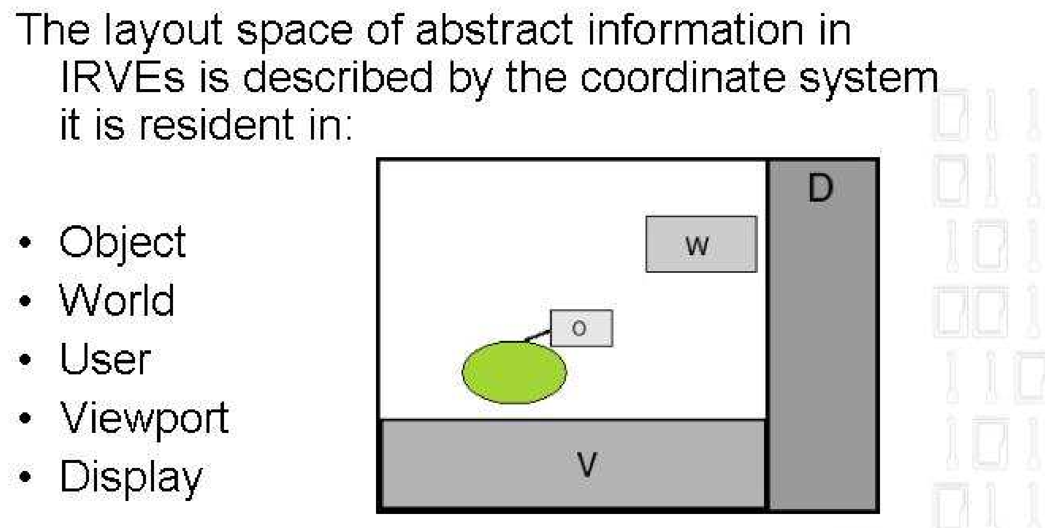


Figure 3.2: IRVE Layout Spaces, a schematic view

There are important usability implications arising from an annotation's layout space. For example, if an annotation is in the same coordinate system as its referent object (Object space), the annotation will move when the object is moved. In Object or World space, users may have to navigate through the spatial environment to access the abstract information contained in an annotation. In contrast, in a User or Viewport space, the annotation travels with the user and little spatial navigation is required to access the information. As a distinction of location in the IRVE, different layout spaces may support a number of depth cues consistent with their referent object. The consistency of depth cues between annotation and referent may have a direct impact on how easily and accurately users perceive the relations between information types.

Another layout parameter is **Association**, which defines the visual groupings of abstract and spatial information in the image plane. The varieties of configurations are derived from Gestalt principles, which are well-established for 2D stimuli. These principles include: Similarity, Connectedness, Proximity, Common Region, and Common Fate. The more Gestalt cues between annotation and referent may also have a direct impact on how easily and accurately users perceive the relations between information types. While Gestalt principles may describe effective 2D configurations for grouping, little work has been done on how the principles interact with depth cues present in 3D environments and the relative strengths of these cues.

Layout attributes determine how users parse and interpret the IRVE space - binding and relating the different information types. This research examines the impacts of Layout space and Associations on user performance. Through this research we seek to understand the interplay of spatial navigation, depth cues, and Gestalt association cues in IRVEs. For example, how do designers balance the desire for strong association with the desire to reduce occlusion and clutter in the scene?

Aggregation of Encoding

The last parameter we identify is that of **Aggregation**, which consists of two kinds: first, the level of detail represented by the annotation and second, the nature of its representation- the type of visualization (scatterplot, bargraph, textual, etc.). Level of detail in this usage is not related to the perceptual display of an annotation, but rather the semantic content of the annotation. It refers to the level of detailed information provided by the annotation- how much the abstract information is aggregated.

The choice of semantic detail presented can affect what users recall and what they distinguish as similar or like kinds. For instance, a highly aggregated text label (describing a lexical category or kind) for an object may allow a sparse layout and enable efficient exploration, but may result in poor user performance for classification and problem solving tasks. In a separated approach, instance details can be quickly comprehended, but lead to crowding and layout density problems. Aggregation of encoding is also subject to changing over time as user tasks evolve.

3.5 IRVE Display Components

There are two principal approaches to implementing IRVEs. The first is to embed the abstract information displays in the virtual environment application. The second is to link the virtual environment to other applications, which are responsible for the display of abstract information. We have implemented IRVE display components for both approaches and formalized them in an XML DTD and Schema. This work is described in the following sections.

In looking at the design space of IRVE display components and the combinations of spatial and abstract information, we recall Bederson's call for a new interface physics [1996]. We adapted the idea of semantic zooming and applied it to IRVEs in our notion of 'Semantic Objects'. Semantic zooming appeals to a user's knowledge about the real world in that objects appear and behave differently depending on the context and scale at which they are viewed. We pose Semantic Objects as the design abstraction to unify information visualizations and virtual environments. Semantic Objects respond to interaction events (such as navigation and selection) and can use a variety of display techniques to render information and relations.

We first described our embedded visualization components in [Bowman, 2003a] and then in more detail in [Polys, 2004a; Polys, 2004b; Polys, 2004c; Polys, 2004d; Polys, 2005a; Polys, 2005b; Polys, 2005c]. Our embedded IRVE display components were implemented in VRML and X3D, international standards for interactive virtual environments. We first published on linked IRVE components (with the Snap visualization system) in [Polys, 2004a] and this system was subsequently developed and evaluated (Section 4.1.2).

3.5.1 Embedded Visualization Components

In the embedded visualization approach, IRVE information is typically managed and rendered within one application- the information visualizations are rendered within the virtual environment. This is especially typical in immersive contexts where there is no desktop or windows metaphor. Therefore we developed a number of information visualization components and flavors of Semantic Objects that can render abstract information in Object, World, User, or Viewport space.

Annotations

First let us describe the set of design objects employed to render abstract information within a VE. We term these 'Annotations'. Annotations are a set of custom scenegraph objects that encapsulate information visualization displays and behaviors. For example, nominal, categorical or ordinal information can be rendered as textual and numeric labels. Quantitative information can be rendered with number labels or graphs. Other annotation types including information such as images, video, or audio can be registered to a spatial location, object or group of objects. We have developed annotations for image, video, text, and graph renderings of abstract information.

For annotations that include text and numeric views, we must address the legibility challenge and expose as much visual attribute functionality as possible for the environment author. This includes customizing font color, font family, line spacing, and justification, as well as panel color and transparency. The size of the annotation background panel (a 2D plane) is automatically computed according to the number of lines and the character width of the annotation. For textual and numeric information, we implemented two different panels for common situations: 'unstructured' panels and 'structured' (field-value pairs with a title), which are shown in the left and right of Figure 3.3 respectively. The data content of text and numeric annotations may be dynamically updated from events in the scenegraph.

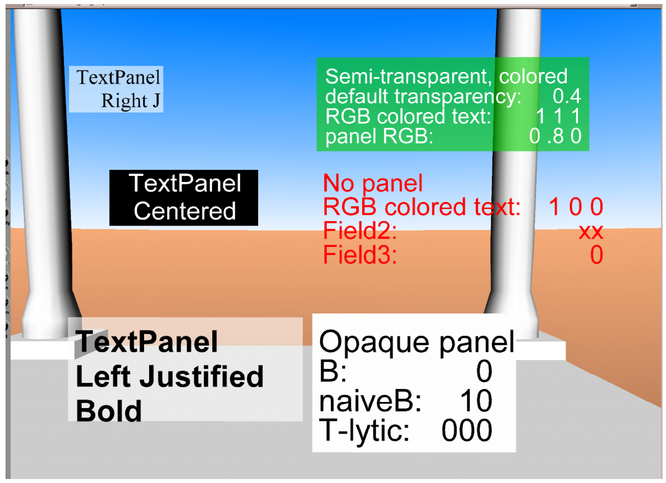


Figure 3.3: A variety of Visual Attribute parameters for text and number annotation panels

For quantitative and timeseries data, graphical representations using the techniques of information visualization may be required. Therefore we implemented Bar graph and Line graph annotations (left and right of Figure 3.4 respectively). These components expose basic visual attributes such as font color, font family, and panel color as well as allowing dynamic data values. Because annotations are encapsulated scenegraphs themselves, they can include their own manipulation and display logic. For example, our Line graph annotation can be scaled, moved or rotated using widgets built into the annotation. Similarly in audio or video annotations, clips may be started, paused, stopped, reset, etc.

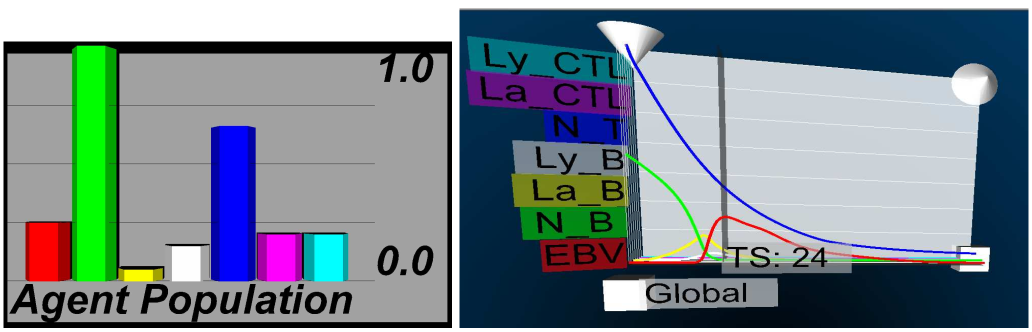


Figure 3.4: Example Bar graph and Line graph annotation components with PathSim data

Semantic Objects

We have encapsulated display and interaction behaviors in the definitions of custom scenegraph objects and implemented a range of design options and layout techniques for the display of abstract information. We call these conceptual and programmatic abstractions 'Semantic Objects'. In embedded IRVE approaches, Semantic Objects are defined with their geometric and appearance information and their related abstract information (annotations). Layout and Association behaviors are encapsulated in the definition of Semantic Objects, which are parameterized for various solutions to the visibility, legibility, association, occlusion, and aggregation challenges mentioned in the previous section.

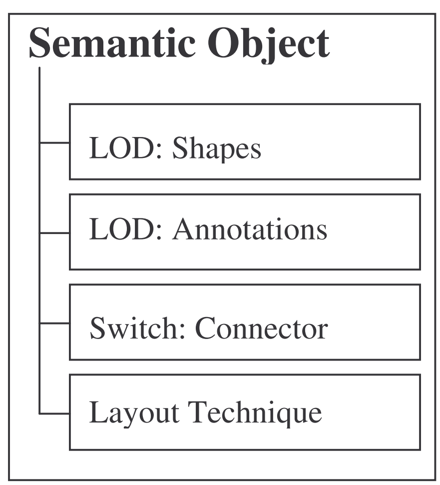


Figure 3.5: Encapsulating IRVE information display behaviors in a prototypical Semantic Object

Semantic Objects in embedded IRVEs contain logic for displaying both spatial and abstract information in response to user interaction events such as navigation and selection. For example Semantic Objects can contain behaviors such as rendering annotations on selection, hoverOver, by proximity, etc. The display logic also includes layout algorithms and graphical elements such as connector lines. Connector graphics may be colored and drawn as lines or as polygons with transparency.

Semantic Objects maintain two sets of ordered children: one for the object shapes and appearances and one for the object's annotations. They also maintain two lists of ranges (distance to user) that specify which child (level-of-detail and level-of-aggregation) is rendered at a given distance. Thus, authors can choose to aggregate abstract information when the user is far away and show progressively more detail as they approach the object. A prototypical structure of a Semantic Object is shown in Figure 3.5.

Layout Techniques

IRVE layout techniques seek to find design solutions to the challenges we described above (visibility, legibility, association, occlusion, and aggregation). In the course of this research, we developed a set of Semantic Object flavors that implement a variety of techniques. These techniques cover the IRVE design dimensions of Layout space and Association. Each of the flavors addresses the association–occlusion. In addition, they can address the legibility–relative size tradeoff in different ways.

In **Object Space**, we designed and implemented five different layout techniques: Relative Location, Relative Rotation, Bounding Box, Screen Bounds, and Force-directed. These techniques vary in the means by which annotations are located in Object space – for example by properties of the viewing angle, the object shape, and the proximity of nearby objects.

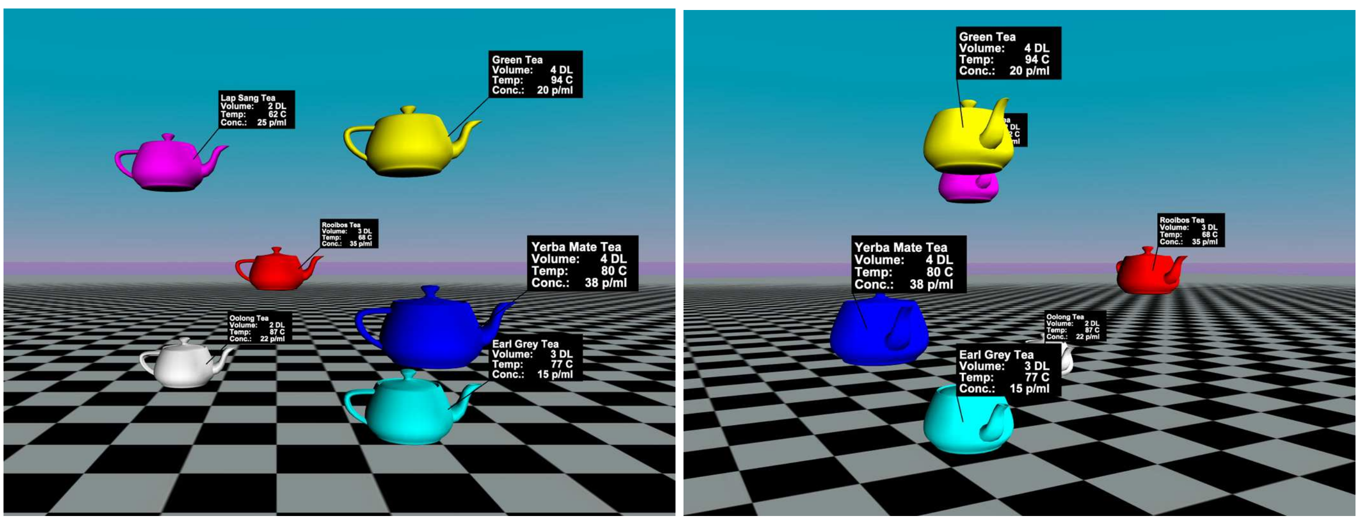


Figure 3.6: Object space layout: Fixed Position

The first of these, '*Fixed Position*', is the base case where an annotation is located in a fixed x,y, and z position relative to its referent object. This technique is useful for situations where a specific, static location is required for the annotation. The downside is that from certain viewing positions, the annotation may occlude the referent object or vice versa (Figure 3.6).

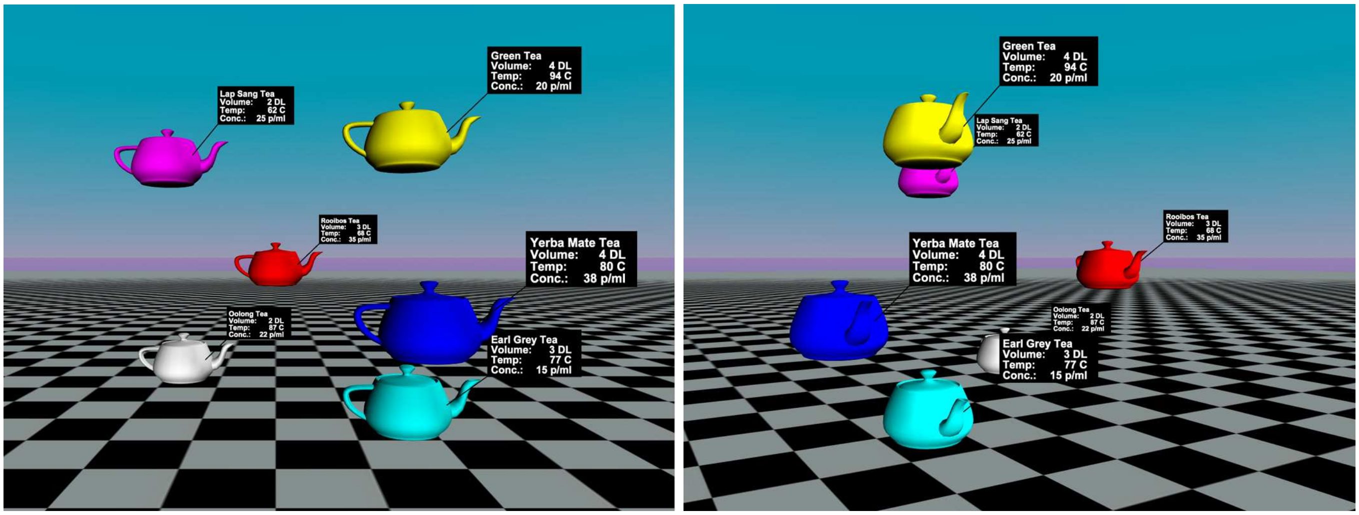


Figure 3.7: Object space layout: Relative Position

The second we call '*Relative Position'* is similar in that the IRVE author specifies the annotation's position relative to the referent object. However this flavor is a dynamic layout. As the user navigates around the object, the annotation and connector rotate to maintain the relative position orthogonal to the user's perspective. Like the VRML/X3D Billboard node, an axisOfRotation can be specified, which constrains the annotation's rotation to a specific axis. The default value is 0 0 0, which enables a screen-aligned billboard. Figure 3.7 shows an example of this technique.

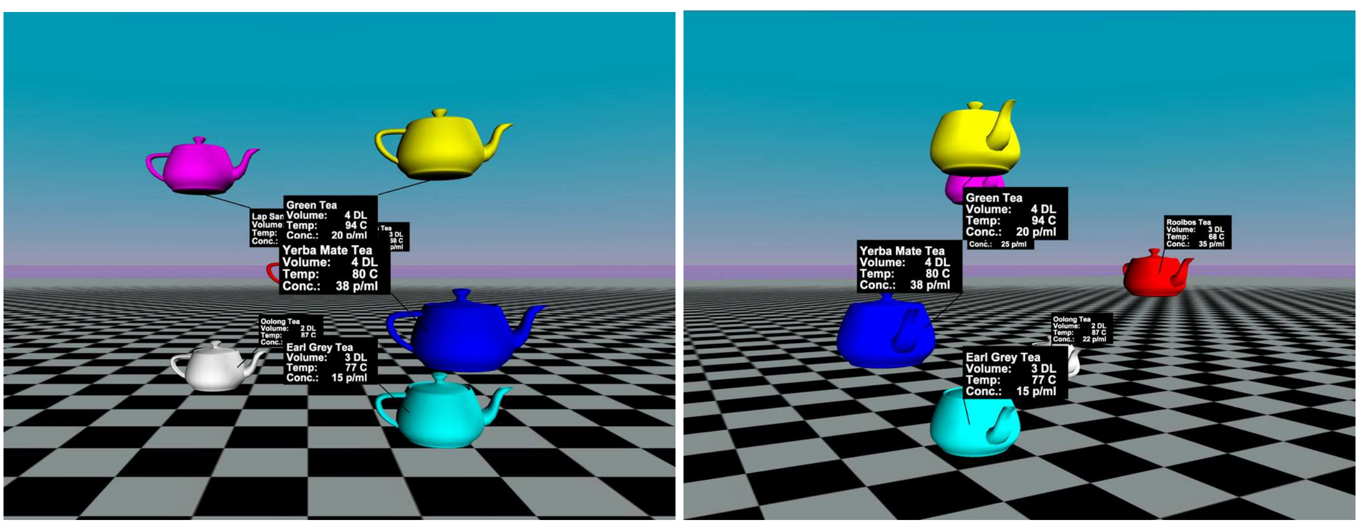


Figure 3.8: Object space layout: Bounding Box technique;

The third spatial layout technique for Object space association we call the '*Bounding Box'* method. This is another dynamic method. In the Bounding Box technique, the IRVE author may specify a series of 8 (x, y, z) coordinates that define a 3D bounding prism containing the referent object. The annotation snaps to the corner of the box nearest the user and shifts its offset accordingly. Figure 3.8 shows an example of this technique.

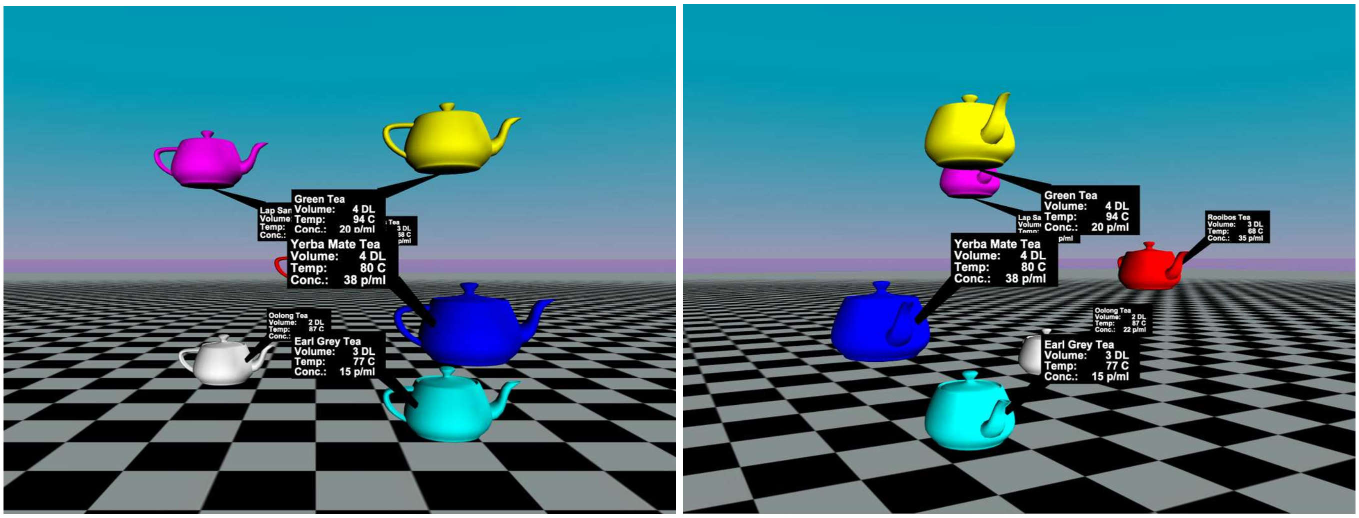


Figure 3.9: Object space layout: Screen Bounds

The '*Screen Bounds*' technique is intended to reduce occlusion and maintain association between the annotation and its referent. For each SemanticObject, 4 points are defined as the referent object's screen-aligned (2D) bounds. The annotation will snap to a different corner of the bounds and the label will shift appropriately depending on the viewing angle of the user. Figure 3.9 shows an example of this technique.

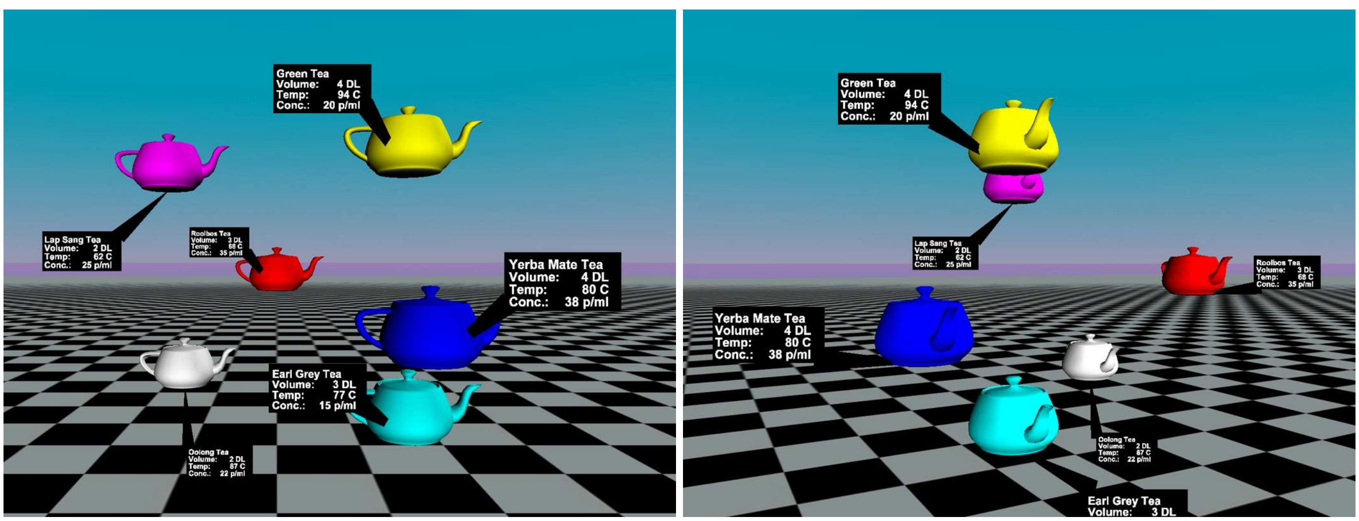


Figure 3.10: Object space layout: Force-directed

The '*Force-Directed*' layout technique is intended to reduce need for the scene designer to explicitly manage the location of annotations in Object Space for occlusion. The algorithm attempts to minimize occlusion in the scene by creating a repulsion constraint between other annotations and objects. The algorithm projects obstacles' force to a screen-aligned bounding circle, and moves the label along the circle away from the obstacles. This vector that is projected to the screen-aligned circle represents a summative repulsion force from other objects and annotations in the scene. The Force-Directed layout maintains the Gestalt association cue of Proximity, but not the same discrete, deterministic spatial relation as the Bounding Box or Screen Bounds technique. This technique results in emergent layout behavior. Figure 3.10 shows an example of this technique.

Our Object-Space flavors include annotations that can also be scaled according to user distance. This can be useful for Annotations that must be legible from a distance. There are three modes of annotation scaling: None, Periodic, and Continuous. No scaling means that the annotation's size is fixed, therefore the user may be required to navigate (spatially) to a legible distance. Periodic scaling provides intervals of scaling according to the user's distance. Continuous scaling scales annotations by some factor for all distances. The tradeoff with this scaling parameter is that to guarantee legibility from a distance, one must remove the depth cue of Relative Size. In addition when far-away annotations are scaled up, they can add occlusion to the scene.

In **Viewport Space**, we implemented a generic Heads-Up-Display with an information display area, an ordinal HUD BorderLayout, and a proximal HUD BorderLayout.

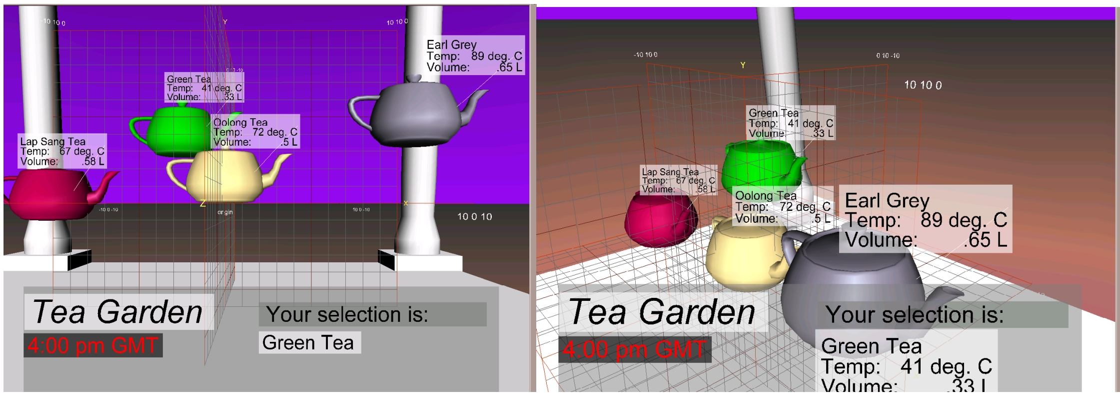


Figure 3.11: Layout of Annotation information on a generic HUD: Semantic Object annotations are displayed by mouse-over (left) and by selection (right)

Our 'G*eneric HUD'* prototype object can take arbitrary sensor and geometry nodes such as text, image, or graph panels as children. Because these nodes are instantiated in the scenegraph, it is trivial to route events to objects in the HUD space and vice versa. This event interaction is crucial to establishing correspondence relations between scene objects and their annotation information through implicit or explicit user interaction (common fate). Figure 3.11 shows an example of our HUD object in use.

In the '*BorderLayout'* [Polys, 2005c], a selected object's label is toggled into a Heads-Up-Display (HUD) in Viewport Space where it is always visible regardless of the user's position and viewing orientation. In the software definition of our interface, we maintain a pixel-agnostic stance and scale and locate labels according to parameters of the environment's projection (rendering). Labels are sized and located in world units relative to the specified Software Field of View (SFOV) and the distance to the near-clipping plane. The HUD is set so that a 20 x 20 grid is located at the image plane with the shortest window dimension is 20 world units across.

The Viewport Space BorderLayout we defined can be specified with container capacity and the fill order for the four directions using the BorderLayoutManager. The location of any given label is determined by the order in which it was selected. If all annotations are visible at load time, the order is determined by the node's realization time in the scenegraph (typically reverse lexical). In the BorderLayout technique, labels can also be connected to their referent objects with lines extending into the scene. The layout of labels in the 2D Viewport space is managed by a parameterized BorderLayoutManager script.

Like its Java Swing inspiration, the BorderLayoutManager divides the layout space into four regions or containers: North and South, which tile horizontally across the top and bottom, and East and West, which tile vertically on the right and left sides. Figure 3.12 shows an example of the Viewport Space layout technique used in a 3D cell environment. In this example, all containers have been filled. If there are more annotations than Viewport locations, the fill order is wrapped. If an annotation is occluding something in the scene or if users want to reposition an annotation, they can click and drag it to a new location in the 2D Viewport Workspace.

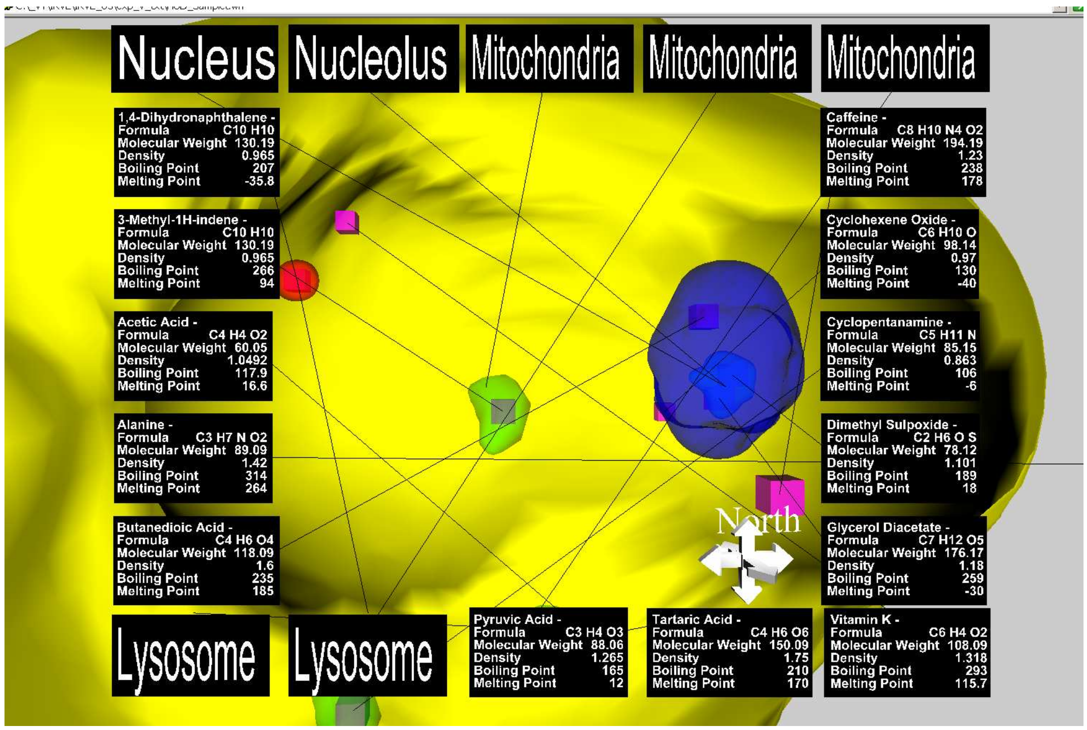


Figure 3.12: Layout of Annotations in the BorderLayout HUD; fill order: N, S, W, E

When rendering an IRVE scene on a large display, the labels and layout scale up proportionately, also becoming larger. Because our Viewport space implementation does not specify label size or location in pixels, the layout is easily adaptable to different screen sizes. Using our Viewport Space approach for example, we can change the label scale and container capacity to display the annotations at an equivalent pixel size as on the single-screen. So using a nine-screen display (3x3) and holding pixel size constant to the value on a single-screen, we can get approximately 3 times as many labels in one container.

We have also implemented a version of the HUD BorderLayout that locates annotations in a container slot closest to the referent object's screen projection. This technique seeks to improve the proximity relationship between the annotation in the HUD and the object in the scene.

3.5.2 Federated Visualization Applications

For Linked Visualizations, developers need to go outside the VE scenegraph and coordinate visualization behaviors with other applications. This is especially common in desktop IRVES where applications are prolific and often babelized. Currently in WIMP architectures, windows are instantiated in display space and developers must formalize messaging protocols or directly integrate with GUI toolkit APIs such as Swing, MFC, or QT. In common Display Space layouts, information visualizations are given separate screen space where there is minimal occlusion, but association must be accomplished through similarity of visual attributes or by temporal associations (as a result of Brushing-and-Linking interactions, for example).

The multiple-views approach may be advantageous because there may be a variety of abstract information types related to a given perceptual data item. For each of these types, there are already effective 2D visualization techniques that we want to use. The Snap event and component architecture fills this requirement nicely, since multiple multiform visualizations can be displayed and coordinated.

Snap is a web-based interface for creating customized, coordinated, multiple-view visualizations. Through web pages and Java applets, Snap provides users with the ability to build layouts of multiple visualizations of data in a database with components such as tables, scatter plots, and various charts and graphs. Users can interactively combine visualization components and specify coordinations between the components for selection, navigation, or re-querying. Using the concept of a 'visualization schema', Snap allows users to coordinate visualizations in ways unforeseen by the original developers. We decided to use Snap because of this end-user flexibility and the ability for developers to define new visualization components.

In [Polys, 2004a; Polys, 2004c] we described our Snap2Diverse system which addresses how the display space of IRVEs can be used for coordinated multiple view techniques. We wanted to identify usability issues in using multiple linked views of spatial and abstract data in IRVEs. We developed a messaging and rendering framework for coordinated multiple views in immersive environments (Figure 3.13). While the framework is applicable to a variety of data domains such as medicine, engineering, and architecture, we have demonstrated it with a set of Chemical Markup Language (CML) data because it exemplifies the combination of spatial and abstract information types (i.e. Figure 1.3, 3.14).

Snap2Diverse uses one wall of the CAVE to render the Information Visualization application (in this case Snap) and exchanges interaction events with the Diverse Toolkit running basic X3D models in OpenGL Performer on the remaining two walls and the floor. The two applications communicate with a simple messaging protocol based on Snap events, which includes: the event (such as 'load', 'select') and the list of data identifiers who are the target of the event. Relations between the visualizations was accomplished via Brushing-and-Linking, which highlights spatial objects and their related information in the other Snap components upon selection (Figure 3.14). The integrated and immersive nature of this system provided us an important foundation to explore how heterogeneous visualizations can be combined both from a technical and an HCI perspective.

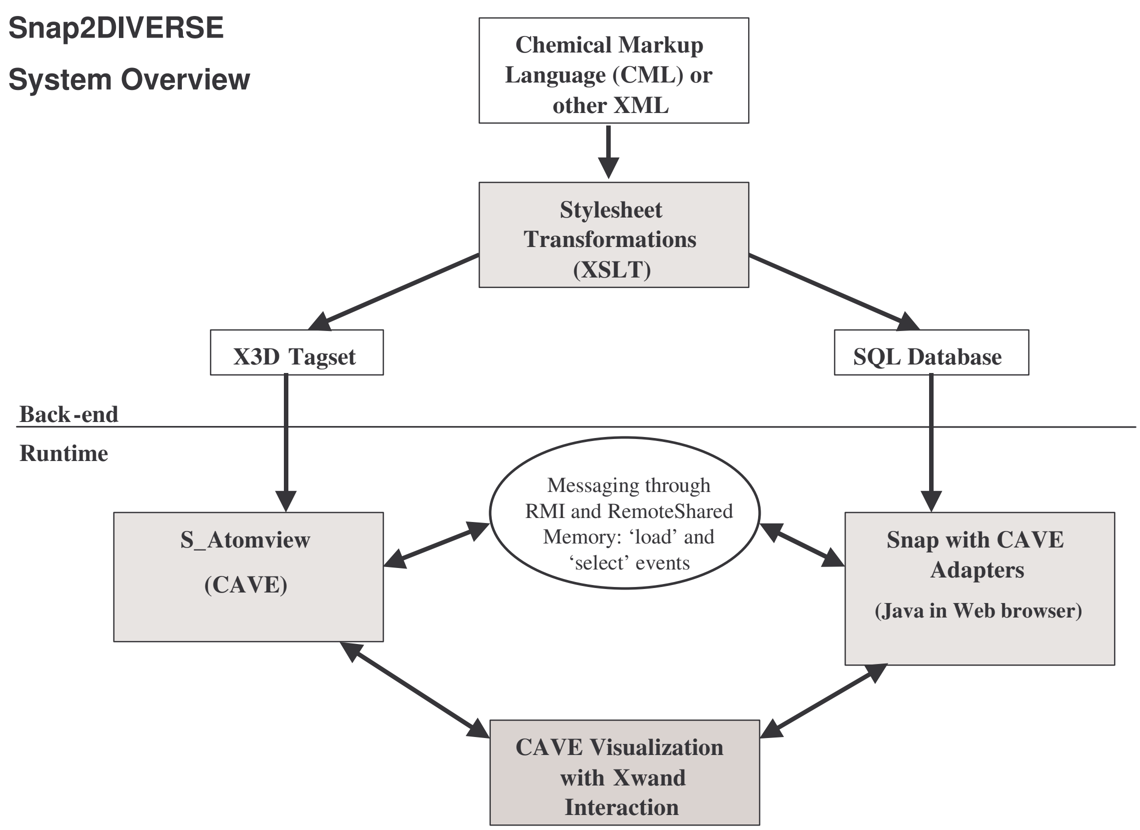


Figure 3.13: Snap2Diverse System Architecture (from Polys et al 2004a)

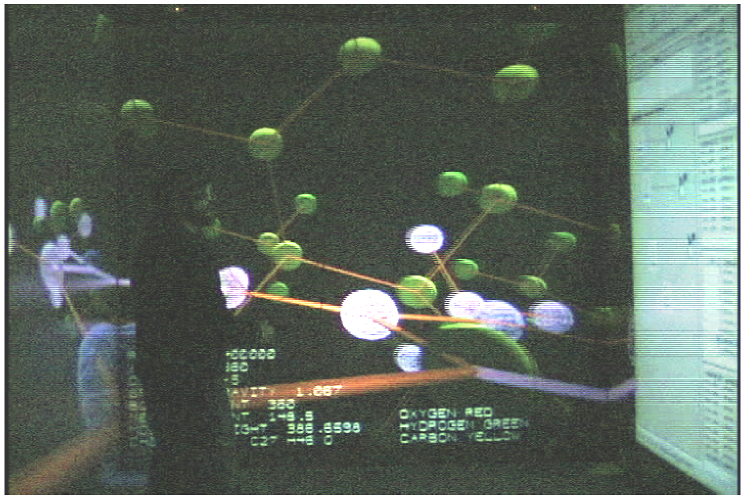


Figure 3.14: Snap2Diverse in the VT CAVE; the user is inspecting Carbon atoms

Although Snap2Diverse could be run in an immersive system or on a desktop with 2D and 3D views in separate windows, it required messaging through UNIX servers and remote shared memory, which limits widespread applicability. Therefore we ported the VE runtime system of Snap2Diverse to a Java Applet by implementing a Snap component using Xj3D [Yumetech, 2005]. Xj3D is an Open-source loader and runtime for X3D and VRML environments. With the full language support of X3D and VRML, we are able to use our Semantic Object abstraction as generators and consumers of interaction events and connect those events to external applications such as Snap.

For example, we added input fields to and logic to Semantic Objects to consume 'load' and 'select' events and generate 'select' events. Semantic Objects flavors could implement multiple responses to events including minimizing or maximizing the annotation, or switching the object's shape to another 'highlighted' shape. The implementation of the IRVE browser is shown in Figure 3.15.

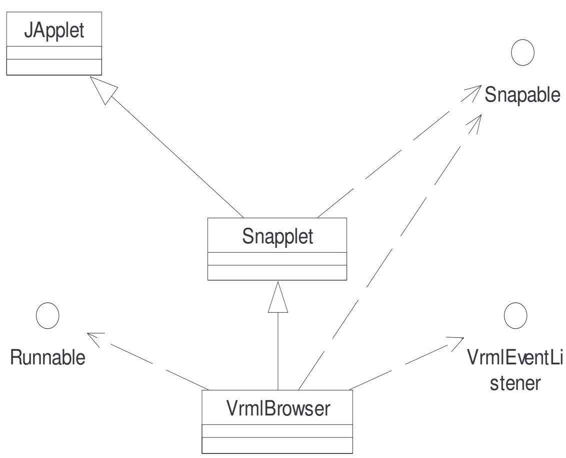


Figure 3.15: The inheritance and implementation of the Xj3D Snap component

In designing linked IRVE applications such as Snap2Diverse or the SnapXj3D component, we note two main requirements: first that there are consistent data identifiers (unique IDs) between the different applications (if the data source is not shared) and second that the different applications are able to generate and consume events in a common format. Because of the lack of occlusion, Display space layouts and coordinated views would seem to provide excellent access to the spatial and abstract information individually. The questions remaining are how these systems support users in understanding the relationships between the information types, and how much switching context between the views impacts performance.

4. Information Architectures

4.1 Publishing Paradigms

As the demands of data and user tasks evolve and expand, the field of Information Visualization presents many challenges for designers and systems developers. Of primary concern is the mapping of data records and attributes to a visual presentation that enables the user to detect patterns and relationships within the data. The goal of this mapping is to minimize the user's cognitive requirements for understanding and insight into the nature of the data that may not be apparent from viewing it in its raw form. The mapping of data to a visualization must take into account the data's volume and types and this chapter will discuss some approaches to this display problem. However, static presentations are limiting in their power to inform because the data and mappings cannot be interactively explored or rearranged. Computer-based visualizations can address this problem because users can now have control over the selection of data records, the encoding of those records as visual markers, and the presentation of those markers in a 2D screen or a 3D world. In this chapter, we will use prior work [Polys, 2005b] to examine how data may be mapped to interactive 3D worlds that may be published and distributed over the World Wide Web (WWW).

In the early days of web publishing, repurposing data content for multiple formats and platforms was expensive and as a result, a majority of useful information was locked into technology 'silos' for a particular delivery format, method, and platform. International standards organizations serve the computing community by developing and specifying open platforms for digital data exchange. By adhering to industry standards, organizations can lower their software and data integration costs, maximize their data re-use, while guaranteeing reliability and user access beyond market and political vagaries. Extensible Markup Language (XML) and Extensible 3D (X3D) are two examples of such standards and are covered in this volume. This chapter provides an overview of issues, strategies, and technologies used for publishing IRVEs with XML and X3D.

4.1.1 File Formats and the Identity Paradigm

Initially, the majority of published information on the World Wide Web was in a format called HyperText Markup Language (HTML). HTML was revolutionary in that it specified a declarative language for sharing documents (web pages) across a network. The resulting boom to multiple millions of web pages at the time of this writing is largely due to the simplicity and portability of this language. Information and images can be easily layed-out, linked, and accessed from all over the globe. If the author knows the HTML content header and tags, a basic document can be produced with a text editor and an image editing program. A document's headings, layouts, images, links, colors, and fonts are all described with HTML tags. More complex or innovative layouts require the use of <table> tags, which are difficult to manage without authoring software.

One major drawback of HTML is that its tags are strictly specified and overloaded. Tags in an HTML document represent both the informational content *and* the presentation of that content; that is, the data and the display information are included in the same file, often in the same tags. This limitation makes HTML tags less attractive as a data storage medium since it is difficult to repurpose data to other formats and applications. For example, if a customer's name and order number are enclosed by separate header tags, such as <h1>, there is no way to distinguish which information is the name and which information is the number from the tags in the file alone. Cascading Stylesheets (CSS) attempts to separate content and presentation in HTML by allowing the author to specify classes of tags with defined display attributes such as font, color, fill, and border. CSS provides flexibility by allowing definitions to reside within document files or as remote resources. CSS is useful for presenting the same page with different styles. However, this flexibility is not really a qualitative improvement in the language because the tagset is mostly unchanged and still finite in its descriptive power for data.

Virtual Reality Modeling Language (VRML) is an international standard (ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2002), but was designed as a portable format for describing and delivering interactive 3D worlds. The VRML standard is similar to HTML in that it is declarative, strictly specified, and carries both data and display information. In contrast to an HTML page, the VRML scene contains: spatial viewpoint and navigation information, 3D geometry with colors, transparency, and textures, text, fonts, links, backgrounds, as well as temporal information such as object animations and behaviors (defined in Interpolators, Sensors, and Scripts). Also, in contrast to HTML, VRML authors have the ability to define their own node types through the PROTO(type) node. The PROTO definitions can reside within the document file or as remote resources.

In VRML and X3D, nodes are analogous to element tags and fields are analogous to element attributes. Nodes are instantiated in a directed, a-cyclic graph called the scenegraph. A VRML file describes an scenegraph of interactive objects in space which the user can see and navigate through. Colored and textured objects are manifested in the world (the scene), animated, and visualized from a viewpoint or camera. When discussing Web3D media, I will refer to the 'viewpoint' as the Viewpoint node itself and the 'camera' as the rendered result of the Viewpoint via any superceding transformations. Similarly, 'navigation' refers to the scale and nature of the user's control over their Viewpoint (by way of the values bound in the active NavigationInfo node).

Early ease-of-authoring was complicated by lack of browser compliance to the standards and scripting support for Javascript (now officially 'ECMAScript') varied widely. In some cases, web publishers were forced to maintain multiple, browser-specific copies of their content in order to guarantee the widest possible accessibility. This amount of redundancy is expensive, even to percolate a small change across multiple website versions. Yet as the standards, client software, and server technologies have matured, HTML, VRML, or ECMAScript compliance is less the reason for maintaining multiple websites. Now, the motivation for permutable content is founded on the goal of customizing information for an audience or partner with a range of capabilities and interests. While HTML enabled the exponential growth of the web, it also required organizations to grapple with content management and personalization issues. The result was the design and deployment of web 'application servers' and web 'portals'. We will examine how these architectures currently apply to web publishing, and then to Web3D content, specifically X3D and VRML.

Hypertext Markup Language was originally designed to describe and deliver hypertext documents over the web. Virtual Reality Modeling Language was originally designed to describe and deliver interactive 3D worlds over the web. They are consequently unable to describe much else. Each is really only suitable as a web publishing format, not as the formats for content storage, archiving, and exchange. If pages are authored and maintained in a specific format (such as HTML or VRML) and the content is also delivered in that format (HTML or VRML), we can characterize the architecture as conforming to the '**Identity Paradigm**'- the source is identical to the deliverable. As mentioned above, this presents some problems both with maintaining a large set of documents and with re-using the documents' information in other contexts. Due to the limitations and expenses of this methodology, there was an immediate demand for other solutions. XML and X3D were designed to meet this demand.

4.1.2 Server Technologies and the Composition Paradigm

In recent years, a number of alternatives were provided by web server technologies and scripting languages to address these issues of maintaining static Identity Paradigm archives. Some well-known technologies include: Server Side Includes (SSI), Perl, Hypertext Preprocessor (PHP), and Java Server Pages (JSP). These technologies do have significant differences, but the common denominator is that they enable the composition and delivery of a document 'on-the-fly' in response to a user request. For example, when a user requests a page through the Hypertext Transfer Protocol (HTTP), SSI can get the current date and time from the web server and display it in the delivered page. SSI can also insert markup fragments into a document. This allows different documents to include consistent display objects (such as headers, menus, tables, and footers in 2D HTML and Heads-Up-Displays (HUDs) in 3D), reducing redundant content across multiple documents.

Scripting languages add another level of capability since they can connect and query online databases to recover information for display. For example, the user requests an online data set, and the server script queries a database and writes it into the delivered (result) document. These solutions can all be classified as supporting the '**Composition Paradigm'** where documents are dynamically generated from one or more data sources. The Composition Paradigm brings more flexibility to web publishing as developers can define common elements in a single location, pull data from multiple sources, and combine them according to a user's request. As a result, dynamic web sites are now commonplace.

One crucial issue in web publishing that relates to the Composition Paradigm is the notion of Content-type headers, or MIME types. MIME stands for Multipart Internet Mail Extension and was originally designed to distinguish files in email attachments. The MIME type tells the client what kind of data is contained in the file so that the client can decode and handle it appropriately. For files on a local machine, this delegation can be accomplished simply by the file extension. In a web server context however, the MIME type is sent first as a single line and does not appear in the document source. Each web server is configured to associate a document MIME type with a file extension and deliver it to the client. Web browsers or client operating systems also maintain such a list which determines what plug-in or application will display the content. So every file on the web has a content header that declares what kind of file it is.

File Format	Content type / filename extension
Text	text/plain
HTML	text/html
VRML V2.0	model/vrml
XML	text/xml
VRML V3.0:	
X3D (Classic encoding)	model/x3d+vrml .x3dv and .x3dvz
X3D (XML encoding)	model/x3d+xml .x3d and .x3dz
X3D (Binary encoding)	model/x3d+binary .x3db

Table 4.1 Principle filename extensions and MIME content types discussed in this chapter

Table 4.1 shows the relevant content types treated in this chapter. X3D (VRML V3.0) is a new standard for creating real-time 3D content. The specification of X3D's Architecture and API are as ISO/IEC FCD 19775:200x. Using the VRML97 specification as its starting point, X3D is cross-platform and hardware independent. It adds a number of new features such as XML integration, multi-texturing, NURBS, and a new scripting API. The X3D 'Classic' encoding is a brace-and-bracket utf8 file encoding that looks like VRML94. The X3D 'XML' encoding uses XML tags and attributes. At the time of this writing, the X3D 'Binary' encoding is still under development, but suffice it to say that a given scenegraph may be equivalently expressed in any encoding. The encoding specification for X3D is ISO/IEC FCD 19776:200x.

In practice, composability is generally accomplished with 3 ingredients: a structured template(s) for the delivered document, an accessible data source, and a server technology such as SSI, PHP, JSP, or Perl to compose the template with the appropriate data. Document templates basically structure the delivered document. As we shall see in section 4.3.2, they are the skeletal form of the file, either implicit or explicit. Accessible data sources include both databases (i.e. SQL) and/or documents or fragments of documents. Server-side scripts manage the data sources and populate the template before it is sent to the user. The composed content is then delivered to the user with the appropriate MIME type.

On an Apache web server, you might want to compose an X3D or VRML scene with PHP, or Perl, or some other server-side scripting language, but still have user's 3D plug-ins recognize it when it is received. In the case of composing VRML or X3D Classic files, you could specify MIME types for a given folder by adding the line:

```
AddType application/x-httpd-php .php .wrl .x3dv
```

or analogous definition to the .htaccess file in the directory on the web server. This line configures the web server to treat .wrl and .x3dv files and .php file requests as PHP files. This way, the Hypertext Preprocessor (PHP) engine is invoked when it serves both types of files and downstream applications such as browser plugins will recognize VRML and X3D content composed from PHP scripts in that directory.

The Composition Paradigm introduced a new level of capability for publishing dynamic web content. In enabled web 'Portals' which refers to a single site that links and includes relevant information for a particular audience or domain. Portals are usually dynamic and customizable per individual user. Users can specify what information is included in what part of the layout, and what look-and-feel they prefer. In most cases, this kind of personalization system requires the user to either login to the site or grant permission to set a cookie on their machine. Once the user is identified by the system, personalized content can be dynamically generated and delivered. This includes delivering customized information content to a user who is logged in from a workstation, a VR system, or a mobile device, such as a PDA.

4.1.3 XML and the Pipeline Paradigm

The World Wide Web Consortium's (W3C) metalanguage codification of Extensible Markup Language (XML) has opened new and powerful opportunities for information visualization, as a host of structured data can now be transformed and/or repurposed for multiple presentation formats and interaction venues. XML is a textual format for the interchange of structured data between applications ([W3C, ; Kay, 2001; White, 2002]). The great advantage of XML is that it provides a structured data representation built for the purpose of separating content from presentation. This allows the advantage of manipulating and transforming content independently of its display. It also dramatically reduces development and maintenance costs by allowing easy integration of legacy data to a *single data representation which can be presented in multiple contexts or forms* depending on the needs of the viewer (a.k.a. the client). Publishers reduce the ratio of maintained source files to presentation venues as source data tends toward semantic markup.

Data becomes portable as multiple formats may be generated downstream according to application or user needs. Another important aspect of XML is the tools that it provides: the DTD and the Schema. The Document Type Definitions (DTD) defines 'valid' or 'legal' document structure according to the syntax and hierarchy of the language elements. The Schema specifies data types and allowable expressions for the language elements and their attributes- it is a primitive ontology describing the document language's semantics. Using any combination of these, *high-level markup* tags may be defined by application developers and integration managers. This allows customized and compliant content to be built by authors and domain specialists. These tags could describe prototyped user-interface elements, humanoid taxonomies, or geospatial representations. Developers describe the valid datamodel for their application using the DTD and Schema, share it over the web, and standardize it amongst their community.

XML can be as strict or as open as needed. Content, or fragments of content, can be 'well-formed' and still processed with most XML tools. Typically, data validation is at author time, but it can be done at serving, loading, or runtime if needed. Publishing advances using XML technologies can be characterized as the **Pipeline Paradigm** –information is stored in an XML format and transformed into a document or parts of a document for delivery. From an XML-compliant source document (or fragment), logical transformations (Extensible Style Sheet Transformations - XSLT) can be applied to convert the XML data and structure to another XML document or fragment. A series of such transformations may be applied ending with a presentation-layer transformation for a final delivery-target style, content-type integration, and display.

Numerous developer resources exist for the W3C's XSLT specification (Kay, 2001; White, 2002). However, a review of the typical XSL Transformation process is in order:

1. An XSLT engine parses the source XML document into a tree structure of elements
2. The XSLT engine transforms the XML document using pattern matching and template rules in the .xsl style-sheet
3. Template elements and attribute values replace matched element/attribute patterns in the source document to the result document.

The Web3D Consortium's next-generation successor to VRML is X3D. Like XML, which moves beyond just specifying a file format or a language like VRML or HTML, it is a set of objects and interfaces for interactive 3D Virtual Environments with defined bindings for multiple profiles and encodings collected under a standard API (Web3D, 2002; Walsh, 2001). Like VRML, the X3D specification describes the abstract performance of a directed, a-cyclic scenegraph for interactive 3D worlds. In addition, it takes advantage of recent graphics advancements such a MultiTexturing and information technology advancements such as XML. X3D can be encoded with an XML binding using DTDs and Schema (Web3D, 2002). The X3D Task Group has provided a DTD, Schema, an interactive editor, and a set of XSLT and conversion tools for working with X3D and VRML94. Using the XML encoding of X3D, authors can leverage all the benefits of XML and XML tools such as user-defined markup tags, XSLT, authoring environments, and server systems.

Additionally, rather than defining a monolithic standard, the X3D specification is modularized into components which make up 'Profiles'. Profiles are specific sets of functionality designed to address different applications - from simple geometry interchange or interaction for mobile devices and thin clients to the more full-blown capabilities of graphical workstations and immersive computing platforms. The notion of X3D Profiles is important for publishing visualizations and we will examine them in more detail in subsequent sections. X3D may be presented in a native X3D browser [Web3D], or transformed again and delivered to a VRML97 viewer.

4.1.4 Hybrid Paradigms

The last publishing paradigm we will describe is the **Hybrid Paradigm**. The Hybrid Paradigm combines the Pipeline and Composition paradigms. Data from various sources and transformational pipelines can be dynamically composed into a scene and delivered to the client machine. Apache Cocoon, and Perl with the Gnome XML libraries are two well-known examples of technologies that enable such a flexible scheme. Figure 4.1 shows the principal differences between the paradigms I have described in this section.

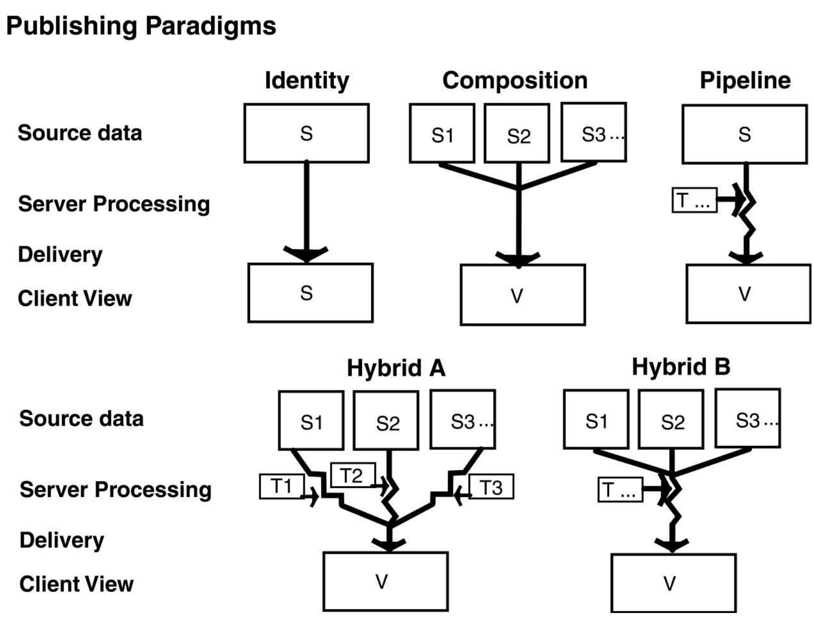


Figure 4.1 Publishing Paradigms Summarized: S = Source, V = View, T = Transformation

4.2 Design Principles and Interactive Strategies

Many challenges exist in the design of interactive 3D worlds and interfaces when integrating symbolic and perceptual information [Bolter, 1995]. Similar to efforts for 2D Visualization, researchers have experimented with the mapping of attributes to various visualization metaphors including the cone-tree, the city, and the building metaphor [Dos Santos, 2000]. They have shown that accurate characterization of the data is crucial to a successful 3D visualization, especially when the scenegraph is auto-generated. Bowman et al ([Bowman, 1998; Bowman, 1999; Bowman, 2003a]) have implemented and evaluated 'Information-Rich Virtual Environments' (IRVEs) with a number of features that are common to most Web3D information spaces. Information-rich virtual environments "…consist not only of three-dimensional graphics and other spatial data, but also include information of an abstract or symbolic nature that is related to the space," and that IRVEs "embed symbolic information within a realistic 3D environment" [Bowman, 1999]. This symbolic information could be attributes such as text and numbers, images, audio clips, and hyperlinks that are related to the space or the objects in the space. In this section, we will attempt to formalize an approach that is consistent with the capabilities of X3D.

Delivering arbitrary XML data to information visualizations with the Pipeline Paradigm requires both design and implementation considerations. As mentioned above, the generation of a 'data table' is the first step in the delivery of a visualization. The transformation of raw data to the data table may be accomplished by XSLT, or extracted by an XPath query or a query to a database. For the second phase of mapping - the data table to visual structures - we should remember from our definition that the abstract data in the table does not contain any inherently spatial information, thus it requires that the author determine the visual markers that will be employed. We will examine this step in more detail in section 4.4 especially as it relates to XSLT and X3D.

When designing 3D scenes for any purpose, a crucial step is that of 'Storyboarding' which helps authors specify what objects the scene contains and their appearance, from what points of view it can be perceived, and what kinds of interaction are appropriate at various points in time and space. When designing a usable visualization, Shneiderman's mantra [1996] of information design should ring in your head: "Overview first, zoom and filter, then details-on-demand".

4.2.1 Scene production process

Beginning with user requirements, a typical scene production process will follow these steps:

1. Define environment & locations
2. Define user interface & viewpoints
3. Define interactions
4. Organize declarative scenegraph
5. Model objects
6. Build Prototypes
7. Transform data and compose visual markers
8. Deliver to user

Steps 1 through 4 can be accomplished from the storyboard. Step 5 is typically done with a 3D modeling package that can export X3D or VRML. Steps 4 and 6 require at minimum a text editor and a developer familiar with the scenegraph capabilities of X3D or VRML. Steps 7 and 8 use server technologies and scripts to manifest the scene and deliver its final presentation form to the user.

4.2.2 Scene structure

Structured design in the case of X3D means dividing a scene into blocks which account for the various functional parts of the world. Using a modular structure to build a scene means that it may be built (composed) and managed from any number of applications or databases to the final target presentation. The result of this approach should be an implicitly structured X3D document template describing scenes in the form of:

Served Content type

- Header
- Scenegraph root
    - Custom node declarations: PROTO definitions and/or EXTERNPROTO references
    - Universe set (Backgrounds, global ProximitySensors)
    - HUD & User Interface
    - Scripts
    - World & Inhabitants set (lighting, geometry & objects)
    - ROUTEs

The X3D specification defines a set of standard nodes that can be instantiated in the scenegraph, what kinds of events they can send and receive, and where they can live in the scenegraph. The *transformation hierarchy* of a scenegraph describes the spatial relationship of rendering objects. The *behavior graph* of a scenegraph describes the connections between fields and the flow of events through the system. Events in the X3D scenegraph are called ROUTEs and exist between nodes. If nodes are uniquely named (DEFed), data events can be programmatically addressed and routed to that node. Custom logic and behaviors can be built into a scene with Script nodes which use ECMAScipt and/or Java to execute data type conversion, computation, and logic with events.

Designing scenes in modular blocks has additional benefits. For example, if the universe and HUD are kept consistent while the user navigates an information space, this helps to maintain the notion of presence when the world and its inhabitants change. Such runtime swapping of scenegraph branches (blocks) is possible with Browser API method calls in a Script node (see below, *section 4.4*).

A primary consideration in mapping data to a visual form is the range of values in the data. For quantitative and ordinal data, designers should examine the highest and lowest values in order to scale coordinates properly. For categorical data, the number of categories will determine the colors that can be employed. Since visual mappings must be comprehensible, axes, labels, and color legends should be instantiated. Designers may choose to put axes and labels in the universe block or the world block, depending on the design and compositional resources of their visualization application.

Custom Nodes

Authors can aggregate nodes and field interfaces into 'Prototype' nodes (PROTOs) which can be easily instantiated and reused in other scenegraphs and scenegraph locations. Prototypes allow the efficient definition, encapsulation, and re-use of interactive 3D objects. As we will see, Prototypes are especially suited to design visual markers and interactive widgets. In the interest of promoting the re-use of code without redundancy, Prototypes can also be defined in external files (EXTERNPROTOs). This prototype definition is a separate, singular resource that can be instantiated into multiple scenes.

One caveat to this abstract document structure is important to mention: the ability to use Prototypes (eg PROTOs and EXTERNPROTOs) to create user-defined objects and to use Scripts to define special behaviors (eg world or interaction logic), exist only in the "Immersive Profile" (and higher) of X3D which is analogous (but not identical) to the functionality enabled by VRML94. As we mentioned above, Profiles are specific sets of functionality designed to address different application domains [Web3D]. The "Interchange Profile" contains a node-set to describe simple geometries, materials, and textures for sharing between applications such as modeling tools. The "Interactive Profile" adds interpolator nodes for animation, sensors and event utilities for interactive behaviors, and a more capable lighting model. Additionally, on top of the Immersive Profile other software components may be defined and implemented. Currently specified components include Humanoid Animation (H-Anim), Geospatial 3D graphics (aka Geo-VRML), and Distributed Interactive Simulation (DIS). The "Full Profile" refers to full support for all components currently defined in the X3D specification. Authors should design to Profiles as they define what capabilities the client has – what nodes it can read and render.

Viewpoints and Navigation

An X3D scene defines objects in Euclidean coordinates, and animation interpolators generally proceed along linear time (although programmatic generation and manipulation of time values is possible with the

Script node). Virtual environment X3D scenes would not be visible or explorable without a way to describe user viewpoints and navigation. A key to understanding how this is accomplished with X3D (or VRML) is the idea of a runtime 'binding stack'. A binding stack is basically a list of 'bind-able' children nodes in the scene where the top node is active or 'bound'. The first Viewpoint and NavigationInfo nodes defined in a file are the first to be actively bound. Other Viewpoint and NavigationInfo nodes are made active by ROUTEing a boolean event of TRUE to their set_bind field. When this happens, the user's view and navigation function according to the field values of the newly bound node. Alternatively, events routed to the active node change the observed behavior of that node. For example, the Viewpoint node has fields for position, orientation, fieldOfView, and jump. The fieldOfView defines the user's viewing frustum and can thus be modulated to create fish-eye or telescoping effects. It is recommended to use a FALSE value for the jump field, as the user's view is then smoothly animated to that Viewpoint when it is bound, reducing disorientation [Bowman, Koller et al., 1997b].

Similarly, the NavigationInfo node carries fields that have a direct impact on the user's perception including avatarSize, speed, and type among others. For example, as a user navigates into smaller and smaller scales the avatarSize and speed fields should be proportionally scaled down as well. Specified X3D navigation types are**: "**WALK**", "**FLY**", "**EXAMINE**", "**LOOKAT", "ANY**",** and "NONE". While the first 5 types gives the user different ways of controlling their movement within the scene, in some cases it may be preferrable to use "NONE" in order to constrain their movement. Such a value would be desirable in the case of a 'guided tour'. If developers have access to mouse or wand data in their runtime engine, they can build their own navigation types using prototypes, scripts, and other scenegraph nodes.

Example Scenegraph: a Heads-up-Display

ProximitySensor nodes output events called position_changed and orientation_changed. By placing a ProximitySensor at the origin, we have access to constant updates of the user's location and direction in the 3D world. If appropriate, we can then place a Heads-Up-Display (HUD) in front of the user and within their field-of-view. ROUTEing the output of the ProximitySensor to the HUD's parent transform allows the HUD to continually travel with the user. The following code fragments illustrated this basic design:

```
<Scene>
<ProtoDeclare name="markerP">
 <ProtoInterface> …
 </ProtoInterface>
  <ProtoBody> …
  </ProtoBody>
</ProtoDeclare>
 …
<Group DEF="universe_context">
  <ProximitySensor DEF="universe_origin" center="0 0 0"
        size="1000 1000 1000"/>
      <NavigationInfo type="EXAMINE ANY"/>
  <Background/>
 </Group>
 <Group DEF="HUD_UI">
  <Transform DEF="HUD">
   <Transform translation="-0.05 0.03 -0.2">
    <!— some hud scenegraph translated by an offset
                                              to user's point of view -->
          <Shape DEF="hud_geometry">
        <Box size=".1 .1 .1"/>
        <Appearance>
         <Material diffuseColor="1 1 1"/>
        </Appearance>
       </Shape>
   </Transform>
  </Transform>
 </Group>
```

```
…
<Group DEF="worldGroup">…
</Group>
…
 <ROUTE fromField="position_changed" fromNode="universe_origin"
  toField="set_translation" toNode="HUD"/>
 <ROUTE fromField="orientation_changed" fromNode="universe_origin"
  toField="set_rotation" toNode="HUD"/>
</Scene>
```

4.3 X3D and XSLT Techniques

[Kim, 2002] demonstrated the power of the content/presentation distinction when they used XML, Schemas, and XSLT to render their XML descriptions of dynamic, physical systems to different 3D visual and system metaphors they call *rubes*. [Dachselt, 2002] have demonstrated an abstracted, declarative XML and Schema to model Web3D scene components and especially interfaces. More recently [Dachselt and Rukzio, 2003] leverage object-oriented concepts and XML Schema to componentize scenegraph node sets in the definition of user interface 'Behavior Graphs' which can be applied to arbitrary geometries or widgets. Finally, XSLT data transformations for audience-specific interactive visualizations have been shown for the delivery of Chemical Markup Language (CML) using X3D and VRML [Polys, 2003].

Applying the power of XSLT to the delivery of interactive 3D scenes is relatively new, and much more research is required in this area. As we mentioned in 4.1.3, the representation of an XML document is by a tree data model. The nodes of the source graph can be selected and their attributes operated on in XSLT by the definition of <xsl:templates match=""/> that use XPath expressions and the <xsl:variable name=""/> element. XPath provides 13 axis by which the data tree may be navigated: child, descendant, parent,, ancestor, following-sibling, preceding-sibling, following, preceding, attribute, namespace,self, descendant-or-self, and ancestor-or-self. The target X3D tree (scenegraph) can be composed with the DOCTYPE:

```
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
```

There is a content model in X3D (expressed in the DTD and Schema) that constrains the target output and lets tools validate scene. While more formal theories including graph transformation principles are still forthcoming, we can begin to describe techniques for mapping data to visual structures (X3D nodes) for information visualization.

Including the X3D and VRML specifications, a number of resources exist ([Walsh, 2001],[Ames, 1997] that describe the syntax and behavior of nodes in the scenegraph. Therefore, we will not cover all nodes in detail in this chapter, but rather show how particular nodes may be used to manifest visual markers for information visualizations. We will consider the X3D Immersive Profile as the target platform, though position, orientation, size, color, and shape can be mapped to the Interchange and Interactive profiles. All that is required to deliver content to these platforms is an alternative set of XSLT stylesheets that map the data to the supported target nodes and fields (attributes).

4.3.1 Target Nodes - Geometry

The Transform node manifests its children in the scene and provides fields such as translation, rotation, and scale that account for position, orientation, and size respectively. The Transform node's translation field takes an SFVec3f (a 3 float tuple) to define coordinates in 3-space where the children are located. Rotation is an SFRotation field where the first 3 values define a vector which serves as the rotational axis and the last value is an angle in radians which is the amount of rotation around that axis. The scale field is also a SFVec3f which defines a scaling factor for the node's children between 0 and 1 along each dimension (x, y, and z).

Shape is obviously a crucial X3DChildNode. The Shape node describes both geometry and its appearance, such as color and texture. The X3D color model is defined in RGB space and specified in the Material node. In X3D, color is specified by RGB values. The specularColor and emissiveColors modulate the diffuse color by lighting, shape, and point-of-view. In the literature on information visualization, there is a distinction between hue and saturation as visual markers in display mappings [Mackinlay, 1986]. When colors are interpolated, the VRML Sourcebook[Ames, 1997] notes that colors are converted to HSV space (which does have a saturation factor) and then converted back to RGB. For readers interested in specifying saturation factors or converting between these color spaces, I recommend consulting [Foley, 1995]. When mapping data to color as a visual marker, it is important to use distinctive or contrasting color scales so that users can differentiate the rendered values.

3D Geometry in an X3D scene may be built with any number of nodes, including the geometric 'primitives' (Box, Cylinder, Sphere) and others, such as: PointSet, IndexedLineSet, IndexedFaceSet, Extrusion, the Triangle* family, and Text. Each of these nodes has its own field signature and depending on the designer's goal or user's task, the same data may be mapped to these different markers. Some brief notes about these shapes are in order. The PointSet node may be used for a scatter-plot for example, but as a point does not have any volume, their specific values may be difficult to perceive in the rendering. The way some primitives' dimensions (eg the Cylinder's height and the Box's size) are defined, they usually need to be Transformed (offset) by half of this dimension. IndexedFaceSet and IndexedLineSet geometries require a coordIndex field to specify the order in which the Coordinate points are connected.

In addition, X3D has extended VRML geometries by adding the Geometry2D component. Arc2D, ArcClose2D, Circle2D, Disk2D, Polyline2D, Polypoint2D, and Triangleset2D are defined with this component. Similar 2D primitives are defined in SVG (W3C, 2002; this volume). The shapes in this component are new to Web3D worlds, and we expect them to be very useful in future visualization and interface designs. Currently, the Geometry2D component is only supported in the Immersive Profile.

4.3.2 Target Nodes – Hyperlinks and Direct Manipulation

The Anchor node is a grouping node that provides the ability for the user to click on its children and load an external resource. This is analogous to the hyperlinking <a> tag in HTML and the default behavior is for the resource to totally replace the currently loaded scene. The url field is of MFString type that lists the location of one or more resources. The browser attempts to find the first resource and load it; if it is not accessible, it tries the next one. Similar to the HTML hyperlink, the Anchor's parameter field can specify a frame or window target where the resource is to be loaded. When X3D or VRML files are specified as the resource, the link may also include a Viewpoint which is to be bound. This is done simply by appending #DEFedViewpointName to the url. The specified resource may also be a CGI script on the server and variable values may be passed to it. For example:

url "http://www.somedomain.org/sample/vistransformer.pl?marker=markerP&data=autos"

In this case, the CGI script is responsible for delivering the content header and composing the scene.

Direct manipulation (such as clicking on an object and dragging it) in X3D can be accomplished through the use of DragSensors such as the PlaneSensor, CylinderSensor, and the SphereSensor. These nodes are activated when the user clicks on any of its sibling nodes and the output values are typically ROUTEd to a Transform node to effect a translation or rotation. A TouchSensor generates events such as isOver, and touchTime events (among others) that can be ROUTEd to other nodes in the scenegraph such as Scripts to process user actions. Again, depending on the application and interactivity requirements, these may also be included in a Prototype definition.

4.3.3 Examples

Using the knowledge we have outlined above, let's have a look at some examples (Figures 4.2, 4.3, and 4.4) of using XSLT to transform some abstract data into X3D scenes. Here is some sample XML data :

```
<Vehicles>
    <Auto name="SUV2001" MPG="8" Cylinders="8" Price="40,000"/>
    <Auto name="SUV2000" MPG="12" Cylinders="8" Price="35,000"/>
    <Auto name="Van2000" MPG="16" Cylinders="6" Price="30,000"/>
    <Auto name="Pickup1990" MPG="23" Cylinders="4" Price="21,000"/>
    <Auto name="Sedan1999" MPG="30" Cylinders="4" Price="18,000"/>
```

```
    <Auto name="Compact2002" MPG="38" Cylinders="4" Price="14,000"/>
 </Vehicles>
```

In order to transform this data to an X3D visualization with XSLT, we define a template (or set of templates) that extract the source elements and attribute values we are interested in. The templates in an XSLT stylesheet provide a mapping from XML data to X3D informational objects. Common XSLT design patterns have been described such as fill-in-the-blank, navigational, rule-based, and computational [Kay, 2001]. Based on this mapping, the XSL Transformation engine writes the data values into the template X3D tags and writes the result to the network or to a file (as in section 4.5). For this example source data, we might write our XSLT as follows:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" encoding="UTF-8" media-type="model/x3d+xml" indent="yes" cdata-section-elements="Script" doctype-system="http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"/>

    <xsl:template match="/">
        <X3D profile="Immersive">&#10; <head>&#10;
                <meta content="translatedVehicleData.x3d" name="filename"/>
                <meta content="XSLT translation 1" name="description"/>
                <meta content="n_polys" name="author"/>
            </head>
            <Scene>
             <!-- Insert EXTERN / PROTO declarations, universe set, UI,
                                                   and Scripts as needed -->
                <xsl:apply-templates/>
            </Scene>
        </X3D>
 </xsl:template>

 <xsl:template match="Vehicles">
     <Group DEF="worldGroup">
        <xsl:for-each select="Auto">
          <xsl:variable name="name" select="@name"/>
          <xsl:variable name="mpg" select="@MPG"/>
          <xsl:variable name="cyl" select="@Cylinders"/>
          <xsl:variable name="price" select="@Price"/>
                <Transform>
                    <!—- Manipulate the variables as necessary and
                      instantiate X3D visual markers (target geometry) -->
                </Transform>
        </xsl:for-each>
 </Group>
 </xsl:template>

</xsl:stylesheet>
```

Let's take a look at how some visual markers may be instantiated in an X3D scene. The following code fragment (shown in Figure 4.2) generates a scatter-plot view of the automobile dataset using an XSLT stylesheet to map quantitative data to a Transform node's translation field and categorical values to Material.

```
<Group DEF="worldGroup">
 <Transform translation=".8 4 1">
  <Shape DEF="marker1">
    <Appearance>
          <Material diffuseColor="1.0 1.0 1.0"/>
          </Appearance>
   <Sphere radius=".15"/>
  </Shape>
 </Transform>
```

```
 <Transform translation="1.2 3.5 1">
  <Shape>
    <Appearance>
        <Material diffuseColor="1.0 1.0 1.0"/>
        </Appearance>
   <Sphere radius=".15"/>
  </Shape>
 </Transform>
 <Transform translation="1.6 3 2">
  <Shape>
   <Appearance>
          <Material diffuseColor="0.31 0.3 0.61" />
        </Appearance>
   <Sphere radius=".15"/>
  </Shape>
 </Transform>
 <Transform translation="2.3 2.1 3">
  <Shape>
   <Appearance>
        <Material diffuseColor="0.89 0.44 0.89" />
        </Appearance>
   <Sphere radius=".15"/>
  </Shape>
 </Transform>
…
</Group>
```

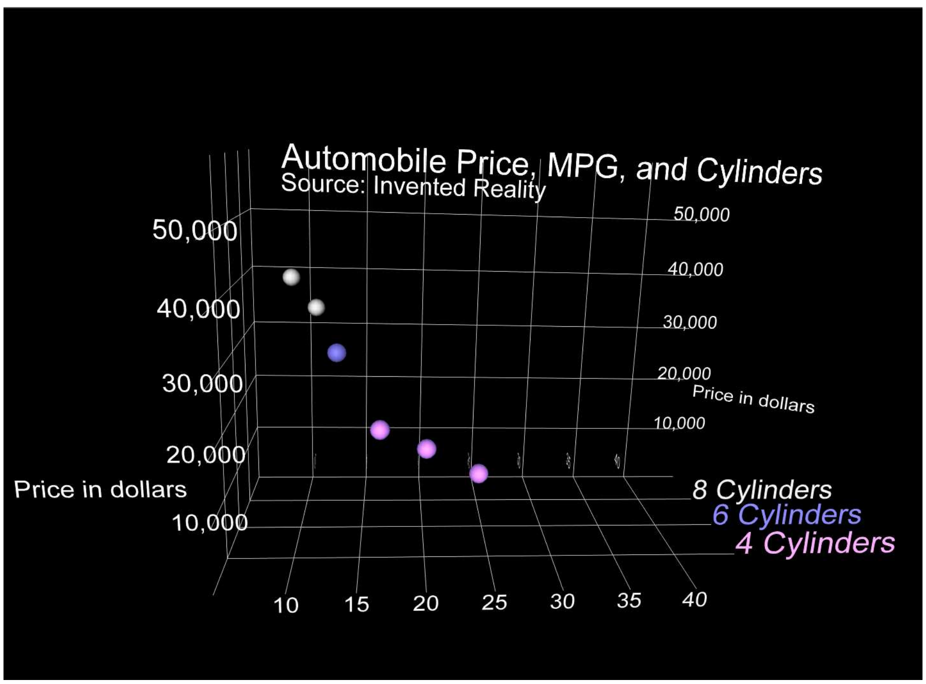


Figure 4.2 X3D scatter-plot geometry using positioned, color-coded Spheres as the visual markers

The second example (Figure 4.3) implements quantitative values mapped to Cylinder height (which are Transformed vertically by half their height value) and categorical values mapped to Material. The target X3D code for this example would be as follows:

```
<Group DEF="worldGroup">
 <Transform translation=".8 2 1">
  <Shape DEF="marker2">
```

```
   <Appearance>
            <Material diffuseColor="1.0 1.0 1.0"/>
            </Appearance>
  <Cylinder height="4" radius=".15"/>
 </Shape>
</Transform>
<Transform translation="1.2 1.75 1">
 <Shape>
   <Appearance>
            <Material diffuseColor="1.0 1.0 1.0"/>
            </Appearance>
  <Cylinder height="3.5" radius=".15"/>
 </Shape>
</Transform>
<Transform translation="1.6 1.5 2">
 <Shape>
   <Appearance>
              <Material diffuseColor="0.31 0.3 0.61" />
            </Appearance>
  <Cylinder height="3" radius=".15"/>
 </Shape>
</Transform>
<Transform translation="2.3 1.05 3">
 <Shape>
   <Appearance>
            <Material diffuseColor="0.89 0.44 0.89" />
          </Appearance>
  <Cylinder height="2.1" radius=".15"/>
 </Shape>
</Transform>
…
</Group>
```

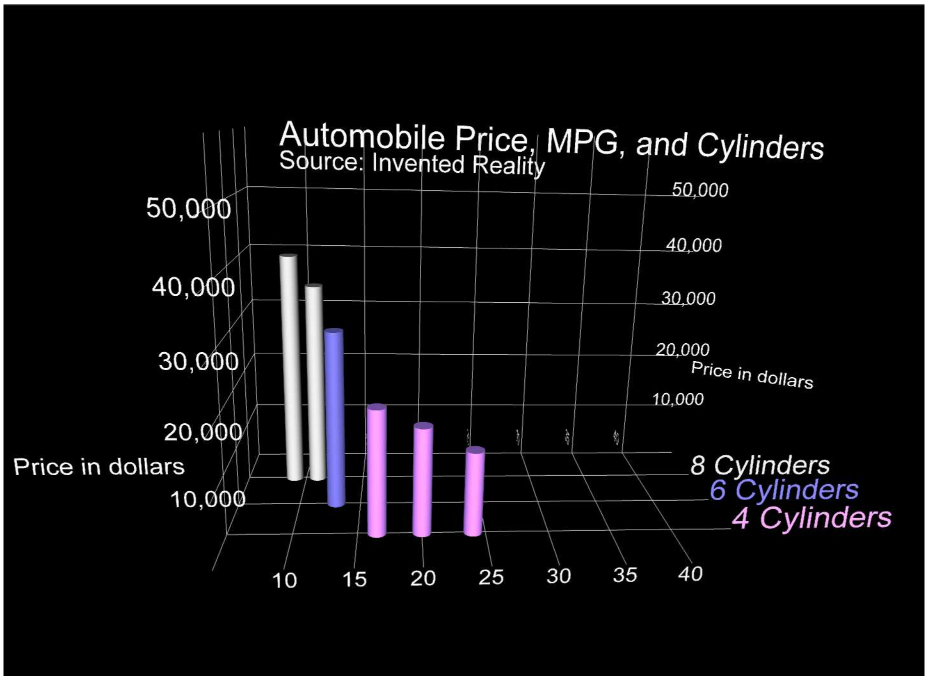


Figure 4.3 X3D bar graph (or histogram) geometry using positioned, color-coded Cylinders and markers. Box primitives could also be used in this way.

Prototypes' definitions can add another level of efficiency to the definition of data objects where multiple nodes can be encapsulated and re-used. In the first two examples, the initial overview Viewpoint gives us a rough idea about the distribution of automobiles across the 3 variables. However, we would likely want to find out more detailed information about an automobile that met our criteria. To accomplish this without cluttering the visual space, we can define our visual markers with an LOD (Level-of-Detail) functionality, which renders different children based on the user's proximity. One such design would show the detailed view (a Text node reading the name, miles per gallon and price) when the user zooms in closer to an item of interest. In addition, Text could be placed on a Billboard node that rotates its children around their y-axis to always face the user.

Our third example populates a PrototypeInstance with values and has the high Level-of-Detail containing Billboarded Text and the low level containing the geometry from the first example. The PrototypeDeclaration is named "markerP". Here is the code for these visual markers using the automobile dataset:

```
<Group DEF="worldGroup">
 <ProtoInstance name="markerP">
  <fieldValue name="position" value=".8 4 1"/>
  <fieldValue name="cost" value="40,000"/>
  <fieldValue name="name" value="SUV2001"/>
  <fieldValue name="numcyl" value="8"/>
  <fieldValue name="miles" value="8"/>
  <fieldValue name="color" value="1.0 1.0 1.0"/>
 </ProtoInstance>
 <ProtoInstance name="markerP">
  <fieldValue name="position" value="1.2 3.5 1"/>
  <fieldValue name="cost" value="35,000"/>
  <fieldValue name="name" value="SUV2000"/>
  <fieldValue name="numcyl" value="8"/>
  <fieldValue name="miles" value="12"/>
  <fieldValue name="color" value="1.0 1.0 1.0"/>
 </ProtoInstance>
 <ProtoInstance name="markerP">
  <fieldValue name="position" value="1.6 3 2"/>
  <fieldValue name="cost" value="30,000"/>
  <fieldValue name="name" value="Van2000"/>
  <fieldValue name="numcyl" value="6"/>
  <fieldValue name="miles" value="16"/>
  <fieldValue name="color" value="0.31 0.3 0.61"/>
 </ProtoInstance>
 <ProtoInstance name="markerP">
  <fieldValue name="position" value="2.3 2.1 3"/>
  <fieldValue name="cost" value="21,000"/>
  <fieldValue name="name" value="Pickup1990"/>
  <fieldValue name="numcyl" value="4"/>
  <fieldValue name="miles" value="23"/>
  <fieldValue name="color" value="0.89 0.44 0.89"/>
 </ProtoInstance>
…
</Group>
```

Figures 4.4 and 4.5 show a sample visual marker PROTO that includes LOD, Billboard and Text features. From outside the detail LOD range, the scene would look exactly as Figure 4.2.

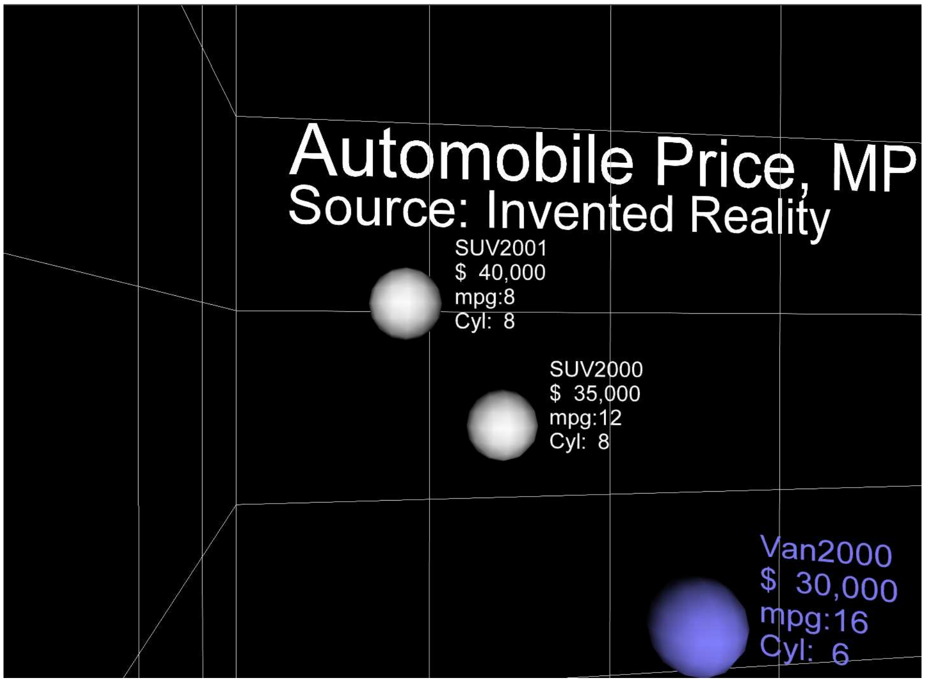


Figure 4.4 A zoomed-in view of Prototyped visual markers encapsulating perceptual and abstract information. The user has navigated into the higher price range.

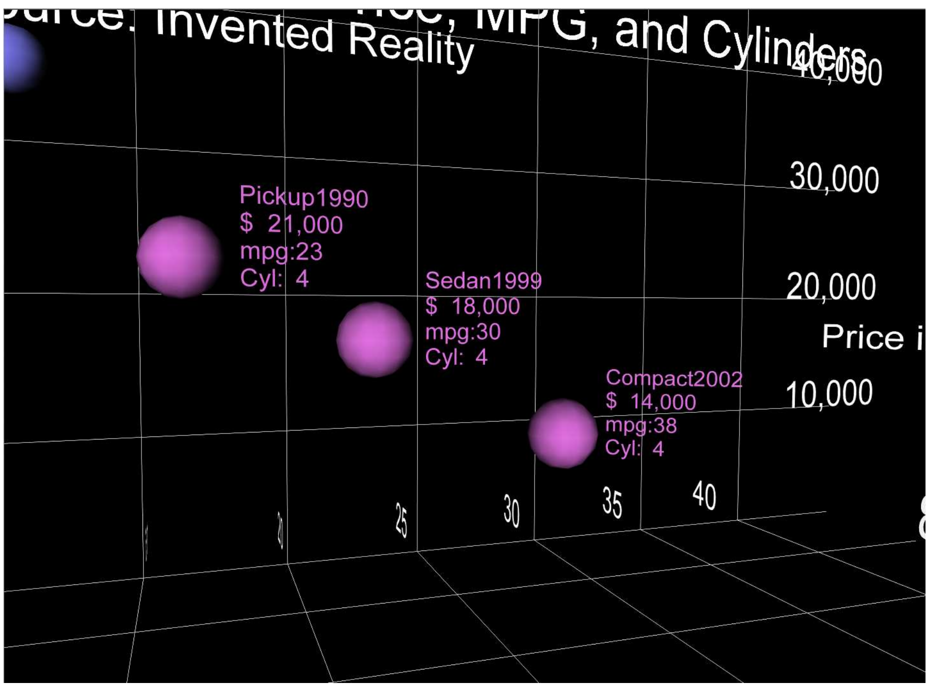


Figure 4.5 A zoomed-in view of Prototyped markers encapsulating perceptual and abstract information. The user has navigated into the lower price range.

XSLT can, of course, also be used to transform and compose X3D from data that has inherent spatial meaning such as locations, sizes, and connectivity. For example, Figures 4.6 and 4.7 show the results of 2 different stylesheets that process a Chemical Markup Language (CML) file of the cholesterol molecule to X3D. The first version (Figure 4.6) builds geometry from atom and bond elements and text from abstract attributes and other meta information.

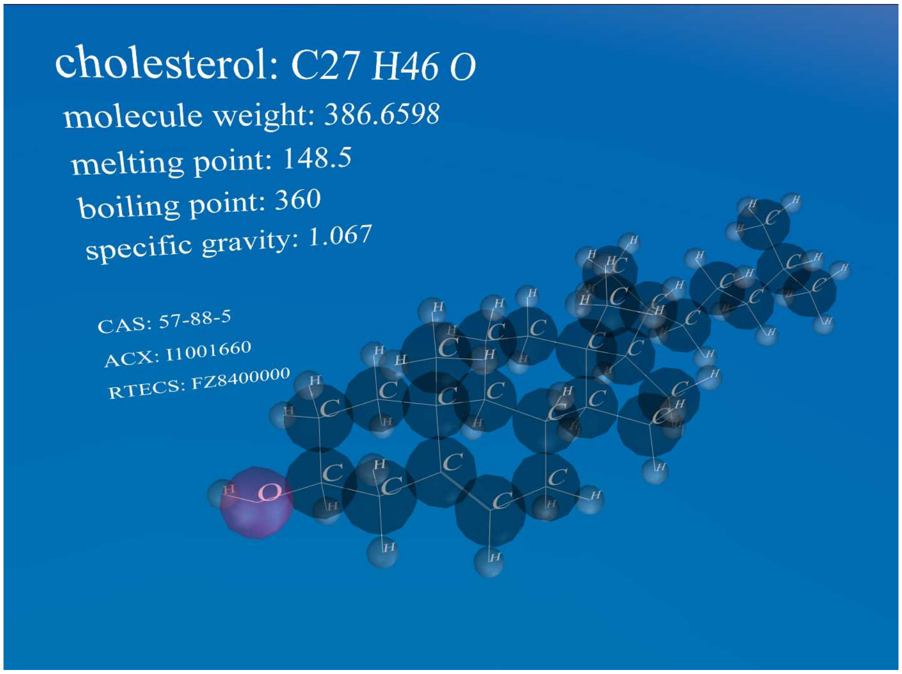


Figure 4.6 The results of an XSLT transformations of a CML file for cholesterol

The second transformed version (Figure 4.7) shows that the XSLT can add control widgets to the resulting X3D scene; in this case, a slider controls the transparency of every atom. In addition, the transformation in Figure 4.7 shows a new text style as well as movable measuring axis instantiated in the 'universe block' of the scene.

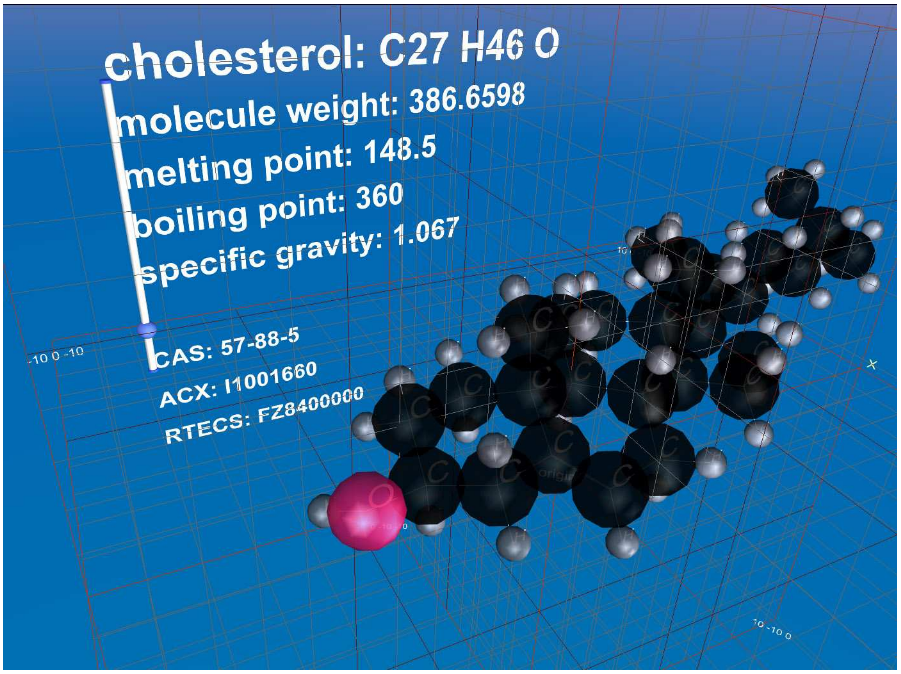


Figure 4.7 The results of an XSLT transformations of a CML file for cholesterol. A new FontStyle has been used, and a slider widget has been added during the transformation and ROUTEd to visual markers in the scene.

Figure 4.8 and 4.9 illustrate the XML to X3D transformation results of a finite-difference mesh of tissue used for *in silico* biological simulation (PathSim, Chapter 5).

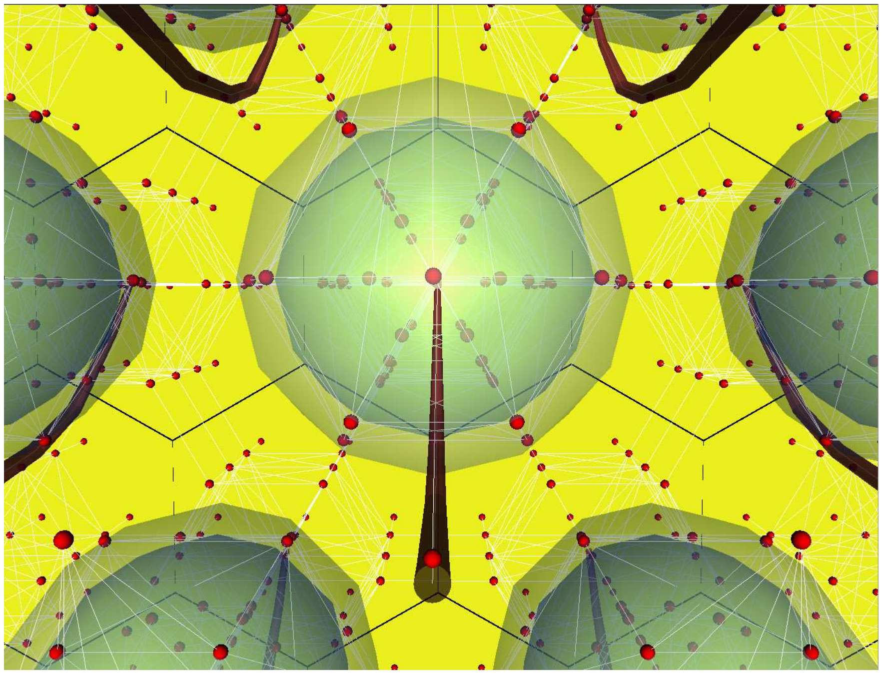


Figure 4.8 Underside view of an XML finite-difference mesh description generated via XSLT to X3D in order to visualize the spatial locations and connectivity of mesh points (PathSim, Chapter 5).

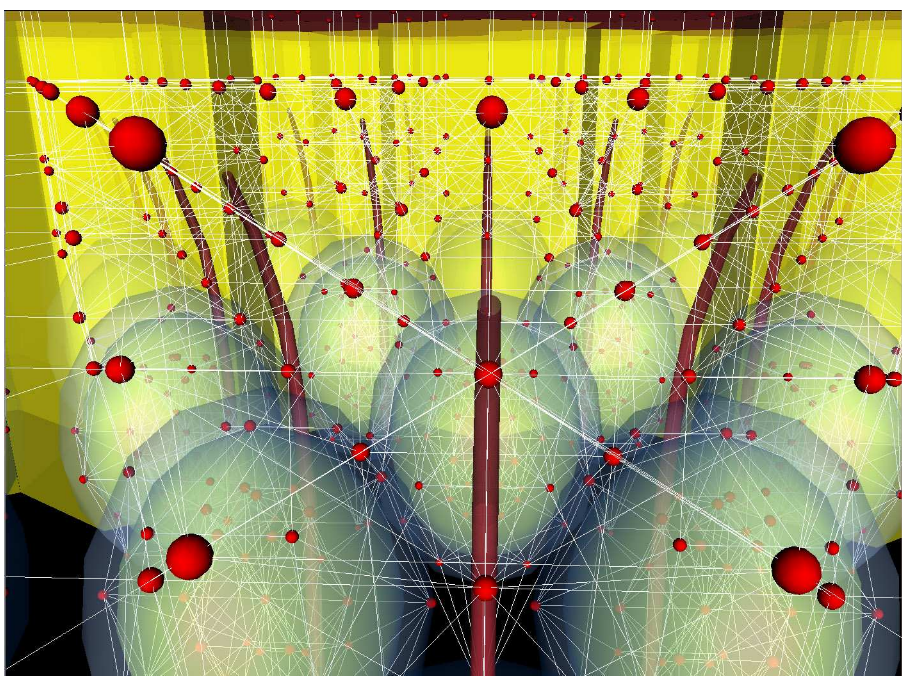


Figure 4.9 A front view of the XML finite-difference mesh (PathSim, Chapter 5)

4.4 Scene Management and Runtimes

Another important consideration in the composition and maintenance of world content is the use of the Inline node. In VRML, Inlines were opaque in that events could not be ROUTEd between the inlined and the inlining scenes. This event opacity is also a limitation of the Browser.createX3DFromURL method since nodes in the new world are not programmatically addressable. If authors wanted to dynamically replace a world block and connect it with event ROUTEs, the not-so-obvious solution in VRML has been to define the entire replacement scene as Prototypes and then use the Browser.createX3DfromString method to add the new node and the Browser.addRoute method to connect events to it. The new X3D API is called the Scene Access Interface (SAI), and unifies the object definitions for both internal and external scripting. The SAI is a much more rich and rigorous programming specification than VRML supported and it introduces a number of new objects and functions. The bindings for the Java and ECMAScript languages are described in ISO/IEC FCD 19777:200x.

The Browser object interface, for example, has a number of useful methods for managing content dynamically such the Browser.createX3DFromURL or Browser.createX3DFromString methods that can be invoked from a Script. These methods (whose analogs were specified in VRML97) allow scene content to be swapped during runtime. The content is added to a specific part of the scenegraph by specifying a DEFed node which the new content replaces. If the world has been designed in a modular way as we described above, this can be a very powerful technique.

Other important functionality newly introduced in X3D is the use of IMPORT and EXPORT keywords with Inlines. The IMPORT statement provides ROUTEing access to all the fields of an externally defined node with a single statement and without a PROTO interface wrapper and Scripts building String objects. The EXPORT statement is used within an X3D file to specify nodes that may be imported into other scenes when inlining the file. Only names exported with an EXPORT statement are eligible to be imported into another file (Web3D, 2002). In this way, entire X3D files can declare event communication routes for embedding and embedded files. This is a significant improvement in the composability and re-use of X3D worlds themelves.

4.5 Publishing Technologies

We have examined some techniques for transforming XML data to X3D with the use of XSLT stylesheets. The X3D Task Group has provided a number of XSLT style-sheets for the transformation of X3D to VRML97 as well as X3D to HTML. Also, courtesy of the National Institute of Standards Technology, a translator application for VRML97 -> X3D data migration has been made freely available and been integrated into a number of Web3D editing tools including the structured editor X3D Edit [Web3D] and others. Within the Pipeline and Hybrid paradigms, there are 2 general ways we will consider in publishing XML content to X3D (or other): the back-end production of a file archive, and the serving of a transformed and presented source document in response to a 'live' (networked) visualization request Thus we distinguish between the auto-generation of content archives, and the serving of dynamic content for on-the-fly service.

Given server overhead, bandwidth, and delivery constraints, periodically auto-generating content archives may be appropriate. These approaches use X3D source files and directories with naming conventions with scripted XSLT to produce framed HTML, VRML, and X3D document trees complete with linked with chapters, titles, and embedded views of the source file. The generated document trees can be organized and hyperlinked for navigation with a web browser for example. The X3D Task Group's web collection of X3D content examples is an ideal showcase of this technique [Web3D]. The auto-generation can be done with straightforward batched XSLT Java [McLaughlin, 2001; White, 2002; Kay, 2001] or Perl [Kay, 2001; McLaughlin, 2001; Brown, 2002; White, 2002; Polys, 2003] scripts. These content publications can then be served over the web or distributed on CD or DVD as in the Identity Paradigm.

The second approach is using XSL Transformations 'on the fly' using common web server software such as Apache Cocoon, Perl and the XML Gnome libraries, or PHP [Brown, 2002]. This approach can provide custom presentations of the source data with a proportionate server and network overhead. Either of these delivery approaches may be classified as conforming to the Pipeline, Composition, or Hybrid paradigms depending on how the data is transformed and composed.

For visualization systems using XML for data interchange and X3D for data delivery, we constructed a set of tools to process, generate, and deliver IRVE presentations. This is accomplished through the specification of parameterized IRVE display components via a DTD and Schema. Robust mappings of content to presentation can be acheived through XSLT.

We have formalized an XML language and content model for IRVE displays using the W3C's DTD and Schema tools. This DTD and Schema provide syntactic and semantic production rules for IRVE display spaces. These XML tools can be used to describe, validate, and generate IRVE scenes. For example, the Schema for Semantic Objects populated with our display components is shown in Figure 4.16. The full DTD and Schema documents are included in Appendix A.

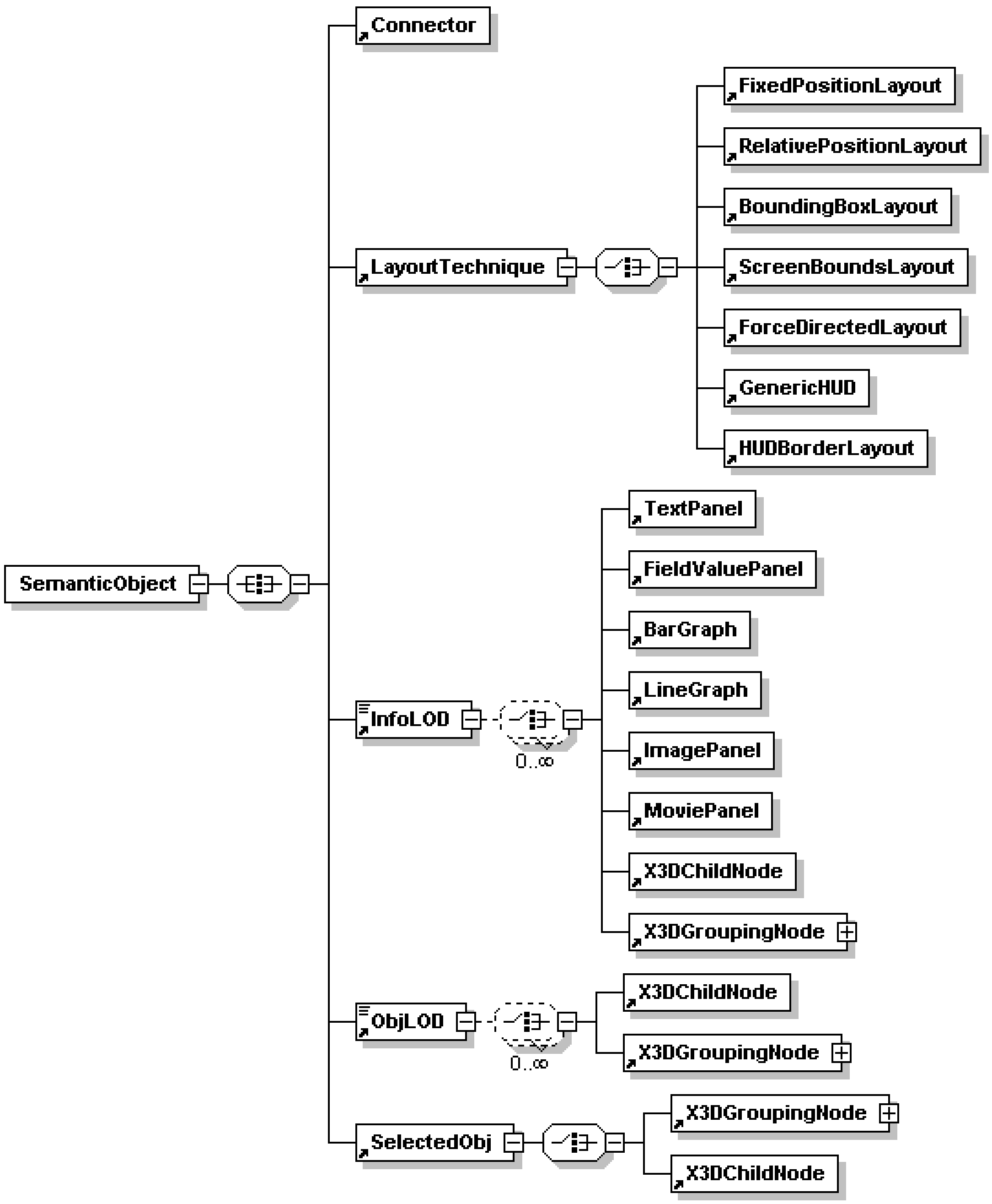


Figure 4.16: Strawman XML Schema for Semantic Objects implemented in this research

Using such an IRVE content model, developers can mark up XML data sets and transform them into X3D and VRML code that implements our display components. By populating these IRVE tags with data, we have an **information mapping configuration** for the integrated information space. We can apply these mappings to any sort of virtual environment content as well as abstract information types. The formalism of SemanticObjects comprises our IRVE testbed - a software platform that can support the systematic manipulation and empirical evaluation of IRVE display and interaction parameters. The syntax and semantics of the information mapping file can provide a concise description of IRVE design composition. For this research, exemplar IRVE data sets will be constructed. These descriptions (of data sets and display configurations) comprise the space of independent variables that will be explored in this research.

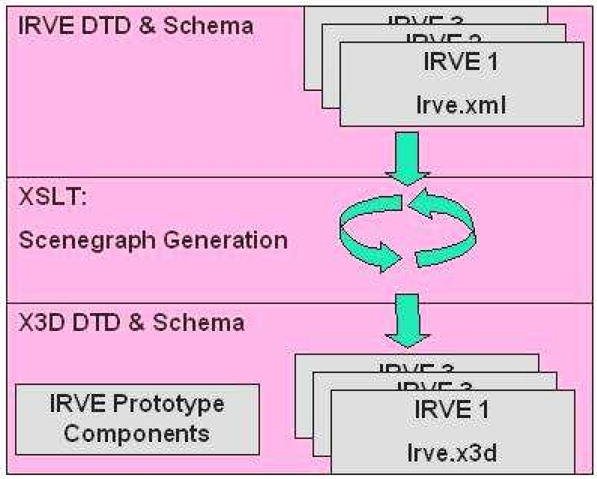


Figure 4.17: XML tools in the Description, Validation, and Generation of IRVEs

4.6 Summary

In this chapter, we looked at modular approaches to X3D scene design and production and examined how XSLT can be used to transform and deliver XML data to X3D visualizations within current publishing paradigms. The separation of content from presentation in XML gives organizations a great deal of flexibility in how developers re-purpose and publish their data. The XML encoding of X3D allows developers to leverage the power of XML to transform the same data to multiple forms and interactive contexts. As XML databases and server technologies improve, we can expect further refinements to the techniques we have outlined.

The investigation of human computer interaction for information-rich 3D worlds and visualizations is still in its infancy. We expect that by enumerating effective data mappings, the combinations of coordinated information and media types, and interaction strategies for information-rich virtual environments, we can work toward advantageous computational, compositional, and convivial systems for real-time exploration, analysis, and action. This work will have a direct impact on the usability and design of such heterogeneous 3D worlds. With such mappings, coordinations, and strategies in hand, effective displays and user interfaces may be automatically generated or constructed by users depending on their expertise level and their current task.

5. PathSim Case Study

We presented the first version of PathSim Visualizer (v0.1) in a conference paper at the Web3D Symposium 2004 [Polys, 2004d]. PathSim v0.2 was completed in August 2005 and additional publications are in preparation. PathSim is an ideal application example that illustrates both the challenges and opportunities for standards-based IRVEs. Through the process of developing PathSim Visualizer, we articulated many of the design tradeoffs for IRVEs and built many of the IRVE display components described above (Chapter 3). This chapter details the motivation, development process, and results for the PathSim IRVE.

5.1 Introduction

The emerging paradigm of digital biology is providing researchers with new computational tools for modeling and analysis. The multi-disciplinary field of Bioinformatics has advanced the application of new simulation techniques, algorithms, and data modeling to biological systems across genomics, proteomics, metabolomics, immunology, and epidemiology. Not only are the systems complex, spanning multiple scales and factors, but they also generate massive quantities of data. This data is heterogeneous, meaning it consists of spatial, temporal, and abstract types, each with its own structure. Temporal and abstract information may be related to spatial, biological structures such as cells, tissues, organs, and systems for example. This data may also be distributed across a variety of local and remote machines and application servers.

For effective scientific visual analysis, researchers and clinicians need integrated access to this variety of information resources and consequently, improved systems for the management and presentation of this data. We have been working with medical and bioinformatics researchers to design and develop next-generation interfaces to explore and understand biological data such as models, simulations, and their references. PathSim Visualizer takes the approach of displaying 3D anatomy (spatial information) in an interactive virtual environment (temporal information) that is annotated and enhanced with a variety of abstract information about the anatomy. This abstract information may include text, numbers, hyperlinks, graphs, videos or audio resources referring to some object, world, or user state. We have described the principal interface design challenges for this class of problem using the term 'Information Rich Virtual Environments' (IRVEs) [Bowman, 2003a]

5.1.1 Usability Engineering

We applied the usability engineering process [Rosson and Carroll, 2003] to develop a visualization tool for in silico immunology simulations. In silico experiments are useful when clinical data is difficult or expensive to collect, or when experiments are too dangerous or unethical to perform in vivo. Once a biology simulation model is validated and tuned to known data, such simulations can help researchers test 'what if …' hypotheses and develop interesting experimental questions for further investigation and investment.

The PathSim Project [Polys, 2003; Harris, 2004] simulates pathogen and host interaction with an agent-based computer model built from current biomedical knowledge. In PathSim, systems biology investigators are concerned with different infection behaviors as they are related to various systems and parts of the anatomy over time. PathSim simulations may run on large servers or clusters, but the results must be accessible to researchers on desktop machines across the network.

Our work has been iterative, gathering user requirements, designing and implementing the interface framework, and refining it through user evaluations. This chapter enumerates the problems and tradeoffs we encountered in building the prototype system for PathSim Visualizer and provides the rationale and details behind our design solutions. These solutions involve encapsulating physical scales and information behaviors into custom scenegraph objects that manage scale, timeseries, and information visualizations for in silico research and analysis.

5.1.2 IRVEs

In our work, we are developing *information-rich* virtual environments (IRVEs)[Bowman, 2003a]. In a nutshell, IRVEs are a combination of traditional virtual environments and information visualization; that is, they provide a realistic sensory experience that is enhanced with some representation(s) of related abstract information. In this way, IRVEs can provide for: a better understanding of the relationships between perceptual and abstract information, improved learning of educational material, greater enjoyment and engagement with the VE, and a decreased dependence on other tools for the viewing of abstract data. This combination of sensory and abstract information is typical for data generated by biological simulations and biomedical research systems such as PathSim.

The goal of the IRVE research agenda is to understand how media designers can disambiguate perceptual stimuli and enable users to accurately form concepts about and mental models of the phenomena they perceive. By taking account of how humans build their cognitive models and what perceptual predispositions and biases are in play, designers can take steps to minimize or leverage their effect. This line of inquiry has been termed the 'inverse problem of design' by Joseph Goguen [Goguen, 2000] and 'Information Psychophysics' by Colin Ware [Ware, 2003]. The research and analysis we present here is couched in a framework for understanding user activities and requirements known as user-centered and scenario-based design [Rosson and Carroll, 2002].

5.1.3 IRVEs for Medicine and Biology

While PathSim has obvious medical applications, it goes much further than that. It may also serve as a basic research tool for life scientists working on a range of questions, and a teaching tool that could find application from K-12 all the way to professional medical training. The value and need for such tools have long been recognized [Farrell and Zappulla, 1989; Kling-Petersen, Pascher et al., 1999]. It has been shown that conceptual learning can be aided by features of VEs such as: their spatial, 3-dimensional aspect, their support for users to change their frames of reference, and the inclusion of multi-sensory cues [Salzman, 1999].

This is compelling evidence for the value of VEs as experiential learning tools and for concept acquisition during the development of a user's mental model. The NYU School of Medicine [Bogart, 2001] has published a number of anatomy courseware modules in VRML that provide an IRVE interface to detailed models of the human head. The Open Virtual Reality Testbed Group at the National Institute of Standards and Technology has produced AnthroGloss [Ressler, 2003], which is an IRVE Anthropometric Landmark Glossary in VRML. We combine referenced elements and adaptations of these models to provide users with context as they explore PathSim simulation results.

Systems biology researchers have begun to use modern computing power to simulate the immune system using generalized cellular automata (i.e. [Celada, 1992; Grilo, 2001; Puzone, 2002]). These simulations use probabilistic or deterministic rules to govern the interaction of automata on some lattice or in some grid space. There is a broad range of implementation details concerning the simulation that cannot be covered here. These are principally concerned with the nature and evaluation of the rules governing agent interaction. However, the PathSim system is unique in that the agents (Virions, B-cells, T-cells, etc.) may number in the millions (10^8) and they travel and interact on a micro-scale 3D mesh that approximates average human anatomy.

In biotechnology, there are a number of groups that have defined XML-based languages for describing systems and data relating to biology. The Physiome project has specified AnatML, FieldML, and CellML [Physiome Project, 2003] which describe finite element geometry, spatially varying fields, and mathematical cellular models respectively. Systems Biology Markup Language [SBML, 2003]allows the flexible representation for models of biochemical reaction networks. The Foundational Model of Anatomy is a Semantic Ontology describing classes and relations of structures and systems [FMA, 2004]. These languages are considered future integration targets for the PathSim simulation architecture as it becomes more developed and robust.

5.2 Information Types

For each simulation run, a configuration file is used to generate the simulation environment, populate it with agents, and specify runtime parameters. This section details the information types represented and visualized through PathSim.

5.2.1 Multi-scale Spatial Information

PathSim simulations run on anatomical meshes that are generated to a hierarchical archive according to current clinical knowledge. Each point in the mesh represents a certain type and volume of tissue where agent interactions (hosts/pathogens) can take place. We have modeled the lymphatic tissue (especially tonsils), blood circulation, and lymphatic drainage of the Waldeyers' Ring from the macroscopic level to the microscopic level. The Waldeyer's Ring is a collection of lymphoid tissue encircling the top of the esophagus (Figure 5.1).

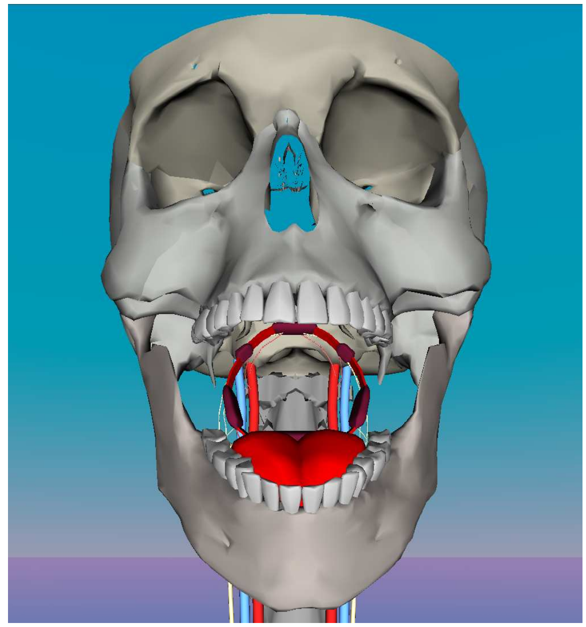


Figure 5.1: The generated Waldeyer's Ring at the 'Macro-scale';
(skull model [Bogart et al., 2001] shown for reference)

The anatomical description is hierarchical XML and distributed across a number of referenced files. The fundamental unit of the anatomical grid is a hexagonal section of tonsiliar tissue modeled to include 72 distinct tissue volumes and their interconnections. The mesh points represent tissue volumes for: the tonsil surface, reticulated epithelium, mantle zone, and germinal center. Figure 5.2 shows a visualization of the unit anatomical mesh with spheres representing the location of tissue volumes (mesh points) and white lines representing the possible travel paths for agents. Each tissue hexagon represents a column of tonsilar tissue of 0.6 mm diameter and we refer to this as our 'Micro-scale' model.

Blood from the circulation system enters the tissue through the High-Epithelial Venule (HEV) and lymph is drained into the lymphatic system from the mantle zone. The Blood and Lymph volumes connect to each unit tissue and are each represented by stochastic reservoirs of agents. Figure 5.3 shows a labeled example of how unit tissues are arranged to approximate the lymphatic tissue of the tonsils.

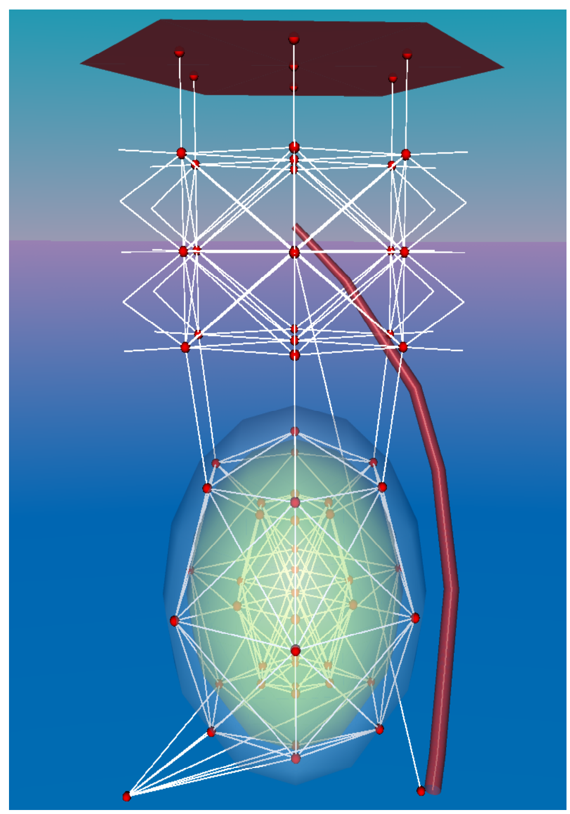


Figure 5.2: a VRML Micro-scale view of the unit section tissue mesh translated from its XML description

PathSim generates interconnected lattices of the unit tissue into larger meshes representing each tonsil. The size of each tonsil mesh is specified by the tonsil's surface dimensions declared in the configuration file. The six main tonsils are connected by another type of mesh (with 18 defined volumes) that represents the diffuse lymphatic tissue, which connects the tonsils into the Waldeyer's Ring. We refer to this as the 'Macro-scale' model.

The relation of all tonsil and connective tissues descriptions is manifested in a macro-level tissue file that defines the simulation environment. Any subsequent processing and visualization is based on references to this hierarchical simulation mesh. In typical simulations, an anatomy may consist of upwards 2300 tissue units for a total of over 166,000 tissue volumes.

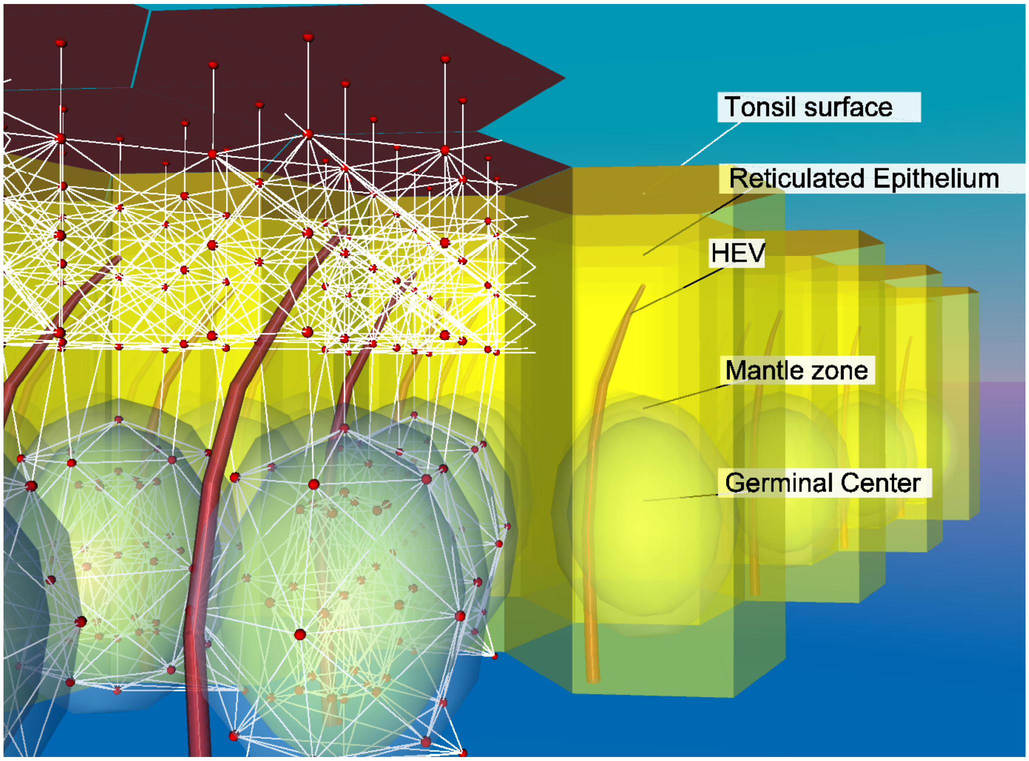


Figure 5.3: A labeled view of the Micro-scale tonsil tissue mesh

5.2.2 Abstract Information

There is a variety of abstract information that may be relevant to a researcher investigating a digital biology simulation through PathSim. For example, there are seven types of agents that interact in the immune system simulation: Epstein-Barr virus, the agents involved in the 'Innate' response (B-cells in their naïve, latent, or lytic phases), and the agents involved in the 'Acquired' immune response (T-cells in their naïve, latent, or lytic phases). This information may be represented graphically or numerically within the virtual environment:

- **Lymphocyte/Virus populations for the system**
- **Lymphocyte/Virus populations per local region or unit**
- **Annotations, hyperlinks, and references about the structure or process being evaluated**

The PathSim Visualizer implements custom software objects to manage, layout, and display this abstract information in the context of the virtual environment. These are described in detail in Section 5.4.

5.2.3 Temporal Information

PathSim Visualizer also renders the dynamic temporal aspect for the abstract and spatial information-how that spatially-registered abstract information changes over time. Through processing components (Visualization Generators), simulation data is transformed into sequencer and interpolator animations. Animation data is used to drive anatomical coloring, as well as global and local population graphs and numerical read-outs.

In PathSim, the simulated timestep is decoupled from the output timestep and both are specified in the simulation configuration file. Simulations may be evaluated at timesteps on the scale of minutes (e.g. 6 minutes) and for time periods on the scale of days (weeks, months, or years). A typical EBV infection will complete its acute phase in the first 45-60 days; because investigators are interested in long-term behavior, runs are sometimes for years of simulation time. Depending on the question being investigated, agent populations may be evaluated and recorded at various resolutions.

Investigators into dynamic systems such as the immune system need capable controls to manage and index the temporal dimension: coarse enough to find a maximum population value in a month of simulated infection time, and fine-grained enough to examine behavior at 15 intervals. PathSim Visualizer synchronizes data across scales through a familiar DVD interface that gives both absolute and relative time control (adapted from NPS SAVAGE archive [NPS, 2003].

5.3 Simulation Services

5.3.1 System Description

PathSim is configured to run either from the command line or as a web service. PathSim is written in C++ and has been run successfully on linux and windows. The linux executable is roughly two megabytes. To provide access to a broad range of users, the PathSim simulation engine is run on a server or High-Performance Computing (HPC) system, but provides setup and visualization facilities through a web-based front end (Figure 5.4).

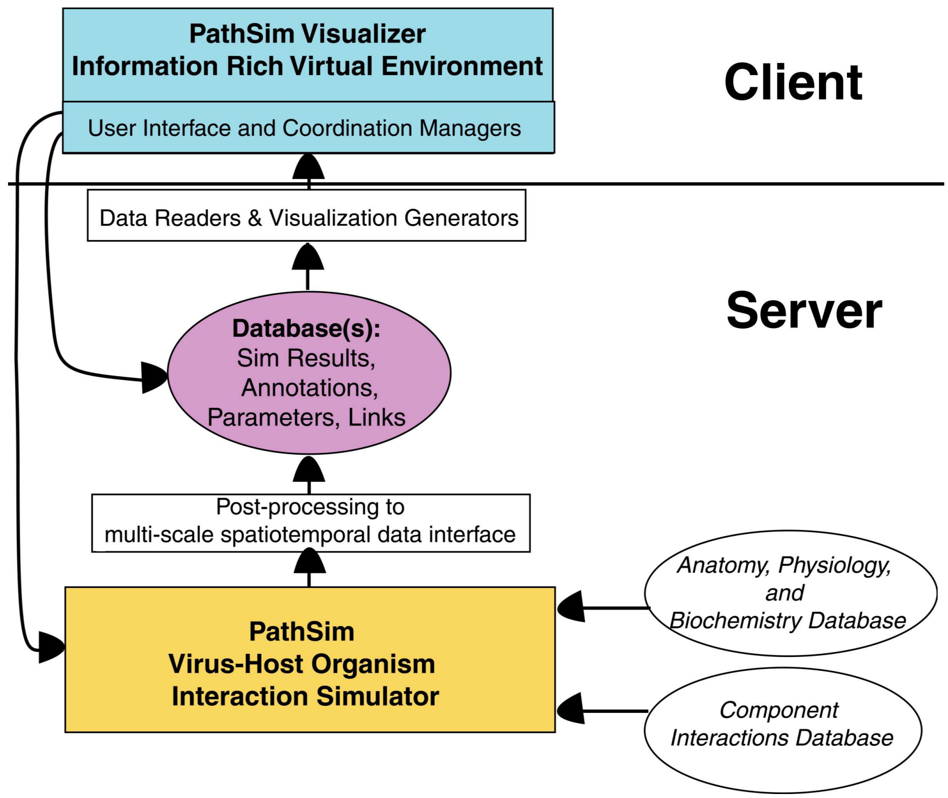


Figure 5.4: PathSim Architecture

Through user interviews and the scenarios generated in the design process, we discovered a set of fundamental activities and goals that users may expect the system to support. The setup activities are presented in a 2D webform interface and include: configuring anatomy parameters, defining agent interaction rules, defining an infection scenario, and setting the simulation parameters such as time interval and duration. User activities for results analysis include: determining the overall behavior of the agent populations during infection, identifying areas of high agent activity (hot-spots), and drilling down to observe agent states and dynamics on local levels.

Users can configure, run, and view PathSim simulations remotely over the web. The mesh description, simulation code, simulation parameters, and results all reside on the server in structured directories and files. The Visualization Generators of PathSim are a set of Perl scripts that process the simulation output files, composing and writing a set of directories and VRML files on the server. One principal challenge (addressed in this paper) is the management and transformation of PathSim simulation results to information-rich objects and scenegraphs that include the anatomical mesh.

Raw simulation results are written into unique files on the server that correspond to the hierarchy of the mesh description files. The results files contain time-stamped population numbers for each agent and each anatomical region at that scale. Visualization Generator scripts read the simulation result files and compute color, string, and float animation values for each region at that scale. Color and float information for each agent type population is normalized to the maximum value achieved during the course of the simulation. Absolute numeric population values are converted to strings for display as field-value pair text. Simulation data is composed into VRML nodes and syntax and the result files saved for on the server for viewing.

5.3.2 Service Architecture

PathSim can be run locally with command line parameters specifying directories for input and output files. Through a set of local scripts, experts and developers can invoke and manage large numbers of runs (e.g., as a way of probing the parameter space, testing code changes, etc). This is a crucial mode of operation; however, it requires proper programming expertise, a properly set up programming environment, and sufficient computational resources.

Due to the large amount of memory and storage requirements for the PathSim and the steep learning curve for the command line tools, we also deploy PathSim as a web-based application service. Users such as biomedical researchers and clinicians can interact with the system through a web interface which allows them to choose values for runtime parameters, manage runs, and view result data. This remote access scheme has a number of benefits including: allowing the simulation to run on a dedicated machine with plenty of space for data, minimizing deployment issues such as user installations and updates, as well as giving end-users a familiar means of operation.

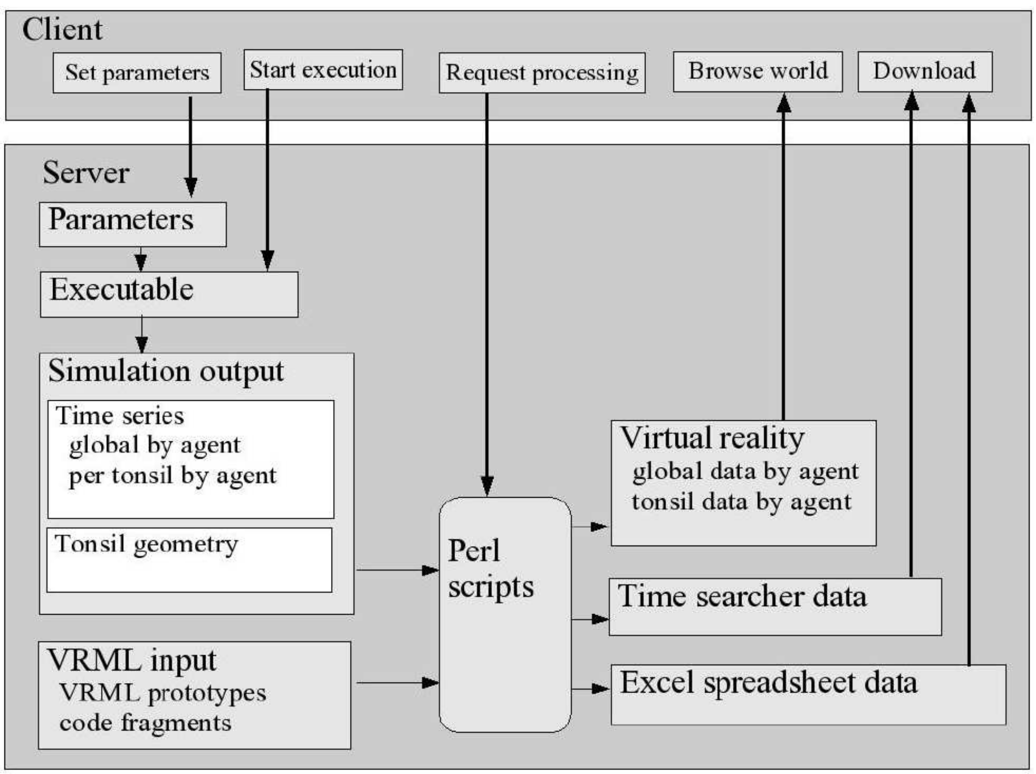


Figure 5.5: Service Architecture for PathSim Web Interface

As a web service PathSim delivers data in three forms: as an information rich browseable 3-D world, as downloadable data for TimeSearcher [Hochheiser, 2004], and as downloadable Excel spreadsheet data.

The service architecture is shown in Figure 5.5. The client is able to set parameters and start the simulator. The simulation engine then produces time series, which represent the course of the simulation and VRML output representing the tonsil geometry. After the engine has run, the user is able to request the processing, which will produce usable output. At this point, a collection of Perl scripts formats the time series data for timesearch and Excel and combines the simulation engine output with VRML prototypes and code fragments to produce the virtual reality environment.

5.3.3 Visualization Software

We adopted a modular approach to the publication of simulation results. First, we created a set of processing scripts that arrange Pathsim result data in a format importable by common tools. For example, data files can be produced for MS Excel, UMD TimeSearcher**,** and MatLab. Once imported, the data can be manipulated and visualized with the tools provided by that application, for example generating time plots or distribution statistics.

Unfortunately there are limitations to all these tools when it comes to large volumes of spatially-registered data. For example, while the user may be able to plot a handful of region populations over time, the spatial relations of those regions are not necessarily represented. In order to understand the spatial behavior of the system (e.g. 'how does the infection spread?'), users must remember and reconstruct the anatomical topology from multiple graphs. This is especially difficult when a large number of anatomical locations are being analyzed. To address this problem, we have built an Information-Rich Virtual Environment (IRVE) interface for Pathsim. The IRVE interface registers abstract data timeseries to 3D anatomy and thus provides a familiar and scalable context for visual data analysis.

Since our first version [Polys, 2004d], we have developed new IRVE scene graph objects that encapsulate multiple view capabilities and improved multi-scale interface mappings. The IRVE is realized in the international standard VRML97 language. In the IRVE, any spatial object (including the global system) can be annotated with absolute population numbers (as a time plot and or numeric table) or proportional population numbers (as a bar graph). Spatial objects themselves can be animated by heat map color scales; heat map color data is also used in the unit tissue visualization to change height for better value discrimination.

5.4 Display Components

PathSim Visualizer displays abstract information related to the simulation in World, Object, and Viewport spaces. PathSim Visualizer gives the user a Heads-Up-Display where system variables and global and macro state are displayed. This HUD functions as a read-out and control panel, travelling with the user throughout the environment. Information displays in the environment aggregate data from smaller scales into suitable, Object space visual representations at larger scales. A video depicting the PathSim v2 IRVE application is listed in Appendix H. This Overview-plus-Detail and multiple-view annotation functionality helps investigators explore and understand the dynamics of the system:• **HUD**- Agent color key, time controller, global and macro population views

• **Agent Lenses –** For each agent type, populations are mapped to: color coding of anatomy, and to height mapping of tissue unit (at micro scales)• **Population Views** – representations for specific agent populations for each anatomical structure can be toggled on or off;

• **Links** – hyperlinked websites, resources, and references may be rendered in additional windows (display-fixed locations)

5.4.1 Nested Scales

Because of this large-scale data, PathSim Visualizer manages an integrated information environment across two orders of magnitude: Macro and Micro scales. Through the standard VRML navigation, users have a number of egocentric spatial navigation options including free-navigational modes such as: fly, pan, turn, and examine. This empowers users to explore the system, zooming in and out of anatomical structures as desired. IN combination with mouse input, expert users can employ control keys (such as 'ALT') for quick mode changes. In addition, the result space is navigable by predefined viewpoints, which

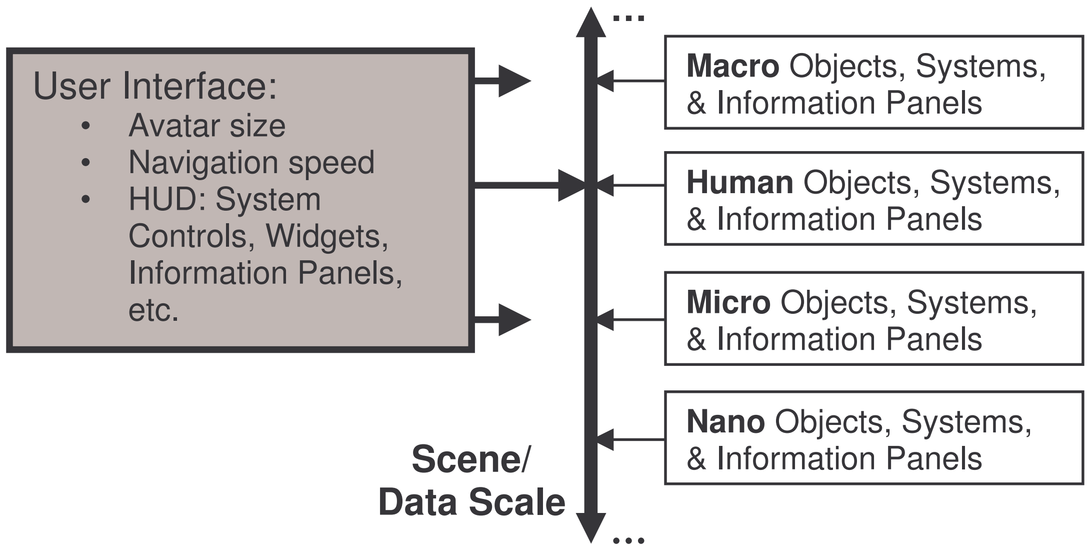


Figure 5.6: Spatial and Abstract Scale requirements for IRVE Activities

can be visited sequentially or randomly through menu activation. This guarantees that all content can be accessible, and users can recover from any disorientation.

PathSim Visualizer manages Macro and Micro scale result visualizations using proximity-based filtering and scene logic `Script`s. As users approach a given anatomical structure, the micro-scale meshes and results are loaded and synchronized to the time on the users' Heads-Up-Display (HUD). To aid wayfinding, certain structures persist across scales (serving as landmarks). Figure 5.6 depicts how global time and simulation data persist across multiple scales.

A crucial requirement for PathSim Visualizer is the capacity to explore simulation results across the macro and micro scales. This presented some interesting scenegraph challenges. Not only did we have to manage a large volume of simulation data for multiple anatomical regions, but also maintain application performance, rendering speed, and interface continuity. For example, the HUD interface should follow the user uninterrupted by zooming and scale changes; the controls on the HUD (such as the DVD Time Controller) must maintain event links to the environment no matter what scale or model is loaded.

The HUD interface is loaded in the top-level file, which also contains `ProximitySensor`s and Scripts to manage scene and state information. In the top-level file, a WorldGroup `Group {}` is defined that contains macro-scale models such as the body, skull. The visualization processing scripts wrap each scale model of anatomy and result animations in a `PROTO` declaration. There is one set_fraction eventIn on the `PROTO` interface that is processed by a TimeManager `Script {}`. Within the Prototype declaration of each scale, the TimeManager script is connected to all sequencers and interpolaters that animate at that scale. This keeps event management encapsulated across scales and allows models to be loaded and connected to the environment easily. A typical zooming sequence is shown in Figures 5.7-5.10.

As users zoom into the head and neck area and the Waldeyer's ring becomes visible, the simulation results are loaded into the WorldGroup using a `Browser.createVRMLFromString` method. The string is an `EXTERNPROTO` definition and an instance. `ROUTE`s between the DVD controller and the new scene are added in order to link the scene to user global time. Similarly, as users zoom into specific anatomical structures (i.e. the tonsils), the appropriate detail geometry and simulation results are loaded into the WorldGroup as an `EXTERNPROTO` instance and animation `ROUTE`s are added. At the micro view, when users select hexagonal tissue sections, a script requests more data from the PathSim server: it calls a cgi-script in the server with the run and section IDs as parameters. A VRML string containing a population view annotation instance is delivered and added to the scene.

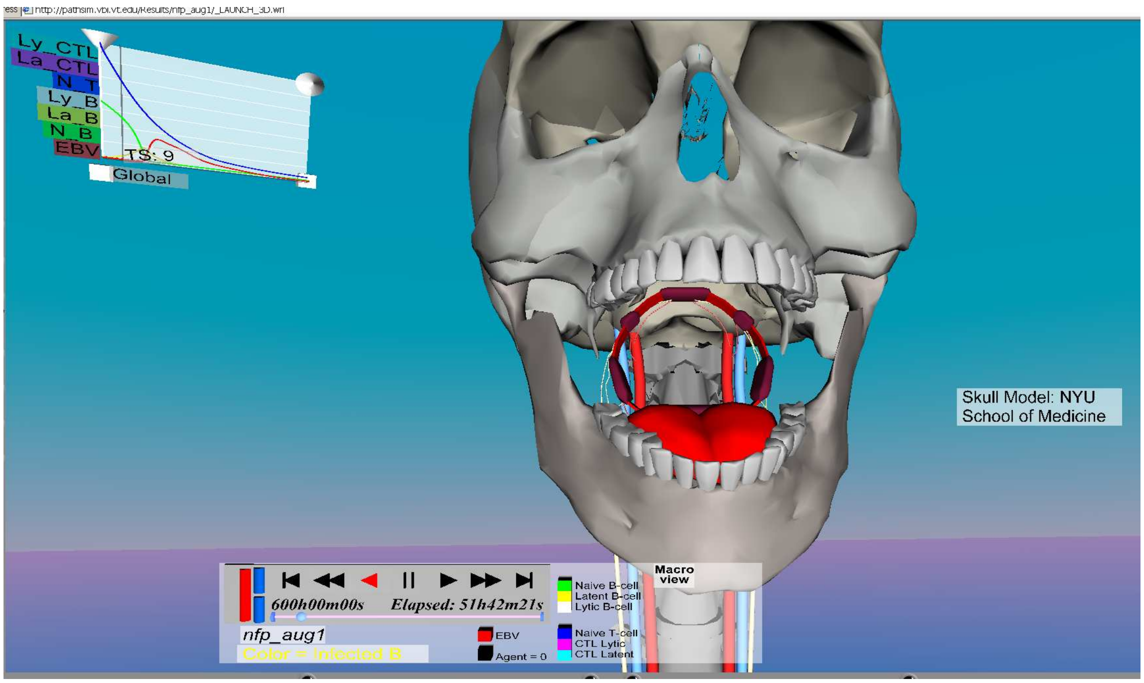


Figure 5.7: A Macro-scale view of PathSim environment and Heads-Up-Display including time controller, agent key, and global PopView.

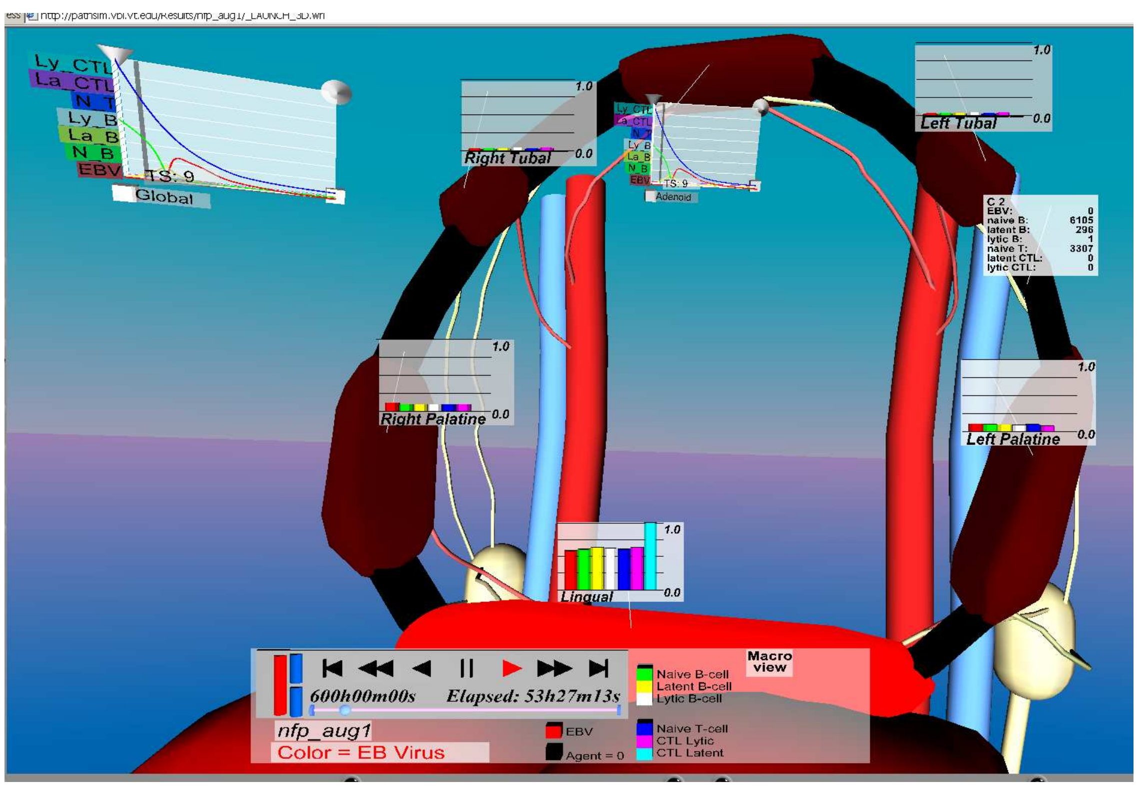


Figure 5.8: A Macro-scale view of PathSim results with agent colormap (Red = EB Virus) and tonsil PopViews

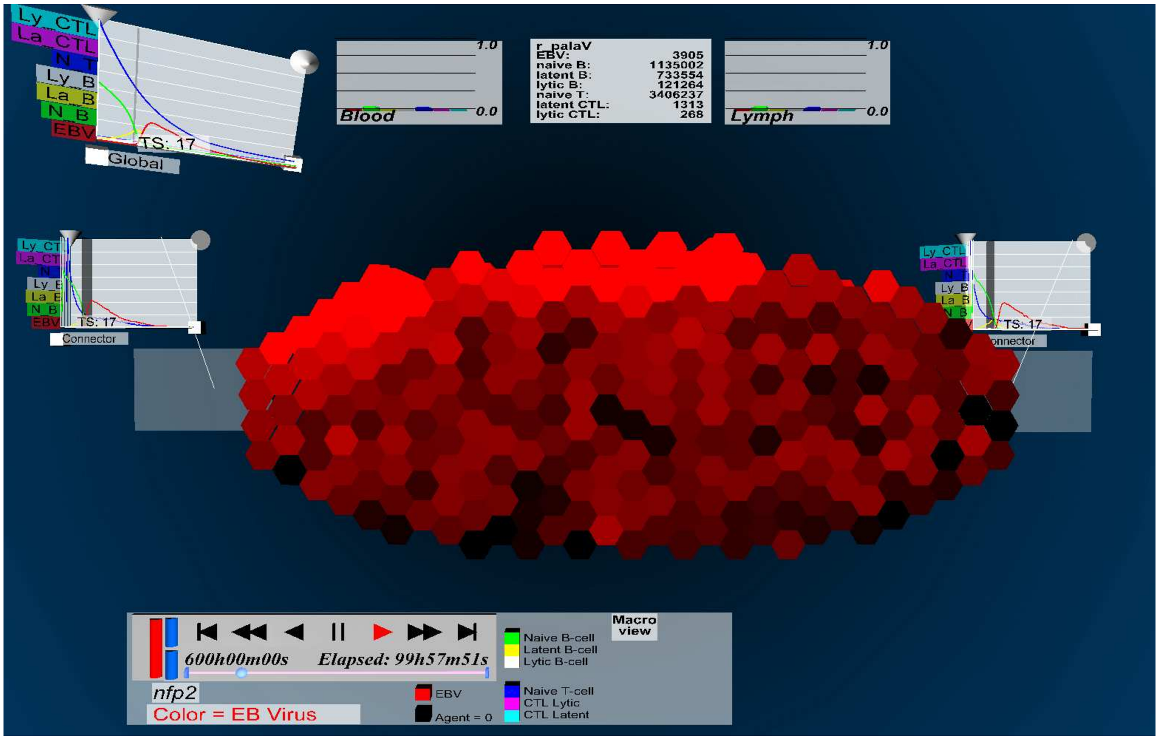


Figure 5.9: Zooming into the Right Palatine Tonsil and its adjacent connective tissue (Micro scale); the unit tissue colormap shows localized EBV population.

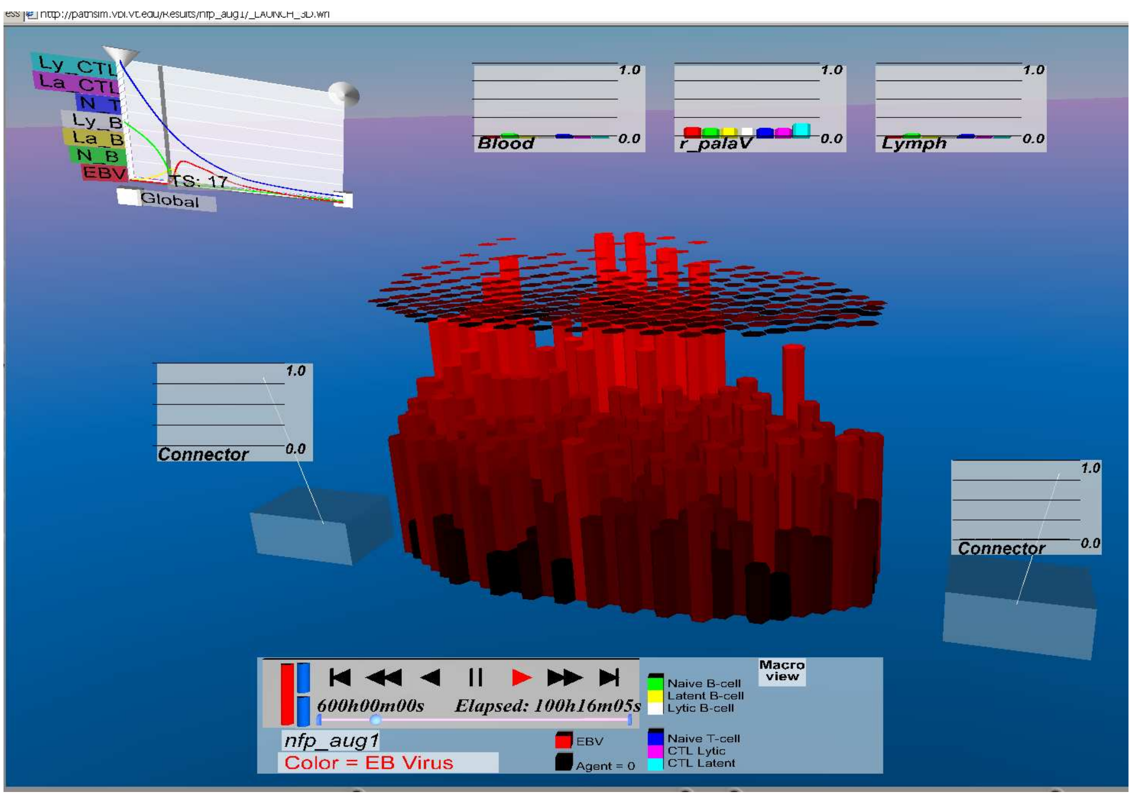


Figure 5.9: A Micro-scale view of an infection in the Right Palatine tonsil; note HUD now includes the overall PopView for the tonsil and Blood and Lymph populations (at top)

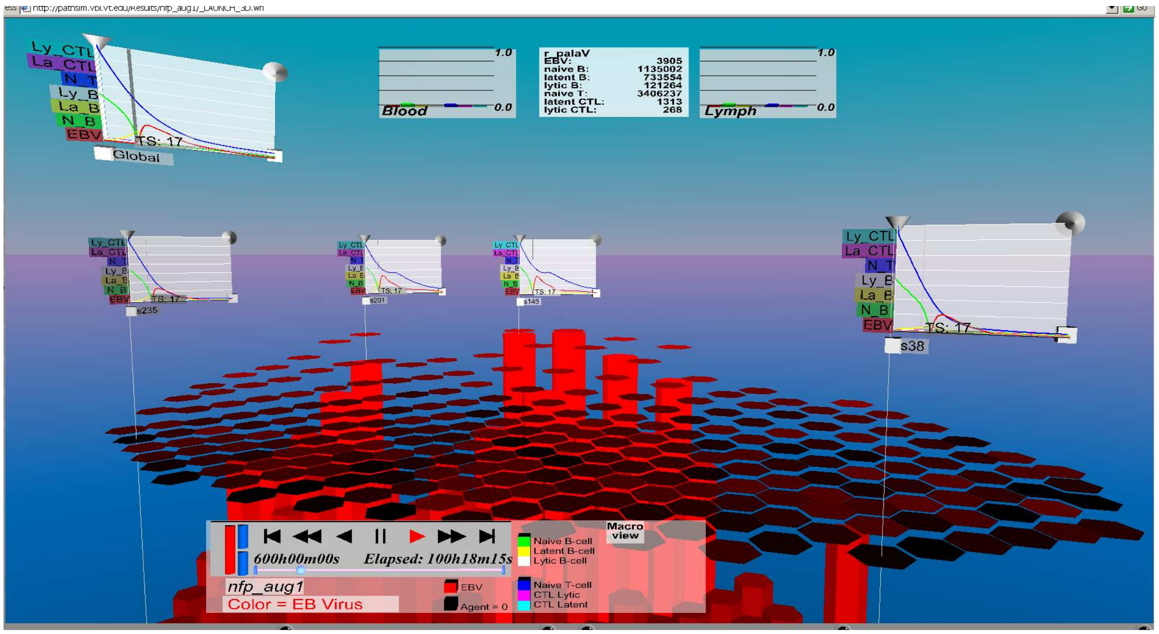


Figure 5.10: Zooming into the Micro-scale view of the infection in the Right Palatine tonsil; note tissue section Popviews retrieved on-demand from the PathSim server

5.4.2 Semantic Objects

Recently in our IRVE research, we have implemented a set of IRVE behaviors encapsulated as Semantic Objects for VR scenegraphs [Bowman, 2003a; Polys, 2004b]. Semantic Objects are a conceptual and programmatic abstraction of spatial objects in the visual space of the IRVE that include their associated information along with their geometric and appearance information. We describe Semantic Objects in detail in Chapter 3. The advantages of defining annotation information and display behaviors along with the objects are briefly: they encapsulate metadata and interaction events under a unique identifier, a central ‘layout manager’ is not necessary, and display behaviors are in the scenegraph and operate independently of the display’s size and resolution. This display independence has made it possible to deploy Semantic Objects and annotation objects across desktops, HMDs, Domes, and the CAVE.

Annotations

We have defined a class of rendering objects we refer to as ‘Annotations’. These annotations are Prototype objects for the display of abstract information in the scene. These annotations are designed to represent multiple data types and are described in detail in Chapter 3. For PathSim, we use both structured (field-value pairs) and unstructured text, and bar-graph and line-graph annotations. The server Visualization Processors write output data to the Prototype’s exposedField interface and ROUTE `fraction_changed` events to drive the annotation’s animations. During runtime, these annotation renderings can then be driven by the visual simulation’s timestamp.

The exposed functionality of the text annotation panels allows authors to specify typographic parameters along each of the VRML attributes (i.e. font family, style, color etc.). These appearance and typography parameters on text display objects give IRVE designers flexibility to define the visual characteristics of text labels or field/value pairs across a range of environments. For example, in order to aid text legibility across a wide variety of scenes, text panels may be instantiated with or without a label background whose color and transparency may be specified. While the text’s background is automatically sized to the number of lines and characters in the MFString, this is a platform-specific feature since VRML does not give authors script access to a string’s rendered extent. We proposed this feature to the Web3D Consortium’s X3D Working Group and the functionality is included in X3D Amendment 1.

In PathSim, we manage multiple views of the dynamic population values through a higher –order annotation called a 'PopView' (population view). A Popview is an interactive annotation that provides three complementary representations of the agent population. The representations can be switched through in series by simple selection (e.g. Figure 4.8). The default view is a color-coded bar graph where users can get a quick, qualitative understanding of the agent populations in a certain location at that timestep. The second is a field-value pair text panel, which provides numeric readouts of population levels at that timestep. The third is a line graph where the population values for that region are plotted over time.

Layout Behaviors

There are a number of principal parameters on Semantic Objects and their combined functionality can aid authors in mitigating the aggregation, association, density and tradeoffs in IRVE design. First, separate 'level of detail' groups can be defined for the object geometry and the annotation information; this insures the capability for designers to aggregate referring information independent of the object's levels of detail. In VRML, the `LOD` node is defined as a suggestion to the browser for optimization. For Semantic Objects, we implemented our own LOD logic that would guarantee the switching of children based on proximity and output an SFInt32 `level_changed` event to alert others in the runtime which child was active (being drawn). We proposed this `LOD` feature to the Web3D Consortium's X3D Working Group and the functionality is included in X3D Amendment 1.

Second, Semantic Objects may show their annotations when a user toggle-selects the object. This can be used to pop-up (or hide) information panels for secondary anatomical structures. In PathSim, we set all annotations to 'off' at load time. Third, a Semantic object's abstract information display can be associated to the geometry by way of the Gestalt connectedness principle - such as a drawn line (i.e. [Ware, 2000]). For PathSim, we use a simple line connector. Fourth, the scaling of the annotation group is a function of user visibility and proximity with options for fixed size, periodically sized, or continuously sized. In PathSim, we deploy periodically-sized annotations.

Finally, our abstract display objects act as true 3D Billboards insuring legibility from any viewing angle. Our set of Semantic Objects includes layout algorithms that vary the spatial location of the annotation group relative to the object. The display location of the annotation is typically a function of the user's position and viewing angle to the object. The details of our layout algorithms are described in Section 3.5. In PathSim, macro scale annotations are rendered with the relative rotation technique (Figure 4.8); at the micro scale, annotations are rendered with the relative position technique (Figures 4.9, 4.10).

5.4.3 MFSequencers

In order to drive data to the various visualization components in PathSim Visualizer, we wrote a set of Sequencer nodes that derive their interface from the abstract *X3DSequencer* node type. These nodes output discrete, Multi-Fielded (MF) events along a timeline. We introduced the integer field `batch` in order to specify how many values are in the eventOut array. Consequently, the number of keyValues must be evenly divisible by the batch value. We have implemented the MFStringSequencer and MFFloatSequencer to drive data to abstract information display objects such as text panels and bar-graphs. The string and float `keyValues[]` are populated from the simulation results during the visualization processing.

In a given simulation run, the duration and time intervals for evaluation are the same for all objects' Sequencer and Interpolator animations. In order to keep the resulting file size down, the processing script first writes a file containing the animation prototype declarations, each with the same key[] field. When the processing script instantiates the animation node into the result files, all that need be specified is the `keyValue []`.

5.4.4 Heads Up Display

Finally, we defined a generic Heads-Up-Display (HUD) for user-fixed controls and global and macro level abstract information (Figure 4.7). We used a simple `ProximitySensor` setup, routing position and orientation to the HUD parent. The HUD can take a set of children and an offset that specifies the distance from the user's active camera. While extremely useful for maintaining visibility of overview information and system controls, the HUD in this implementation has some drawbacks. Most important are the facts that the HUD is rendered with the rest of the scene, and that browsers vary on where they implement the near clipping plane. In cases where users have zoomed into very small scales, objects may actually come between the user and the HUD geometry.

5.5 Summary and Future Work

Through the PathSim project, we have implemented a number of custom information and interaction objects meeting the requirements of Systems Biologists to explore multi-scale, heterogeneous information. These scenegraph objects attempt to resolve tradeoffs on the dimensions of the IRVE design space [Bowman et al., 2003]. In the process of implementing these objects, we have discovered deficiencies and opportunities in current Web3D standards languages. In the process of parameterizing and deploying these objects, we note the lack of design guidelines for annotation layout.

In our formative evaluations and through participatory design, each of the scenegraph objects described in section 4.4 (Nested Scales, Annotations, Semantic Objects, MFSequencers, and Heads-Up-Display) has been identified as distinct and usable in our application across a range of platforms including desktops, HMDs, and Domes. Some of the functionality, as implemented in VRML/X3D, has known limitations (such as the HUD clipping problem).

In review, in IRVEs there are at least three principal possibilities for how abstract information is related to spatial information:

- abstract information varies continuously across the space
- abstract information is embedded/associated with points/regions in spatial data;
- the structures of the spatial data and the structured abstract data are mutually interlinked

The first is a technique widely used in scientific visualization or visualization of population/census data. If the abstract data is structured by the spatial data, the data values are a function of the space. In PathSim, the color and tissue animations per anatomy and the nested scales fall into this category. The multi-fielded Sequencer nodes fit easily into X3D paradigm and could be candidate nodes for future standardization. The Nested Scales functionality is addressed by new X3D capabilities such as IMPORT/EXPORT where Inlined worlds can communicate events with their parent world.

The second relation can take the form of visual items (pop-up labels, hyperlinks) where the abstract data is related to localized objects in the space. For example, a text description of an organ of the human anatomy or a numerical description of an atom or molecule. This functionality, as defined in our Semantic Objects, provides high-level user interface behaviors that may be collected into an online resource (i.e. PROTO library).

The third IRVE relation, where the abstract data and spatial data each have a structure of their own, should be common. In PathSim, this would be extending our concept of the PopView Annotation to other information representations and tools. For example, defining 'Application Surfaces' where the windows of other analytic tools can be mapped to a pickable 3D surface. This seems most appropriate to pursue in the Compositing component interface. Such functionality is extremely desirable, especially when the application and content is loaded into an immersive system. Previously, we have implemented display-fixed prototypes using the DIVERSE toolkit and XWindows for a molecular IRVE application in the CAVE [Polys, 2004a]. Further work in this vein must address and resolve operating system, software, and hardware architectures.

The feature summary as implemented for PathSim is shown in Table 5.1. In conjunction with the X3D Specification Working Group, we are developing future components as a foundation to address these interface requirements. These include the Annotation Component, Layers component, Layout Component, and the Compositing Component (in progress).

A proposed Annotation Component for example, would provide better support for the functionality we encapsulate in Semantic Objects. In the proposed component, associated information lives as geometry in a coordinate space parallel to the display surface; there is a reference point, an offset, and a connection point that can be connected by a lead line.

While some browsers can support ‘Overlays’ or rendering ‘Layers’ (Xj3D and BitManagement respectively), the interoperability problem can only be solved through improvement of the standard. The Layering and Layout Components would allow more sophisticated author control over rendering (i.e. Z-order, clipping, screen position, etc.). This would improve support for Heads-Up-Displays, which are common in applications, but awkward between browsers.

Future work for IRVE research includes further exploration and optimization of these object/information behaviors through formal usability evaluations. Some display components may be proposed as future components for the X3D standard. For PathSim itself, we intend further integration of information resources such as published biochemical and cellular models, new multi-scale data and visualization architectures, and interface improvements such as analytic tools and indexing through the abstract data. Future bioinformatics research will involve using PathSim with other anatomical models, mesh generation techniques, and pathogen agents.

Information Feature	**Information Location**	**Information Association**	**Information Aggregation**
Agent Population View at Macro and Micro Scale: **Agent Heatmap and height field mapping**	Object’s perceptual properties	Identity	Low
Agent Population View at Macro and Micro Scale: **Annotation per object/unit**	Object space	Proximity, Connectedness, Common Fate; Occlusion, Motion Parallax	Low
Agent Population View at Micro scale: **Macro Annotations**	User Space	Common Fate	High
Agent Population View at Macro and Micro Scale: **Global Annotation**	User Space	Common Fate	Highest
Time Read-out and Controller	User Space	Common Fate	Low
Links & References	Display Space	Common Fate	High

Table 5.1: PathSim Design Features and IRVE Design Dimensions

6. Comparisons of Layout Spaces

In an Information-Rich Virtual Environment, there may be a wealth of data and media types embedded-in or linked-to the virtual space and objects. Users require interfaces that enable navigation between and within these various types. The design challenges and techniques of integrated information spaces boil down to the problem of combining the techniques of virtual environments (VEs) and information visualizations (InfoVis). Specifically, the goal of this research program is to understand the tradeoffs in the IRVE information design space concerning fundamental IRVE activities such as Search, Comparison, and Finding Patterns and Trends [Polys, 2004b; Polys, 2004c].

While supporting information architectures and runtime systems are required, the crucial issue remains one of design: how can IRVE interfaces present and manage the volume and diversity of information in a comprehensible way? How can applications support users in relating abstract and spatial information, and how can they use those relations to understand patterns or trends within and between the respective data types?

Our research has focused on two important tradeoffs in IRVE information design: the Association-Occlusion tradeoff and the Legibility-Relative Size tradeoff. From a perceptual standpoint, the visual coupling of annotations to their referent relies on both Gestalt and Depth cues in the visual buffer. What are advantageous visual configurations that reduce cognitive overhead by facilitating the perceptual binding of annotation and referent? Are these advantageous configurations task or display-specific?

In chapter 2, we discussed the growing literature on multimedia illustration and learning processes. We also seek to understand how to render abstract information in relation to spatial/perceptual information. In this work on IRVEs however, we are interested in the principal visual design tradeoffs that exist between the *Gestalt* (Grouping) and the *Layout Space* dimensions (see Chapter 3).

In IRVEs, annotations may reside in a number of coordinate systems, what we term the 'Layout Space'. As mentioned above, these are: World, Object, User, Viewport, and Display spaces. Table 6.1 shows our initial design dimensions. High Association and High Occlusion reside in the top left corner. Low Association and Low Occlusion reside in the lower left corner.

	Common Region	Proximity	Connectedness	Common Fate	Similarity
Object	x	x	x	x	x
World	x	x	x	x	x
User	x	x	x	x	x
Viewport	x	x	x	x	x
Display	x	x	x	x	x

Table 6.1: The orthogonal Layout Space and Association dimensions in IRVE design

The evaluation we describe in this paper seeks to understand the visual design tradeoffs that exist concerning the layout of abstract information in relation to its referent object in the virtual environment. Specifically, we are interested in the Association-Occlusion tradeoff. This tradeoff occurs because the stronger the visual cues of association between object and label, the more of the spatial environment is occluded; the less visual association, the less occlusion. This tradeoff can be summarized by the following design claim:

More consistent depth cues and Gestalt cues between annotation and referent

> **(+)** May convey more information about the relation between annotation and referent (i.e. less ambiguity)
>
> **(-)** May cause result in more occlusion between scene objects and therefore less visibility of information

There are many combinations of display techniques that are possible within this design space. What are the factors of IRVE display techniques that make one combination better than another? In order to understand the strengths and contributions of these different dimensions for relating abstract and spatial information in Search and Comparison tasks, we have run a set of experiments, which are summarized in the following sections.

The first experimental evaluations are described in this chapter and compare display techniques *between* layout spaces in desktop and large-screen situations. Thus, these experiments provide a broad sampling of the usability of IRVE display techniques. Section 6.1 details our investigations of Object versus Viewport Space; Section 6.2 details our pilots with Display space and an evaluation of Object versus Display space.

6.1 Experiment 1: Object Space vs. Viewport Space

The goal of this evaluation was to understand the usability of annotation layout spaces across different display sizes and different Software Fields of View (SFOVs). This work has been initially published in [Polys et al, 2005]. Specifically, in this experiment, we were interested in the perceptual cues provided by two different layout spaces and their tradeoffs for performing fundamental types of tasks across different monitor configurations (1 and 9) and different projection distortions (60 or 100 degrees of vertical angle). The monitor configurations used in this experiment are shown in Figure 6.1.

Figure 6.1: Single and nine-screen display configurations used in this experiment

Questions we set out to answer with this experiment include:

- "Is a layout space with guaranteed visibility better than one with tight spatial-coupling for certain tasks?"
- "Do the advantages of one layout space hold if the screen size is increased?"
- "Do the advantages of one layout space hold if the SFOV is increased?"

The two layout spaces we examine in this research are termed: '***Object Space***', in which annotations are displayed in the virtual world relative to their referent object, and '***Viewport Space***', in which annotations are displayed in a planar workspace at or just beyond the image plane. In Object Space, abstract information is spatially situated in the scene, which can provide depth cues such as Occlusion, Motion Parallax, and Linear Perspective consistent with the referent object; in addition, the annotation and referent are visible in the same region of the screen (Gestalt Proximity).

The Viewport space, in contrast, is a 2D layout space at or just beyond the near-clipping plane. As such, annotations and geometry in the Viewport space are rendered last and appear as over-layed on top of the virtual world's projection. Annotations in Viewport Space typically do not provide depth cues consistent with their referents, but do provide guaranteed visibility and legibility of the annotation.

The results of this empirical evaluation provide insight into how IRVE information design tradeoffs impact task performance and satisfaction and what choices are advantageous under various rendering distortions. In addition, this evaluation addresses the problem of how designers should consider the transfer of IRVE interfaces between single-monitor and multiple-monitor displays.

6.1.1 Information Design

Object Space

One existing layout technique, termed ‘Object Space’, is to locate the abstract information in the virtual world and in the same coordinate system as its referent object. By co-locating the enhancing information with its referent object in the virtual space, this technique provides depth cues that are consistent with the referent object; if the object is moved or animated, the annotation is moved or animated, thus giving a tight visual-coupling between annotation and referent.

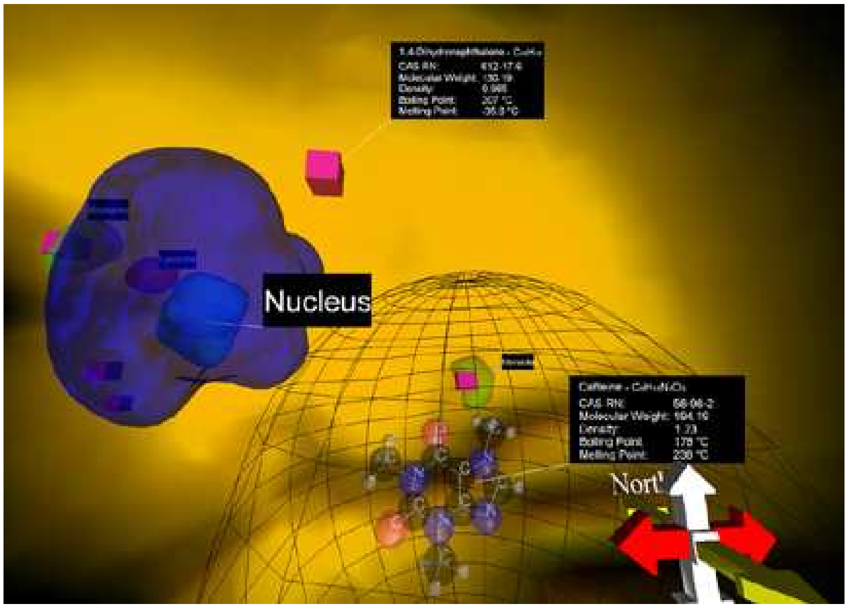


Figure 6.2: The Object Space IRVE layout technique

In Gestalt terms, Object Space can provide strong association cues including Connectedness, Proximity, and Common Fate [Ware, 2000]**.** However, there are some limitations to Object Space, especially for Search and Comparison tasks. For example, when using Object Space layouts, not all labels may be visible at once and spatial maneuvering may be required to make them visible as well as legible. In addition, when comparing abstract information that is rendered as a graph for example, the effects of the Perspective depth cue can make comparison difficult. Figure 6.2 shows an example of the Object Space layout technique used in a 3D cell model.

We have previously described software objects that encapsulate a number of IRVE layout behaviors – Semantic Objects (sections 3.5 and 5.4.2). These Semantic Objects allow the specification of multiple levels of detail for both objects and their labels, which enables proximity-based filtering on either type. Labels may be located in the object’s coordinate system through a number of means including: Fixed Position, Relative Position, Bounding Box, Screen-Bounds, and Force-Directed methods. In addition, Semantic Object labels can be: billboarded to always face the user and maintain upright orientation, connected to the object with a line, and scaled by user distance through a number of schemes (such as None, Periodic, and Continuous).

Viewport Space

To address the limitations of Object Space layouts, we designed and implemented a new IRVE interface we call the ‘Viewport Workspace’ where a selected object’s label is toggled into a Heads-Up-Display at the image plane where it is always visible regardless of the user’s position and viewing orientation. In the software definition of our interface, we maintain a pixel-agnostic stance and scale and locate labels according to parameters of the environment’s projection (rendering). Labels are sized and located in world units relative to the specified Software Field of View (SFOV) and the distance to the near-clipping plane.

In the Viewport Workspace, labels can also be connected to their referent objects with lines extending into the scene. The layout of labels in the 2D Viewport space is managed by a parameterized BorderLayoutManager script. Like its Java Swing inspiration, the BorderLayoutManager divides the layout space into 4 regions or containers: North and South, which tile horizontally across the top and bottom, and East and West, which tile vertically on the right and left sides.

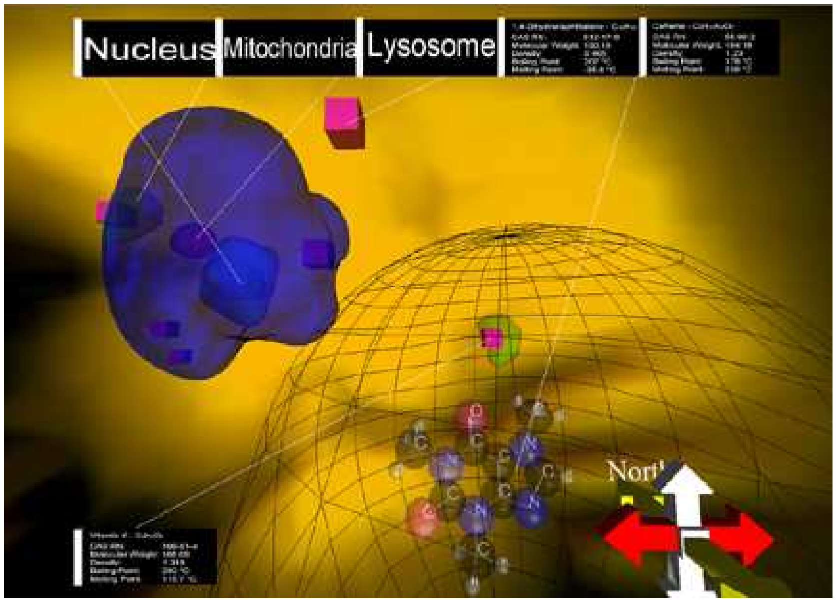


Figure 6.3: The Viewport Space layout technique

The Viewport Space BorderLayout we defined can be specified with container capacity and the fill order for the four directions using the BorderLayoutManager. In this particular instance, the location of any given label is determined by the order in which it was selected. Subsequent flavors were developed for the Viewport space experiment described in Section 6.2. In addition, we added hooks for an extra transformation that allowed users to select and reposition (click and drag) annotations in the HUD Viewport workspace. Figure 6.3 shows an example of the Viewport Space layout technique used in a 3D cell environment.

By providing a pixel-agnostic layout space and manager at the image plane (layout positions are not described in pixels), we can scale labels and containers to the size of the display and projection. For example, we may only be able to fit a half-dozen labels legibly in one container on a single-screen display. However when we render that same interface on a nine-screen display, the labels scale proportionately and also become larger. Using our Viewport Space approach, we can easily adapt the label scale and container capacity to display the labels at an equivalent pixel size as on the single-screen via the scenegraph. On a nine-screen display and holding pixel size constant to the value on a single-screen, we can get approximately three times as many labels in one container.

Field Of View

In understanding how humans perceive a virtual environment on a particular display, the concept of Field of View (FOV) is essential. For desktop displays we can describe at least two important kinds of FOVs: the Display Field of View (DFOV) and the Software Field of View (SFOV). DFOV refers to the amount of visual angle that the physical display surface occupies in the user's visual field: a nine-screen display offers approximately three times more DFOV angle than a single-screen when viewed from the same distance. For example, a 17-inch monitor viewed from 65 cm provides a 22.5˚ vertical DFOV; three stacked 17-inch monitors viewed from the same distance provides a 61.7˚ vertical DFOV. It follows that a larger DFOV will require larger saccades and head movements for users to traverse it visually.

The SFOV on the other hand, refers to the viewing angle of the camera on the virtual scene, which is rendered (projected) onto the display surface. Larger SFOV values create fish-eye effect while smaller values create tunneled, telescoping effects. We decided to fix SFOV to two levels for our experiment: 60˚

vertical SFOV (which approximately matched the nine-screen DFOV) and 100˚ vertical SFOV to assess any impact on the performance of search and comparison tasks.

Formative Evaluation

An informal pilot study was performed to understand how users perceive and interact with our IRVE interfaces in different SFOVs across the different monitor conditions. The goal of the formative evaluation was to find initial values of SFOV and drag mappings for the full study. Users were given a large-scale virtual model of the Giza plateau and given 7-10 minutes to learn the navigation controls of the VRML browser. On standards-compliant engines for VRML/X3D, the SFOV is defaults at 45˚ (.785 radians) measured along the shortest screen projection (typically the *vertical*).

When subjects were comfortable with the navigation interface, the initial designs of Object Space and Viewport Space annotation layouts were presented to two users from the study pool, each on both screen configurations. The layout techniques were presented in a cell environment like those used in the later full study. Subjects used the up and down arrow keys to dynamically increase or decrease the SFOV as desired.

Pilot Results

In the cell environment, novice users were able to tolerate much higher SFOVs than we had anticipated. The average across all interface layouts and display sizes was 90.5˚ (1.58 radians) vertical. On the single-screen the average SFOV was 5.1 times the DFOV, while on the nine-screen, the average was 1.4 times the DFOV. Still, there is not enough statistical power to draw any real conclusions here. In addition, the user tendency to high SFOVs is interesting because in a cell environment there are few, if any, sharp edges or 90 degree angles.

More interesting perhaps were user strategies with a dynamic SFOV control. In the Object Space layout, Users increased the SFOV to gain overview spatial information and also increased the SFOV to recover detail abstract information (when it was just out of view for example). In addition, Users decreased the FOV to focus in or telescope to targets in the projection; however users sometimes confused reducing the SFOV to actually navigating to the target.

In the Viewport Space layout, users increased the SFOV control to gain overview spatial information and then had to decrease it to make detail abstract information legible. Users' association of annotation to its referent appeared to have a strong temporal component. For example, when looking up information, users commonly oriented to labels' appearance or disappearance on screen as a result of selection / de-selection rather than tracing the connection lines between objects and labels. This suggests that common fate is a strong association cue in Viewport Space. Finally, users did not identify the dragging affordances of the annotations on the Viewport workspace (even though the cursor changed from a directed arrow to a hand icon).

The initial data and observations were used to improve the IRVE layout prototypes for the final study. This included choosing two levels of SFOV condition that were higher than the VRML default. Once the target SFOVs values were chosen, all mouse drag mappings were calibrated between the interface layouts on all display sizes and SFOVs. In addition, we added a handle bar to the Viewport Space labels to emphasize their drag-ability.

6.1.2 User Study

To test the relative effectiveness of our IRVE layout spaces across displays and task types, we designed an experiment to test the following hypotheses:

- *Hypothesis* 1 : With its guarantee of visibility and legibility, the Viewport workspace should provide an advantage for search tasks as well as tasks involving comparison of abstract information. The Viewport workspace does not provide depth information and thus tasks involving spatial comparisons may be difficult.
- *Hypothesis* 2 : We hypothesized that the increased display size and corresponding spatial resolution of the nine panel display will be advantageous for tasks where exhaustive search and comparison is required because more information panels can be displayed at once.

- *Hypothesis* 3 : Higher software FOV will aid search tasks by including more of the scene in the projection. Higher software FOV will hinder some spatial comparison tasks due to fish-eye distortion.

Participants

Participants were drawn from the international graduate population of the College of Engineering. There were 11 males and 5 females. 10 of the 16 subjects wore glasses or contacts and all of the subjects used computers daily for work. 81.25% of the subjects also used computers daily for fun and the remainder used them several times a week for this purpose. All subjects had at least a high-school level familiarity with cell biology. Two subjects were known to be actively working as Research Assistants on bioinformatics projects and they were assigned to different display groups.

Subjects self-reported their familiarity with computers: 87.5% reported 'very familiar' with the remainder reporting 'fairly familiar'. 31.25% of the subjects reported not having used a 3D VE system before. Of those that had, 63.6% had used immersive systems such as HMDs or a CAVE; the remainder had used desktop platforms only, typically for 3D games.

Equipment

We used a cost-effective large display system consisting of nine tiled normal PC monitors supported by five dual-head peripheral component interconnect (PCI) high-end graphics cards on a 2.5 GHz Pentium 4 PC. With the support of Microsoft Windows XP operating system's advanced display feature, we could easily create an illusion of a single large screen without using any special software and hardware. The dimension of the 9-screen display is 103.6 cm x 77.7 cm in physical size with 3840 x 3072 = 11,796,480 pixels. The dimension of the small normal display is 35.5 cm x 25.9 cm in physical size with 1280 x 1024 = 1,310,720 pixels. Subjects were seated at a distance of 60 – 70 cm from the screen with their heads lined up to the center monitor.

Content & Domain

Environments built for the study were based on a 3D model of a cell and its constituent structures (e.g. nucleus, mitochondria, lysosomes). These objects provided landmarks within the cell and a basis for showing spatial relationships such as 'next-to', 'inside-of', etc. All cellular structures were defined with different levels of detail so that from far away they appeared semi-transparent, but within a certain distance were drawn as wireframes. In this way, when a user got close enough to a structure, they could pick (select) objects inside of it.

For each trial, a set of 3D molecules was shuffled and arbitrarily located in the various structures of each cell including the cytosol; these were the targets for the search and comparison tasks. In each cell environment there was a nucleus, a nucleolus, three mitochondria, two lysosomes, and 13 molecules (all organic and with a molecular weight of less than 195). Since molecular scales are a few orders of magnitude smaller than cellular scales, molecules were represented by pink cubes when the user was far away; the molecular structure was drawn when the user got within a certain distance. Each cell structure was labeled with its name and each molecule's label included its name, formula, molecular weight, and melting and boiling point.

The choice of a cell model as the content setting was made for a number of reasons. First, there is a wealth of organic chemistry data suitable for IRVE visualization [NIST], [Murray-Rust, 2001] and its natural representation is in a biological context. Second, in these contexts there is no horizon, and requirement for physical constraints such as gravity, landmarks and targets are distributed in all three dimensions, making effective annotation layout and navigation a challenge. Third, education researchers [McClean, 2001] have shown improved student performance by augmenting science lectures with desktop virtual environments including the 'Virtual Cell' environment for biology and the processes of cellular respiration[Saini-Eidukat, 1999; White, 2002]. It is our hope that our interface design lessons may be directly applied to biomedical research and education software.

For each task, landmarks and targets in the cell model were shuffled to insure naïve search for every trial. Regardless of independent variable conditions, each environment had identical mappings of mouse movement to cursor movement, and picking correspondence was maintained for navigation, selection, and manipulation interactions in the virtual environment. In addition, all environments included an identical HUD compass or gyroscope widget, which helped the user maintain their directional orientation within the cell. All interface components are realized entirely in VRML.

Information Design Conditions

In both the Object Space and Viewport Space layouts we devised, labels were always drawn 'up' regardless of the user's orientation. All labels were connected to their referent objects with a drawn white line (Gestalt Connectedness). In both the Object Space and Viewport Space layouts, label size was determined by the minimum label size for text legibility on a 1280x1024 display, in this case 206x86 pixels.

In the Object Space conditions, labels were located relative to their referent object and offset orthogonally by a set distance (Relative Position). When toggled on, Object Space labels were Periodically scaled according to user distance. This scaling was established to guarantee a minimum size for legibility from any distance. The actually-rendered label size (when viewed head-on) could vary between 1 and 1.2 times the pixel area of a label in the Viewport condition depending on the distance to the object. In making this choice, we remove the depth cue of Relative Size in favor of access to the abstract information contained in the label. The Depth cues of Occlusion and Motion Parallax remain. Because Object Space labels are co-located with objects in the virtual world, they are subject to the same magnification and distortion problems as other objects in the periphery of the projection. As a result, a label may appear to stretch (keystone) and scale as it moves away from the line of sight.

In the Viewport Space condition, we used the BorderLayoutManager described above; the container fill order was set to ['N', 'S', 'E', 'W']. The minimal legibility sizing meant that five labels could fit in any given container on the single-screen. As mentioned previously, when a Viewport Space is rendered on a nine-screen display, its projection is simply scaled up. In order to understand how the properties of larger screens affect usability, we decided to keep the label's pixel size constant. This means that we could now fit fifteen labels in a given container on the nine-screen. While we realize this may be a confound to some degree, it allows us to if we can improve Viewport performance by leveraging the larger screen size (with constant spatial resolution).

The relationship of perceptual cues in the conditions tested is shown in Table 6.2. High Association and High Occlusion reside in the top left corner. Low Association and Low Occlusion reside in the lower left corner. Videos depicting the techniques tested are listed in Appendix H.

	Proximity	**Connectedness**	**Common Fate**	**Similarity**	**None**
Occlusion	**O**	**O**	**O**		
Motion Parallax	**O**	**O**	**O**		
Relative/Size / Perspective					
None		**V**	**V**		

Table 6.2: Depth and Gestalt Cues presented by Object (O) and Viewport (V) Space layouts used in Experiment 1

Tasks

In order to test how our IRVE layout techniques impact usability for search and comparison, we define four kinds of tasks (below). The task types are denoted by the following convention:

[IRVE_TaskType: informationCriteria -> informationTarget]

IRVE Search Tasks [S:*] require subjects to either:

- Find a piece of abstract information (A) based on some perceptual/spatial criteria (S).
  Task example [S:S->A]: 'What molecule is just outside of the nucleolus?' , or
- Find a piece of perceptual/spatial information (S) based on some abstract criteria (A).
  Task example [S:A->S]: 'Where in the cell is the Pyruvic Acid molecule?'

IRVE Comparison Tasks [C:*] require subjects to either:

- Compare by some spatial criteria (S) and determine an abstract attribute (A).
  Task example [C:S->A]: 'Find the lysosome that is closest to a mitochondria. What is the melting point of the molecule in the lysosome?', or
- Compare by some abstract criteria (A) and determine a spatial attribute (S).
  Task example [C:A->S]: 'Where in the cell is the molecule with the lowest melting point?'

Experiment and Method

We used a mixed design for this experiment (Table 6.3). Subjects were randomly divided into two groups for the between-subjects independent variable, which was the display size. One group performed all tasks on the single-screen display configuration and one group performed all tasks on the nine-screen display configuration. There were two within-subjects independent variables of two levels each: layout technique (Object or Viewport Space) and SFOV (60˚ or 100˚ vertical). For each condition, users were given one of each of the four task types mentioned above. Thus a total of 16 trials were presented to each subject in a counterbalanced order.

Users were introduced to each control mode of desktop VE navigation under the Cortona VRML browser. The metaphor was fixed to 'FLY' and users were educated and guided on how to use the plan, pan, turn, roll, go-to, restore, for control in the virtual world. Users were given the *Kelp Forest Exhibit* virtual environment, which is a 3D model of a large saltwater tank at Monterrey Bay Aquarium [Brutzman, 2002]. Users were directed to do things like, 'fly into the tank; turn to your right 90 degrees, is that a shark? Pan up to the surface; now down to the bottom; turn around; follow that diver …'. For the navigation portion of training, subjects took anywhere from 4 to 10 minutes to affirm that they felt comfortable with the controls.

Subjects were then given a sample 3D cell environment with all the common landmark structures they would see in the experiment. In this environment, they were shown how to toggle object labels and how the cellular structures and molecules behaved depending on their proximity. Finally, they were instructed on the nature of the tasks. When users affirmed that they felt comfortable with the cell environment (typically 3-5 minutes), the experiment began.

In each trial, users were timed and recorded for correctness. In addition, they were asked to rate their satisfaction with the interface for that task and the level of difficulty of the task on a scale of 1 to 7. One part of each of three Cognitive Factors tests was given to each subject before the experiment began: Closure Flexibility (Hidden Patterns), Spatial Orientation (Cube Comparisons), and Visualization (Paper Folding) [10]. This was intended to help understand the role of individual differences in utilization or preference of the various interfaces and to identify other possible causes for between-subjects effects. Experimental materials and result tables for this evaluation are included in Appendix B.

N=16; time, accuracy, difficulty, satisfaction		**Within Subjects**		
			SFOV = 60	SFOV = 100
Between Subjects	**1 screen Display**	**Object space**	**S**:S->A **S**:A->S **C**:S->A **C**:A->S	**S**:S->A **S**:A->S **C**:S->A **C**:A->S
		Viewport workspace 1	**S**:S->A **S**:A->S **C**:S->A **C**:A->S	**S**:S->A **S**:A->S **C**:S->A **C**:A->S
	16 conditions = 16 environments per subject;			
	9 screen Display	**Object space**	**S**:S->A **S**:A->S **C**:S->A **C**:A->S	**S**:S->A **S**:A->S **C**:S->A **C**:A->S
		Viewport workspace 2	**S**:S->A **S**:A->S **C**:S->A **C**:A->S	**S**:S->A **S**:A->S **C**:S->A **C**:A->S

Table 6.3: Experimental design for Object vs. Viewport experiment

6.1.3 Results

For each trial, the dependent variables collected were: time, correctness, and user ratings of satisfaction and difficulty. A General Linear Model was constructed for these results to determine any significant effects and interactions of the various experimental conditions to these metrics of usability. Paired Samples t-tests were used to find significant contrasts when interaction effects were found. A post-hoc analysis of the cognitive test scores using an independent samples t-test revealed that there was no significant difference between the two groups in terms of cognitive test scores.

Some general observations are notable. First, most users tended to search serially through the space in a lawnmower pattern and used position controls more often than orientation controls. Across layout spaces, some users tended to select, read, and deselect objects along the way rather than keep them visible and travel on. In general, this strategy results in less visual clutter but required repeated re-selection if they did not immediately recall the information. After one or two experiences with a more exhaustive search, users typically adopted the strategy of leaving selected annotations visible until they occluded or distracted from their search.

Accuracy

There was a significant main effect on user accuracy across all tasks for the layout technique. The Viewport interface (mean = 85.7%) performed better than the Object space layout (mean = 75.6%) at $F_{1,12} = 6.134$; $p = .029$. This result agrees with our first hypothesis and makes sense because with Viewport space, all active labels are visible and the HUD facilitates comparison. Because label size was controlled across levels, we know this is not a difference arising from legibility.

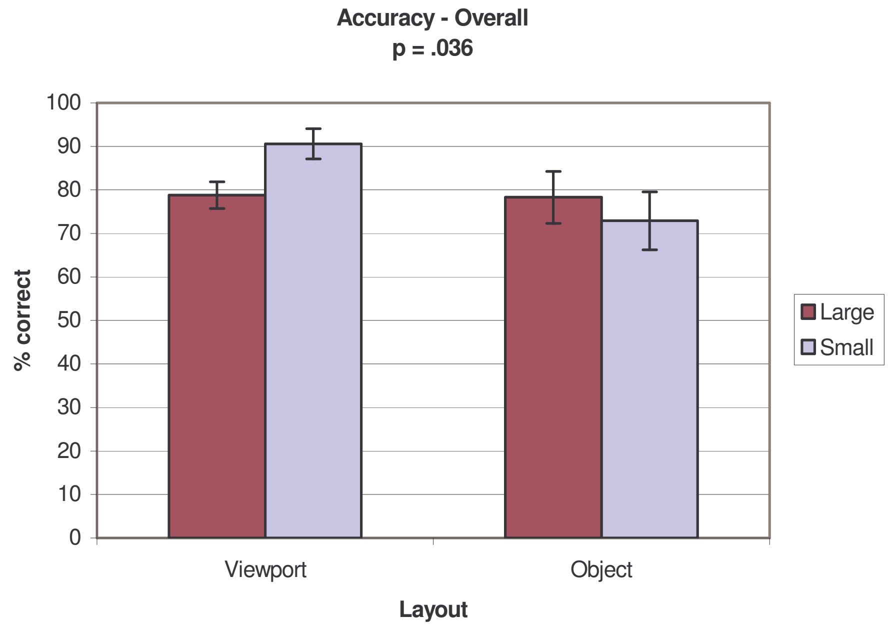


Figure 6.4: Interaction of Display size and Layout technique

There was a significant interaction between display size and layout technique ($F_{1, 12} = 5.587$; $p = .036$). the single-screen group performed better with the Viewport interface (89% vs. 79% correct) and this was significant ($t_{14} = 2.160$; $p = .049$). In contrast, users on the nine-screen display performed better with the Object Space layout (78% vs. 71% correct), but this was not a significant difference by t-test. Figure 6.4 shows this interaction. One explanation for this interaction effect may be due to the fact that on large display with BorderLayout HUDs, users require large saccades and head movement in order to follow a connector line between an object and its label in another part of the display. The tight coupling of Object Space may reduce errors by allowing the information to be matched in one fixation or short saccade.

In contrast, on the small display, there is little or no head and eye movement, connector lines are shorter and a given number of labels may be divided into more than one container. An additional advantage that Object space might have on the large display is that there is less occlusion between labels on the large displays.

There was also a significant interaction between layout, SFOV, and display. On the single-screen display, both techniques were roughly equivalent at small SFOV, but at large SFOV the Viewport interface provided a significant advantage. Figure 6.5 depicts the interaction of these three variables where $F_{1, 12} = 5.798$; $p = .049$. This interaction shows that Viewport space is clearly more effective than Object space (93.8% vs. 78.1% correct) in conditions with high projection distortion (large SFOV) and little screen space (small DFOV). T-tests reveal that this is a strong pair-wise effect at $t_{14} = 3.035$; $p = .009$).

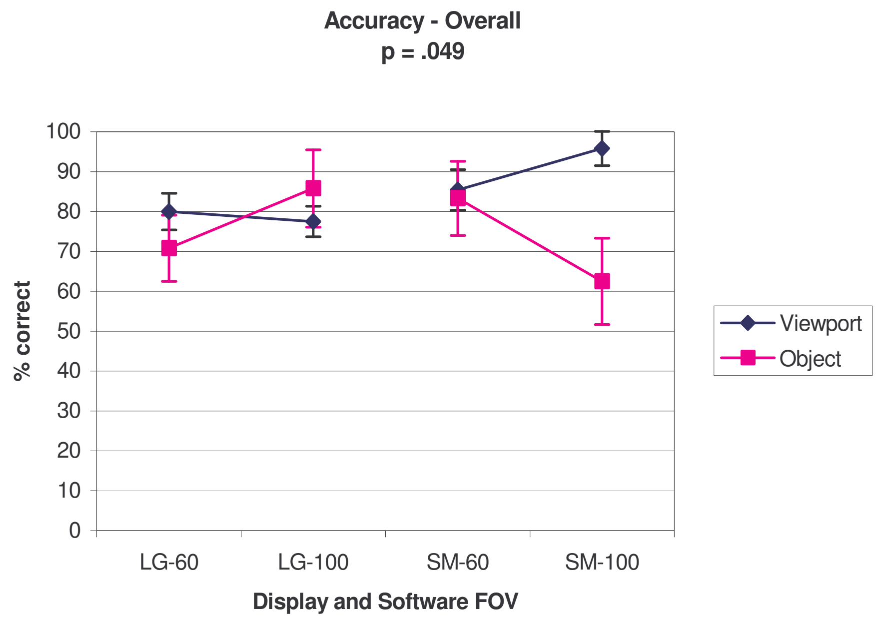


Figure 6.5: Interaction of Layout, Display, and SFOV variables

Task-specific Results

For Search tasks, there was a significant main effect for SFOV ($F_{1,14} = 7.56$; $p = .016$) with the high SFOV being more accurate (95.3%) than the low SFOV (81.3%). This result, which is shown in Figure 6.6, can be explained because with high SFOV, users can see more of the scene in the projection at any given time. For Comparison tasks, small SFOV was significantly more accurate and this was a main effect ($F_{1,14} = 5.61$; $p = .05$). This result also aligns with our hypotheses that Comparison tasks (especially those on spatial criteria) may suffer under visual distortion.

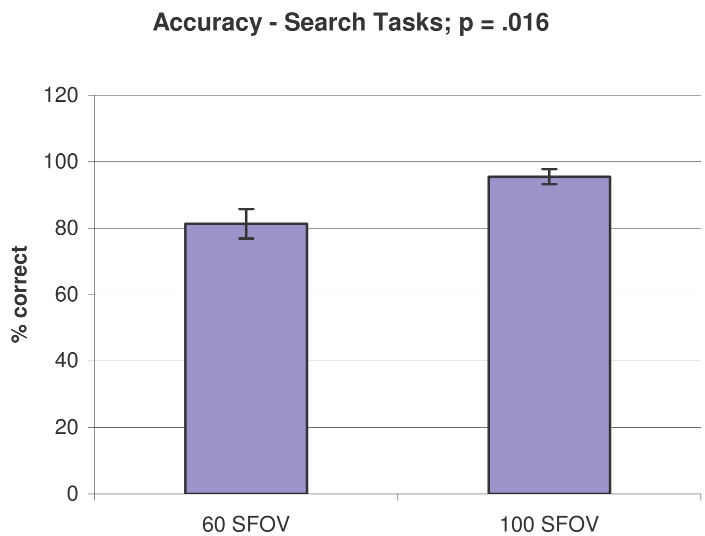


Figure 6.6: Main effect of SFOV for Search task accuracy

The interaction of Layout and Display variables was mostly due to relative performance on Comparison tasks (Figure 6.7). Here, Layout and Display were a significant combination $F_{1, 14} = 13.44$; $p = .003$. On the single screen display, Viewport space layout was significantly more accurate (87.5% vs. 62.5%) at $t_{14} = 3.742$; $p = .002$. The trend was reversed for the large display group where Object space was more accurate (71.9% vs 62.5%); however, this pair-wise difference is not significant by t-test.

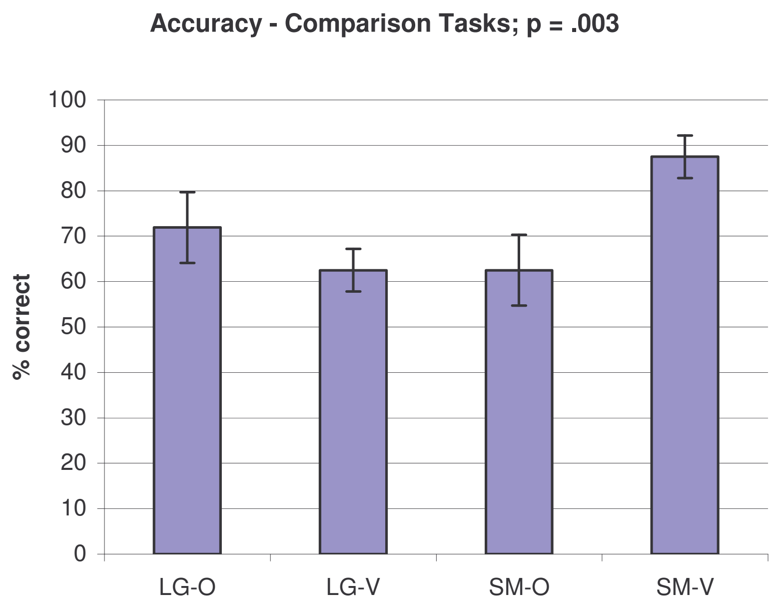


Figure 6.7: Interaction of Screen-size and Layout on Comparison task accuracy

Time

Subjects were timed from their first input event until the time they gave an answer they were confident in. The sum time to complete all 16 tasks was longer for the nine-screen group than the 1-screen group (32% longer), and this difference was almost significant ($t_{14} = .184$; $p = .091$). There are a few interpretations for this result; the most obvious being the slower framerate on the nine-screen rendering (typically 1.2 fps vs. 6.7 fps during travel).

In addition, the physical size of the nine-screen display required users to make more mouse and head motion than when using a single-screen. In order to account for these differences, subsequent analysis was based on an 'adjusted time' for each group where the fastest possible completion time for a given trial was subtracted from each subject's recorded time for that trial. It should be noted that the effects described here were significant regardless of whether raw or adjusted time was used.

Time performance across tasks and displays carried significant main effects for both Layout technique and for Software FOV. Figure 6.8 shows that the Object space interface (mean = 127.7 sec.) took longer than the Viewport interface (mean = 101.4 sec.); $F_{1, 12} = 5.244$; $p = .041$. The low SFOV of 60 (mean = 131.2 sec.) also took longer than the 100 SFOV (mean = 97.9 sec.) with $F_{1, 12} = 11.805$; $p = .005$ (Figure 6.9). This follows our general hypothesis that the Viewport interface would be advantageous over the Object interface and that larger SFOVs would be advantageous over smaller SFOVs. This result is true of both Search and Comparison tasks.

There was also a significant interaction between the Layout and the SFOV ($F_{1, 12} = 19.094$; $p = .001$) variables. On low SFOVs of 60, the Object space technique took longer than Viewport, whereas on 100 SFOV the Object space was slightly faster than Viewport (Figure 6.10). For the tasks we tested, it is clear that a 60 SFOV is a poor performer and in addition, this was a particularly poor combination with Object space layouts, as the user is were forced to more navigation to get the annotation into the viewing frustum. The additional navigation requirements then increase the total time-to-completion.

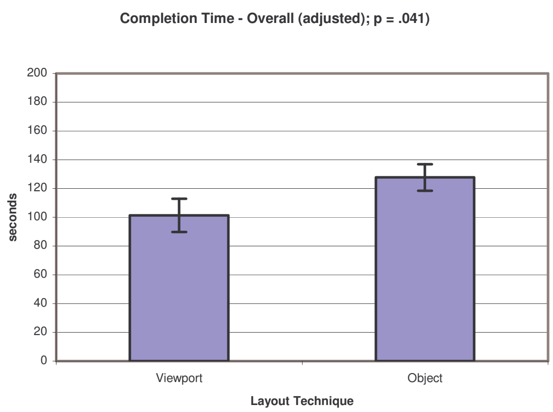


Figure 6.8: Main effect of Layout on Completion Time

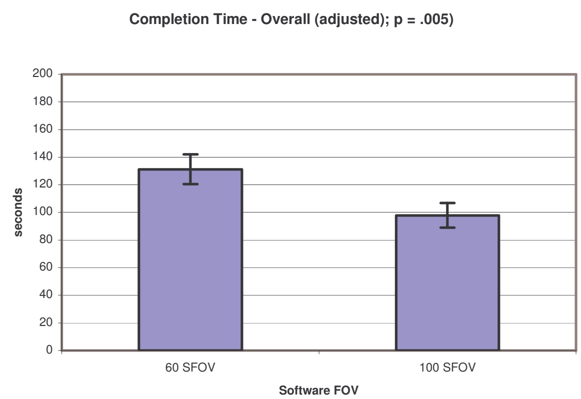


Figure 6.9: Main effect of SFOV on Completion Time

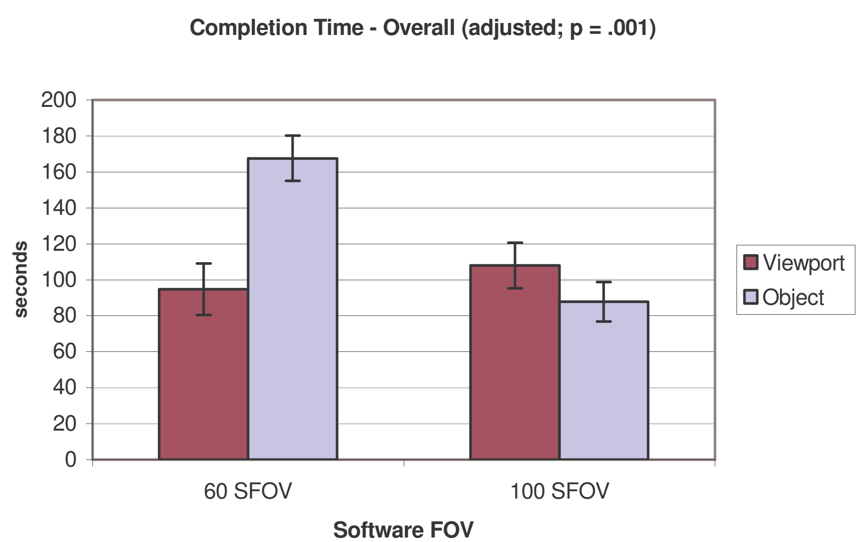


Figure 6.10: Interaction effect for SFOV and Layout technique on completion time.

Satisfaction and Difficulty

Results on these qualitative metrics are what we expect from knowing about the relative performance of interfaces and SFOVs by objective measures. The subjective results actually followed the pattern for Time performance. For example, subjects rated the Viewport interface more satisfying ($F_{1,\ 12}$ = 5.788; p

=.033) and the Object space layout most difficult ($F_{1, 12}$ = 35.396; p <.001). Subjects also rated the low SFOV as more difficult than the high SFOV and this difference was significant ($F_{1, 12}$ = 5.330; p = .040).

There was also an interaction between layout technique and SFOV for both qualitative metrics. While both interface types were rated similarly on the large SFOV conditions, in the small SFOV conditions subjects preferred the Viewport workspace ($F_{1, 12}$ = 8.007; p = .015) and it was perceived as less difficult ($F_{1, 12}$ = 17.684; p = .001). Figure 6.11 depicts this relationship.

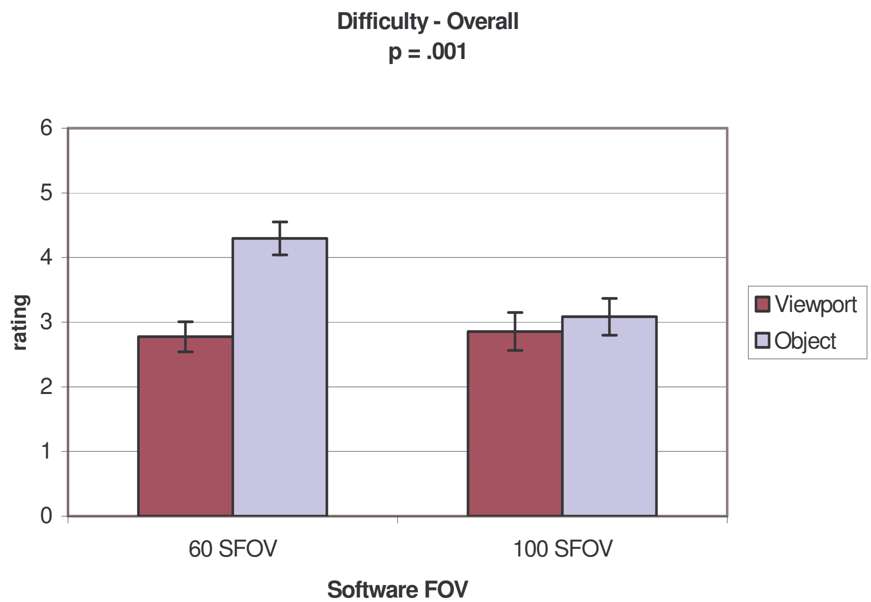


Figure 6.11: Interaction of Layout technique and SFOV on user difficulty rating

6.1.4 Conclusions

Interface designs for Information-Rich Virtual Environments such as those used in cell biology research and education can benefit from a better understanding of the role of depth and association cues in supporting search and comparison tasks. In such environments, objects may be distributed in all three dimensions and there may not be a horizon or gravity constraint on navigation. The challenge facing designers and developers is understanding the relationship of their information design choices (such as layout space) to the usability of their applications. For example, *"Where and how should enhancing abstract information be displayed relative to its spatial referent so that the respective information can be understood together and separately?".* The design problem is further compounded when considering the transfer of design layouts across rendering platforms.

In this study, we explored the relative performance of two IRVE layout spaces for search and comparison tasks in a desktop context. The first was an annotation layout scheme where the labels were co-located with their referent objects in the virtual scene in what we call Object Space. While this technique provides a tight spatial coupling (via depth cues & Gestalt proximity) between the annotation and its referent object, annotations may not be fully visible because of other occluding objects in the scene. To guarantee visibility regardless of position or orientation in the VE, we developed an IRVE layout component that manages annotations on a HUD just beyond the near-clipping plane (Viewport Space). This study investigated the information design tradeoff between the spatial coupling guarantee or the visibility guarantee provided by annotation labels in either Object or Viewport layout spaces. In addition, we asked if the relative advantages of a layout space hold when the scene is rendered on a large screen or under large projection distortion.

Object vs. Viewport Space

The first set of conclusions regards the usability of our IRVE layout techniques on a common single-screen setup. We asked: "Is one layout space with guaranteed visibility better than one with guaranteed tight spatial coupling for certain tasks?”. The results of this experiment showed that overall the Viewport interface outperformed Object space layouts on nearly all counts of accuracy, time, and ratings of satisfaction and difficulty across tasks. In other words, for the set of tasks performed, tight-spatial coupling of annotation to its referent (Object Space) was not as advantageous or preferable as the consistent visibility provided by an image plane layout (Viewport Space).

This result suggests that the development and evaluation of a richer set of Viewport Space layout capabilities (such as the X3D Compositing Component) would be worthwhile. If the tight spatial coupling provided by Object Space layouts is deemed necessary, consider further refining Object Space designs including managed or emergent layout schemes.

Single and Nine-screen Configurations

One of the main drawbacks to using our interfaces on the nine-screen display was the slower frame-rate. The VRML browser we used in the study did not work with the operating system to manage the hardware rendering with multiple video cards and displays. When the browser was enlarged to 1.5 x 1.5 screens or greater, the application switched to a software rendering mode which seemed significantly slower. However, the differences in time to completion across display configurations (due mainly to rendering speed) were not statistically related to task performance. We also found no statistically significant effect of display configuration on user accuracy.

The second research question we posed was: “Do the advantages of visibility or tight spatial coupling hold if the screen size is increased?”. Display size interacted with both Layout and SFOV variables for accuracy. The worst performing combination was the Object Space with a high SFOV on a small display. The best performing combination was the Viewport Space with high SFOV on a small display. However, on the large display, high SFOV the Object Space outperformed the Viewport Space.

With the tight spatial coupling, Object Space annotation schemes render the annotation with the rest of the scene. Annotations end up on the image plane nearby their referents- they provide the additional depth cues of occlusion and motion parallax and the additional Gestalt association cues of proximity with their referents. We can postulate that the advantage of the tight spatial coupling of Object Space only comes into effect when there is enough screen size (DFOV) to avoid the occlusion problem. Also, on the large screen size, tight spatial coupling means that users do not need to perform large saccades or head movements to see and read the annotation.

In examining the transfer of the Viewport BorderLayout interface design across display configurations, we can that say the successful transfer of an interface to a larger display is not simply a matter of scaling. On the large display, our Viewport Space design had the capacity for three times as many annotations. However on the large display, ergonomics require special consideration. The BorderLayout Viewport Space annotations began in the N container, which was above the line of sight at the top edge of the nine-screen display. This made frequent reference fatiguing for users. There is substantial work to be done in exploring Viewport Space annotation designs, especially for large displays. This work suggests that design and management choices for image-plane-interface layouts may be different depending on the size of the display.

Software Field of View

The third research question this study addresses is: “Do the advantages of visibility or tight spatial coupling hold if the SFOV is increased?”. Preliminary results indicated that for the cell environment, users had a high tolerance for large SFOVs, but that the tolerance was much less on the large display. In the study overall, users significantly rated low SFOV conditions more difficult; the differences in satisfaction ratings between SFOVs was not significant. Because we cannot compare subjective metrics between subject groups, the relationship between DFOV and SFOV remains an open research question.

Our study results showed that overall our two SFOVs levels did not significantly affect accuracy performance. However, higher SFOVs were advantageous for time especially on search tasks, but negatively impacted accuracy especially on comparison tasks. This result supports our hypotheses about the benefits of a high SFOV for search tasks (by showing more of the scene in the periphery) and liability of a high SFOV for comparison tasks (by distorting a scene object’s spatial location). It suggests that designers may consider modifying the SFOV dynamically depending on the user task.

Summary

Reflecting on the implications of these results, we can answer our original hypotheses and substantiate the following IRVE information design claims:

- Overall, the guaranteed visibility of Viewport Space offered significant performance and satisfaction advantages over the tight spatial coupling of Object Space annotation layouts. The effect was especially pronounced in the single-screen monitor configuration.
- The advantages of our Viewport Space layout did not transfer cleanly or scale iso-morphically up to the larger nine-screen configuration. On the large display condition for example, tight spatial coupling (Object Space) was more effective for accuracy across tasks but especially for comparison.
- Higher software FOVs decreased search time because they render more of the scene in the projection. Higher software FOV increased spatial comparison times because of fish-eye distortion.

The results of this evaluation contribute to our understanding of a fundamental layout space tradeoff in IRVEs. In addition, they provide initial guidance as to the challenges of designing integrated information spaces that are portable across display sizes and distortions. Still, the relationship between interface usability, Software Field Of View and Display Field Of View is an open research question; for example, what are the thresholds of size or projection distortion where various techniques break down and others become advantageous?

This experiment has shown an overall value for annotation visibility (Viewport); however on the large-screen condition, the proximity provided by Object space became more important. Designs and capabilities for both Object and Viewport layouts must be improved. Chapter 6 describes our efforts in this regard. For example, to be successful, portable IRVEs will require better text rendering facilities, layering and compositing functionality as well as support for pixel-agnostic layout mechanisms for the image plane.

6.2 Experiment 2: Object Space vs. Display Space

We conducted two evaluations using Display Space Techniques. The first, Snap2Diverse, was a survey into IRVE usability issues in immersive contexts. The second, Snap2Xj3D was a full study comparing an Object Space technique and a Display Space technique.

6.2.1 Snap2Diverse: Issues in Display Space

This experiment was a class project for the graduate class in Information Visualization, which was run by Polys, Ray, and Moldenhauer in 2003. The work was subsequently published in [Polys, 2004a]. In this project, we wanted to explore the use of a virtual environment as a view-component of a multiple-view visualization. We were especially interested in understanding context switches between coordinated visualizations inside an immersive 3D world such as the CAVE.

We developed, demonstrated, and tested a system where users can visualize and interact with multiple information types by way of a 'Hanging Picture' window. This hanging picture is an interactive 2D window superimposed (opaquely) on the immersive virtual world – hung on one wall of the CAVE (Figure 6.12).

The specific aspects to evaluate were:

- Viability of a visualization involving simultaneous, coordinated InfoVis and VE displays
- Ability to recognize visual and interactive relationships between the views.
- User preference based on nature of data, i.e. whether users choose appropriate visualizations for different types of data.
- Effectiveness of the 3D visualization of inherently spatial data as the central basis of the IRVE.
- Use of our novel 'XWand' interaction system for interaction with linked visualizations in the IRVE.

Figure 6.12: Relating perceptual and abstract information about molecular structures in a CAVE with multiple views (Snap2Diverse)

Evaluation

Subjects for the usability evaluation were from a variety of backgrounds, including chemistry, computer science, materials science, as well as virtual reality experts from within our lab. The format of the usability study was task and response, using think-aloud protocol. Users' subjective feedback for each task was noted. We also noted down their actions such as where they search for particular information (in 2D or 3D), the problems or discomforts they face with the interaction techniques, etc.

The subjects were given a benchmark set of tasks to be performed. Eight trials were formulated in four categories: *exploration tasks, search tasks, pattern recognition tasks, and comparison tasks*. Exploration tasks involved loading and describing features of various chemical components. The search tasks involved getting the number of atoms or bonds in a molecule or finding a specific attribute of an atom, bond or molecule. The pattern recognition and comparison tasks asked users to detect and describe similarities and differences between two molecules such as their size, molecular weight, and shape.

Qualitative Results

The results of the usability evaluation were obtained from user observation and the feedback questionnaire. They consist of usability issues, technical deficiencies, and suggestions by subjects. Usability aspects included: the time to understand the basic concept of coordinated 2D and 3D visualizations, the learning time for system interaction, and the ease by which users could learn and perform with the interface in the CAVE.

The most important results involved the users' interaction between the spatial and abstract information. Some tasks were designed so that users needed to answer questions about spatial properties based on some abstract information criteria or vice versa. In addition, there were tasks that could be answered by either perceptual or abstract sources. In most cases, users chose suitable visualizations to recover the information required for the finding and comparing tasks. This suggests that users were capable of interacting with and comprehending complex relationships between both the abstract and spatial information via the multiple-views IRVE design. If the task was an exploration task or pattern recognition task and could be accomplished by referring to either the perceptual or abstract information, nearly all users resorted to indexing via the spatial information. This confirms our hypothesis that the spatial information display of the IRVE would serve as a familiar grounding for users of the information space.

Learning time for the brushing interaction was surprisingly low for both VR novices and experts. The use of the wand was initially confusing for novice subjects, but after a few minutes they could fluently use the wand for: navigating by flying (thumb joystick), toggling between the navigation and selection mode (button 1), and selecting abstract information (thumb joystick and button 2). The visually implicit association between perceptual and abstract information (coordinated highlighting via selection) was established between the linked views and sufficient for completion of all the tasks by all users.

This result is important for IRVEs design as it suggests that users can operate multiple coordinated views to accomplish crucial IRVE tasks. We believe it is essential that IRVEs can integrate VE softwares and InfoVis software for the qualities of parallelism of display and coordination of views. This strategy can give users easier, integrated access to information and help them to generate insight through stronger mental associations between spatial and abstract information while preserving mental models of each type of information. The implementation demonstrates that sophisticated interfaces and system behavior can be accomplished through the sharing of data identifiers and simple event mechanisms.

Results Summary

The Snap2Diverse system has shown that there are three crucial issues in the design of IRVEs. First, we need to design better data models that integrate perceptual and abstract information. Second, the transformations that map that information to VEs and InfoVis applications must be carefully considered. Third, we need a flexible but structured way to support the display and interaction coordinations between the VE and InfoVis applications. Snap's event-based coordination mechanism has proven a useful system to integrate abstract and spatial visualization applications.

Our current multiple views implementation makes use of the wand and the Hanging Picture to interact with perceptual and abstract information in the same environment. In terms of the IRVE design space, Snap2Diverse can be characterized as: a display-fixed location, low-density layout with visually implicit association of aggregated abstract information of multiple types. Another interesting design option to explore are the tradeoffs involved in putting the Snap display and interaction window on a hand-held tablet surface. This last possibility raises the question of how to better integrate external applications into IRVEs. The trend toward event-based, Model-View-Controller interfaces could bear this out. For example, the principles that make the Hanging Picture work could be extended to the generalized notion of an 'application texture'. In this approach, the windowing system would manage the rendering and pointer events for an application mapped to some geometry in the world. The X3D Specification Working Group is currently developing a 'Compositing Component', which may address this important functionality.

For our system to be more successful, more virtual environment interaction techniques should be implemented and evaluated. For example, in our prototype version, if the user became lost in the 3D world, there was no way to reset the viewpoint except to reload the application, which was not a good option. Flexible interface widgets and windows (such as Qt components via VEWL [Larimer, 2003]) could be added to manage and expand more system control functionality. Additional 3D interaction techniques for navigation and selection should also be explored including picking touch, laser pointer, and spotlight techniques [32,33]. Future work with Snap could be to relax its data model in order to better deal with hierarchical sources, as well as possibly specifying the component of origin in the SnapEvent class.

This work demonstrates a heterogeneous system for embedding interactive, user-defined 2D visualizations inside 3D virtual environments and coordinating them across a network. The architecture we describe is flexible for linking Snap with DIVERSE and could be applied to any number of data domains. In terms of our chemical visualization and analysis application, Chemical Markup Language and XSLT give us great flexibility in the loading of chemical data into the visualization pipeline from different formats. We also believe this approach could be applied to embed and coordinate any XML data in such Information-Rich Virtual Environments. This work contributes to the field by implementing and assessing an Information-Rich Virtual Environment design for multiple, coordinated views of perceptual and abstract data.

6.2.2 Multiple Views Experiment

This experiment was a class project for the graduate class in Information Visualization, which was run by Shupp, Volpe, Glina, and Polys in 2004 and documented as a VT CS Tech Report [Polys, Shupp et al., 2006]. The purpose of this experiment was to test two extremes of the IRVE information design space and their support for Search and Comparison Tasks (Figure 6.13). These extremes represent either end of the IRVE association – occlusion tradeoff. This tradeoff results from the integrated nature of IRVEs: abstract information and spatial information are interrelated and information visualizations are registered to objects in the virtual environment. When annotations are 'tightly-coupled' to virtual objects though depth cues or 2D gestalt cues, they introduce occlusion to the view – they block objects in the virtual environment and each other. When annotations are given their own screen space, the information types are 'loosely-coupled' - it may not be clear what attributes or properties are related to specific objects.

Object Space

On one end of this tradeoff is Object Space, where abstract information is embedded within the virtual environment and in the same coordinate system as its spatial object (its referent). Object space provides a tightly coupled visualization where abstract and spatial information are strongly associated through Depth and Gestalt cues. A summary of the properties in this Object space condition is:

- A single view visualization where all detail information is distributed in the virtual environment.
- Strong visual cues (Gestalt and Depth) for associating information with its referent - high association, high occlusion.

Display Space

On the other end of this tradeoff is Display Space where the virtual environment is one of multiple visualizations presented or linked together. A summary of the properties in this Display space condition is:

- A multiple view visualization where detail information is aggregated to sibling level information visualizations.
- No visual or interactive association cues between abstract information and spatial referent (no brushing and linking), only identification of data by name (Gestalt Similarity). This condition is extremely low occlusion, low association.

The question we examined was:

"Under what task conditions is the single-view (with tight spatial coupling) of Object Space advantageous over multiple (loosely coupled) views of Display Space?"

or

"Under what task conditions is the cost of a context-switch less than the benefit of no context switch?"

The main hypothesis was that the high association of Object Space would be advantageous for tasks where the criteria were spatial, but that the loosely coupled views would be advantageous for tasks where the task criteria were abstract. We used a cell model and populated it with Semantic Objects generated from a CML dataset.

Environment

The virtual world was an animal cell with nine regions: the membrane, cytoplasm, nucleus, nucleolus, three mitochondria, and two lysosomes. All regions except the cytoplasm and membrane had text labels. Thirteen molecules were placed throughout the cell, with no more than three molecules per region. Five abstract details were displayed for each molecule: chemical formula, molecular weight, boiling point, melting point, and density.

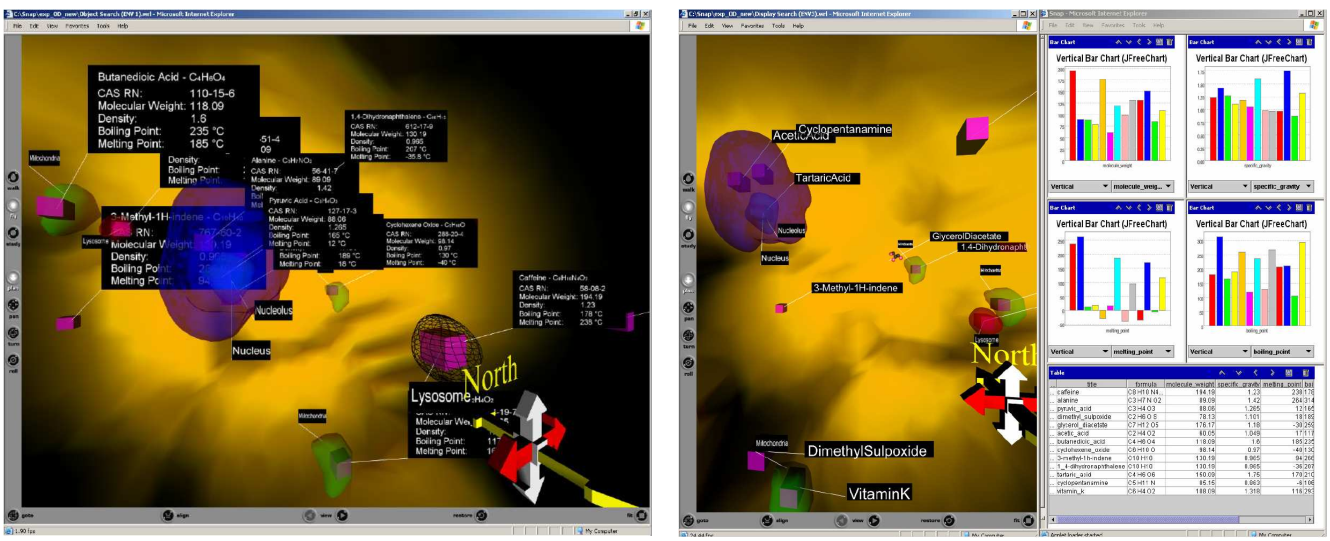


Figure 6.13 Object Space vs. Display Space

Four Eucaryotic cell environments were used, two for Object Space and two for Display Space. The two environments for a technique were grouped based on task type. For example, the first environment was used for all search questions within the Object Space technique, the second was used for all comparison tasks within the Object Space technique, and so on.

The experiment was carefully designed to ensure users could not memorize molecular properties between tasks. First, no task provided or asked for abstract information that was used in any other task. This prevented participants from memorizing abstract information between tasks. Second, no task provided or asked for a spatial property regarding the physical nature of a molecule in any other task (i.e. the molecule's shape). Third, multiple environments were chosen to ensure participants could not memorize the location of a molecule between tasks (i.e. region in which the molecule resides). In each environment, each molecule was placed in a region different than that of any other environment. Furthermore, the three mitochondria, two lysosomes, and nested nucleus - nucleolus pair were also given different locations within the animal cell between environments. These measures prevented participants from memorizing spatial information between tasks.

With the exception of molecule and region locations, all of the environments shared the same design, and there are several noteworthy design decisions. One could toggle the visibility of a molecule text label by clicking on the corresponding molecule. This feature was designed to compensate for occlusion. Furthermore, the molecule text labels were not static in size. During navigation, the labels would dynamically resize for readability. In IRVE design component terms, the Object space condition was FixedRotation with Periodic Scaling. Although every molecule's size relative to the environment was significantly larger than the real world, it was still impossible to see molecules that were not within a close range. Therefore, pink cubes were used as markers for molecules when viewed at a distance (e.g. Figure 6.14).

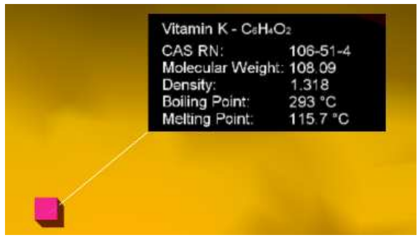


Figure 6.14: The Vitamin K molecule at a distance

Cubes change to the molecular structure when the user is close enough to the molecule. Similarly, the surfaces of the three mitochondria, two lysosomes, and nested nucleus - nucleolus pair all change from opaque to a transparent wire frame so that the molecules or their landmarks can be clearly seen upon approach. The wire frame switch was essential to let users select objects inside other objects in order to minimize their annotation (e.g. Figure 6.15). All of the aforementioned design decisions were explained in participant training.

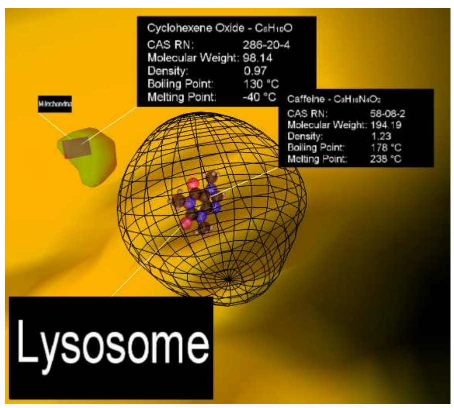


Figure 6.15: Close-up of Caffeine in a wire framed lysosome and an opaque mitochondria containing a landmark for Cyclohexene Oxide in the background

The Cortona VRML Client was used as the Web3D viewer in Microsoft Internet Explorer® to run all virtual environments. For tasks performed in Object Space, the window was expanded to full-screen with no Internet Explorer toolbars visible, maximizing the pixels allocated for the display.

In the Display Space conditions, the CML database was loaded into Snap and five views were built and linked. There were four bar graphs depicting the common attributes of all the molecules (molecular weight, density, boiling point, and melting point). In addition, the molecules table was loaded to provide numeric detail for the attributes.

For tasks performed in Display Space, two Explorer windows were used. The first window was allocated 710x1024 pixels on the left side of the screen to display the virtual environment. The second window was allocated 570x1024 pixels on the right side of the screen. The Snap-Together Visualization system was used to display the abstract information in the second window. Again, no Internet Explorer toolbars were visible in either window.

It should be noted that in the system tested, the information visualizations in display space were not interactively linked to the VE. Therefore there were no consistent depth cues and no Gestalt cues linking annotation and referent. The only association between the views was the molecule's name. The relationship of perceptual cues in the conditions tested is shown in Table 6.4. Again, High Association and High Occlusion reside in the top left corner. Low Association and Low Occlusion reside in the lower left corner.

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	O	O	O		
Motion Parallax	O	O	O		
Relative/Size / Perspective					
None				D	

Table 6.4: Depth and Gestalt Cues presented by Object (O) and Display (D) Space layouts used in Experiment 2

Participants

Sixteen people participated in the experiment. Participants consisted of 6 females and 10 males between the ages of 21 and 31. Nine participants were majoring in Computer Science; one in Computer Engineering; one in Industrial Systems Engineering; three in Human Nutrition, Foods, and Exercise; and one in Biochemistry and Biology. Three participants were undergraduate students, twelve were graduate students, and one was faculty.

Most participants were very familiar with computers, with the exception of a few who were at least somewhat familiar, and all participants used computers at least several times a week if not daily. Eight participants had experience using virtual environments at least once before (e.g. 3D games, the CAVE). All participants completed all tasks, which were counterbalanced and shuffled to eliminate the possibility of memory for previous locations or attributes (Table 6.5).

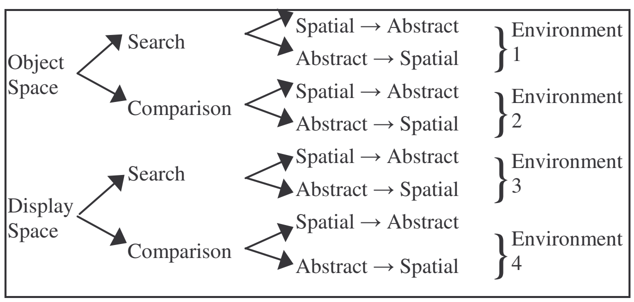


Table 6.5: Task Structure in the Object vs. Display Experiment

Materials and Procedure

This experiment was performed on a Dell Dimension 8200 desktop system with an 18" LCD monitor using 1280x1024 resolution and standard two-button mouse.

The experiment was completed in two sessions. During the first session, participants filled out a preliminary questionnaire, which collected demographic information, and were trained to use the Web3D viewer. Participants trained with two worlds; the first world was the Monterey Kelp Forest aquarium [Brutzman, 2002] and the second was an animal cell (as used in the experiment) using the Object Space technique. They practiced navigating in these worlds until they were comfortable. While training in the animal cell environment, participants were also reminded of the animal cell structure.

Furthermore, noteworthy features and dynamics of the virtual environment were explained. This includes the ability to toggle the molecule text labels, the cubes used as landmarks for molecules, how regions transform from opaque to a transparent wire frame upon approach, and how to recognize both the cytoplasm and the membrane regions. We informed participants that there would be different cell environments between conditions. This session lasted up to 30 minutes.

The second session was the formal experiment. Before starting the second session, participants were asked to read each task and ask us for any clarification before beginning. They were also asked to perform each task as fast and accurately as possible. During each task, evaluators recorded quantitative data, such as the participant's time-to-completion and whether the answer was correct. After completing each task, participants were asked to fill out a questionnaire of qualitative measures. Ratings collected by these questions were later used to determine the perceived difficulty and satisfaction of completing tasks. Participants took breaks between environments as desired. This session lasted about one hour. Five dependant variables were measured during the evaluation: time, accuracy, satisfaction, task difficulty, and 3D navigation difficulty. Experimental materials and result tables are included in Appendix C.

Detailed Results

We constructed a General Linear Model ANOVA for each of the dependent variables. Results are organized below for each measure. Paired Samples t-tests were used to find significant contrasts when interaction effects were found.

Accuracy

The ANOVA for accuracy shows that the main effect for display technique was not significant (p=0.699). However, the three-way interaction between display technique, task type, and task mapping was significant ($F_{3, 13} = 5.662$; p=.010). Our results show that neither display technique was significantly more accurate than the other for Search tasks of either information mapping. The significant differences occur in the Comparison tasks. For the A->S information mapping, the Display Space technique was advantageous over the Object Space technique (85.4% vs. 70.8%). This difference was significant by $t_{15} = -2.150$; p = .048. For the S->A information mapping, the Object Space technique was advantageous over the Display Space technique (64.6% vs. 43.7%). This difference was significant by $t_{15} = 2.825$; p = .013. Figure 6.16 depicts this interaction.

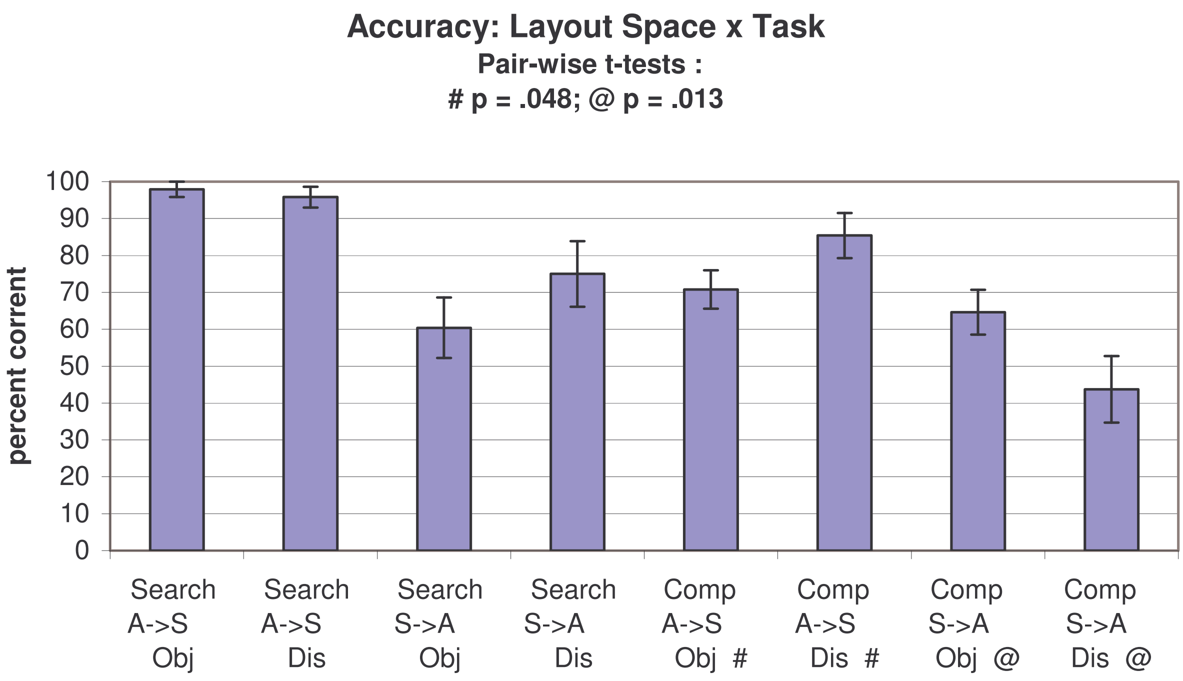


Figure 6.16: Average accuracy of the eight conditions.

Time

The ANOVA for task time showed that the main effect for display technique was not significant (p=0.1522). On average, users completed tasks in Display Space (71.7s) faster than in Object Space (81.2s). However, using the total time-to-completion is misleading. Since some tasks require more navigation than others we decided to compare what we call 'Adjusted Time'. This allowed us to accurately compare the display techniques.

The task time for each run was converted to the adjusted time by excluding the ideal time:

Adjusted Time = Task Time – Ideal Time.

The ideal time for a task is defined as the time it took to complete unavoidable navigations. This ideal time was calculated by taking the fastest time that it took an expert user (one who knows the answer) to complete only the navigations required for the task. These navigations are not limited to the VE portion of the interface; rather, they are all the navigations necessary for completing the task. Since this evaluation seeks to understand which display technique users can utilize more efficiently, we only need to examine the time it takes a user to explore the interface beyond any required navigations.

The ANOVA for adjusted time overall is only modestly influenced by the display technique (p=0.153). On average, Display Space (67.0s) performed better than Object Space (77.62s). Similarly, the two-way interaction using average adjusted time for display technique and information mapping was only modestly significant (p=0.112). However, this interaction reveals that abstract to spatial (A->S) tasks were faster in Display Space (mean 45.9s) than Object Space (mean 69.0s). This difference is significant $t_{15} = 2.729$; $p = .016$. It does not appear that spatial to abstract questions are faster for either display technique. Figure 6.17 shows this relationship.

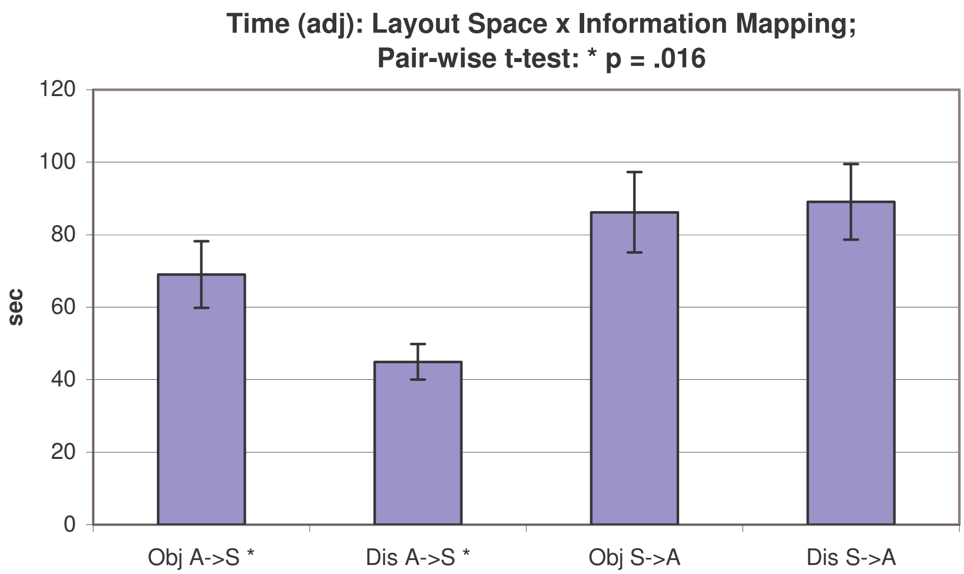


Figure 6.17: Average adjusted time for display technique and task mapping

Satisfaction Ratings

The ANOVA for participant satisfaction shows that the display technique main effect was significant ($F_{1, 15} = 14.596$; p=0.002), Figure 6.18. Display Space (5.0) was on average rated more satisfying than Object Space (4.6). The satisfaction rating is based on a perceived level of satisfaction on a Likert scale of 1 to 7, where 1 was least satisfying and 7 most satisfying.

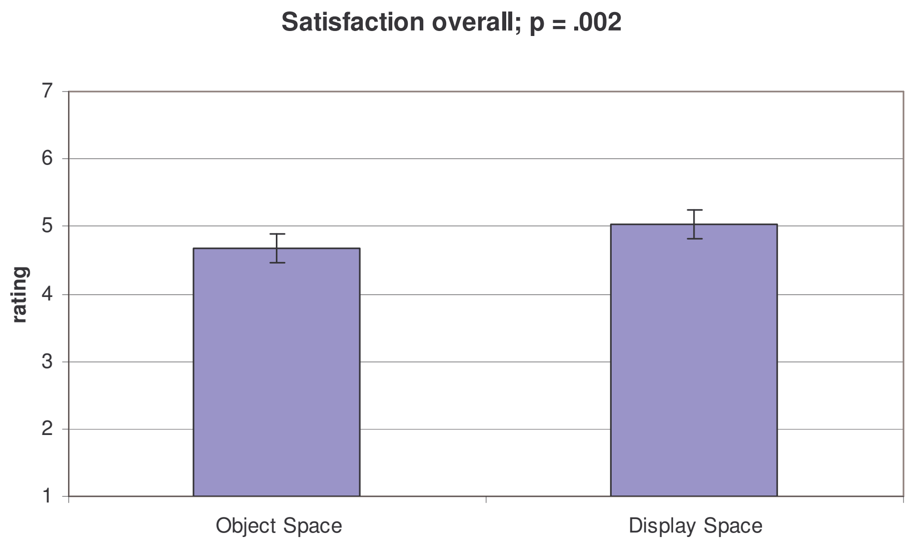


Figure 6.18: Average satisfaction rating for display techniques.

There was also a significant interaction between display technique and task mapping for participant satisfaction ($F_{1, 15}$ = 5.971; p=0.027). Pairwise t-tests reveal that the best condition overall is Display Space for abstract to spatial tasks (A->S); Display Space (A->S) was more satisfying than in Object Space (5.5 vs. 4.7); t_{15}= 3.525, p = 003 (Figure 6.19). It does not appear that display technique influences user satisfaction for spatial to abstract (S->A) questions.

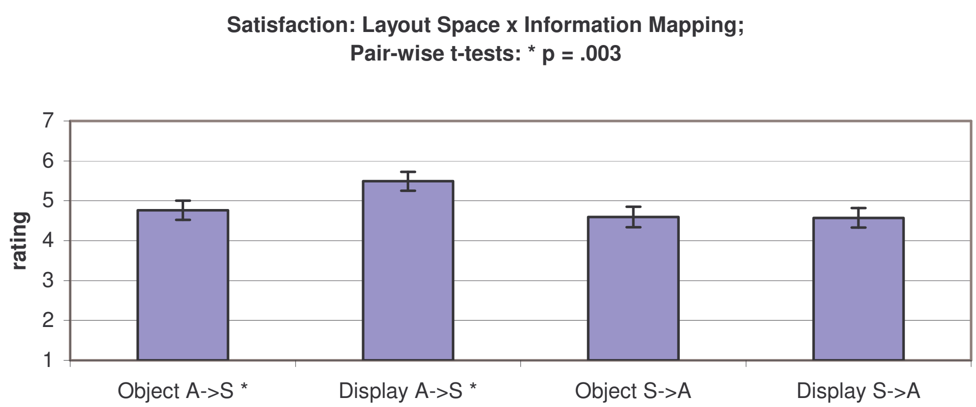


Figure 6.19: Average satisfaction rating for display technique and task mapping.

Difficulty Ratings

Users were asked to rate how difficult the interface was for completing each task. This is classified as task difficulty. For the task difficulty rating, we used a Likert scale of 1 to 7, where 1 was least difficult and 7 most difficult. Users were also asked to rate how difficult it was to navigate in the 3D environment for each task. This is classified as 3D navigation difficulty. The 3D navigation difficulty rating was also measured using a Likert scale of 1 to 7, where 1 was least difficult and 7 most difficult. The ANOVA reported similar results for task difficulty and 3D navigation difficulty.

The ANOVA for display technique shows that Layout Space was not significant for either task difficulty (p=0.181) or 3D navigation difficulty (p=0.387). However, there were some significant interactions. The two-way interaction between display technique and task type was significant for task difficulty ($F_{1, 15}$ = 5.545; p = .033) and almost significant for 3D navigation difficulty ($F_{1, 15}$ = 3.996; p= .064). This is shown in Figure 6.20. Pair-wise t-tests show that for the Abstract to Spatial information mapping (A->S), the Display Space condition is considered significantly less difficult than the Object Space condition (t_{15} = 3.525; p = .003). The relationship is mirrored for ratings of 3D Navigation difficulty with t_{15} = -3.148 and p = .007.

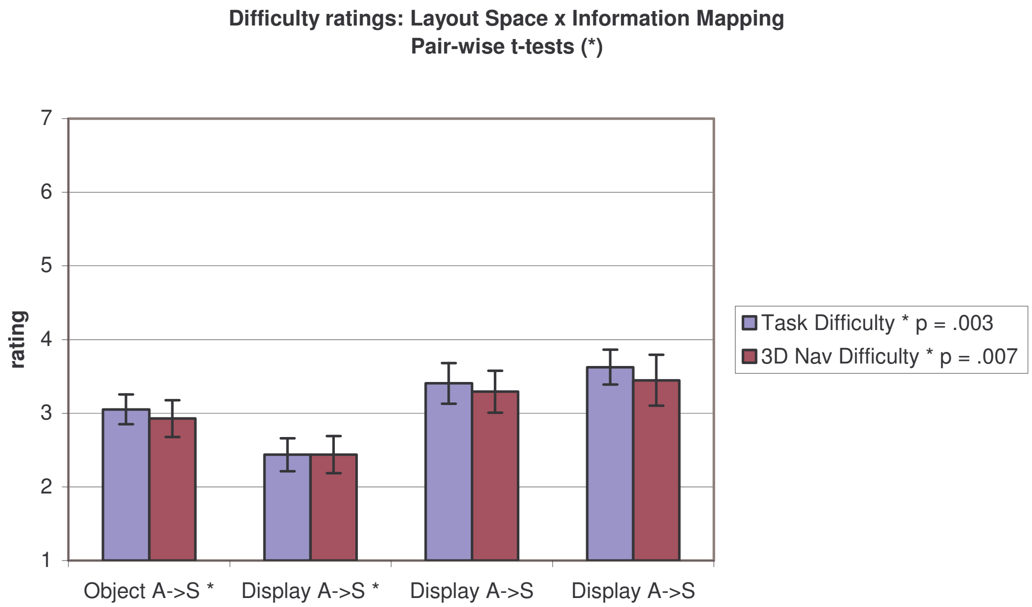


Figure 6.20: Average difficulty ratings for display technique and task information mapping

Some effects were not significant statistically, but warrant mention. On average, users rated search tasks more difficult in Object Space (3.0) than Display Space (2.4). Users also on average rated 3D navigation more difficult for search tasks in Object Space (3.1) than in Display Space (2.6). It does not appear that comparison questions influence the task difficulty or 3D navigation difficulty for either display technique.

Results Summary

The following is a summary of conditions that have statistically significant effects (p < .05) using ANOVA:

Fastest Adjusted Time

- A → S in Display Space

Most Accuracy

- Search and A→ S in either technique
- Search and S → A in Display Space
- Compare and S → A in Object Space
- Compare and A→ S in Display Space

Most Satisfaction

- Display Space
- A → S in Display Space

Least Difficult for Task and 3D Navigation

- Search in Display Space
- A → S in Display Space

For many conditions, our quantitative results (time and accuracy) show users performed better in Display Space. Users were most likely faster in Display Space because they were not limited to 3D interactions for examining abstract information. Additionally, we observed users would often get distracted in Object Space if the task took them longer than usual. This is most likely why users were less accurate in Object Space. On the other hand, Object Space is clearly better for one condition. For comparison tasks that are mapped spatial to abstract, Object Space is more accurate. This is most likely because all the pixels on the screen are available for 3D interaction, which is best suited for examining spatial attributes. Since it is in this condition where users must first examine multiple spatial attributes, it is understandable that Object Space would perform better for this condition.

Our qualitative results (satisfaction and difficulty) show that for many conditions users preferred Display Space over Object Space. We observed that there is a correlation between user satisfaction and time. For both the original task time and adjusted time, tasks were completed faster in Display Space. Therefore, it appears that users are more satisfied with Display Space simply because they can complete the task in less time. It is also clear that users associated task difficulty with the navigation difficulty of the 3D view. Since users were not limited to navigating within the 3D environment for examining abstract information, users most likely favored interaction with the 2D views over the 3D navigation. Therefore, Display Space is most likely rated less difficult because users were given an alternative to the tedious 3D navigations when examining abstract information.

Overall, it appears that users prefer using the Display Space technique and perform better in Display Space. However, Object Space is particularly better suited for answering spatial to abstract comparison (S->A) questions. It is interesting to note that for all dependent variables (adjusted time, accuracy, etc.), the main effects for task type and task mapping were significant (p < .05). Our results showed what we might expect; for example, search tasks are easier than comparison tasks. Also, abstract to spatial (A->S) tasks was easier than spatial to abstract (S->A) tasks. Furthermore, no matter what technique is used in our visualization, the easiest situation is a search task of the A->S information mapping.

6.2.3 Conclusions

We can draw some useful conclusions from these experiments involving Display space.

In the **Snap2Diverse** project, we surveyed usability issues with coordinated, multiple-view visualizations on a large screen immersive display. Overall we found the federated multiple-views scheme to be extremely powerful in terms of our fundamental IRVE activities. The ability to customize information representations and coordinate them with brushing and linking was demonstrated to be learnable and useful for a set of common tasks. However, we noted some serious problems beyond the technical setup that warrant attention for future designs.

For example, the Xwand interface that allowed users to navigate and select objects in the VE *and* select objects in the Hanging Picture was moded and did not have equivalent sensitivity between the modes. This meant that first, the proper button must be pressed, and then the pointer responsiveness to the wand joystick actions were not equivalent between the VE and the InfoVis. So although the pointer could be found in its last position, the switch was challenging because the speed of pointer response was different between the VE and the Hanging Picture. In general, moded interfaces require that mode be visible; this puts the information in the world rather than requiring the user to keep the information in their head. Wand buttons are novel for most users and usually require some learning time.

Another real problem was rendering text in the Hanging Picture while the CAVE was running in a stereoscopic mode. Text had to be enlarged almost 2x to be legible and the lines of the san-serif font were still blurry. Clearly better rendering techniques or resolutions are required for large amounts of text in a CAVE. In addition, because active stereo shutter glasses reduce light, contrast between text and background should be increased.

Finally, while the brushing and linking was helpful in driving user attention to the proper data points, we observed issues with representation including information overload. The Visualization schema we tested with provided more information than was necessary to complete the tasks. Because users were not experts on the data set and had not set up the Snap Visualization schema, this occasionally resulted in confusion. While users will need to be somewhat familiar with their data sets, this observation supports the application of a Task Knowledge structure analysis to IRVE designs. Through such an analysis, designers can be economical and effective about what information is displayed and how it is displayed.

In the **Snap2Xj3D** experiment, where we tested an Object space technique against a Display space technique, there are three results to highlight. First, we can say that the benefits of additional attribute-centric visualizations with reduced occlusion were stronger than the costs imposed by context switching between two visualizations of low association. To put this another way, the benefits of the tightly-coupled association in Object Space were not sufficient to overcome the occlusion problem. In contrast, the Display Space technique showed that the benefits of multiple views with no occlusion were sufficient to overcome the costs of low association (context switching).

Second, contrary to the hypothesis, Display Space was advantageous for a task where the criteria was spatial (Search: S->A). This points to a problem with the Object Space design as tested- the occlusion problem was managed naively (by default, all labels were visible). Finally, even with high occlusion and no attribute-centric visualization, the Object Space technique was better for Comparison: S->A accuracy. This leads us to acknowledge that this layout technique is important to improve at least for this task type (comparisons based on spatial criteria).

These two Display space evaluations show that the low association of linked IRVE visualizations may not be a problematic usability issue if the information visualizations provide appropriate alternative representations. They also support the value of the VE component in a multiple-view visualization - an IRVE. For example, in the CAVE users considered the VE the primary visualization; when given a choice, they typically indexed through the VE. In the desktop situation we showed that the embedded Object space technique was better than the Display space for one task-information mapping (spatial comparisons).

These results also demonstrate the advantages and feasibility of Display space techniques and open the way for further improvement of designs and supporting information architectures. In addition, at least one task-information mapping (spatial comparisons) demonstrated the utility of (embedded) Object space layouts and showed that while occlusion may hinder visibility and legibility, it is also the strongest depth cue.

7. Tradeoffs in Layout Spaces

In the previous chapter, we described a set of evaluations we conducted to identify the issues and tradeoffs between IRVE display spaces across desktop and immersive platforms. These evaluations have given us a general understanding of the usability benefits and liabilities of information design techniques across IRVE display spaces. They have allowed us to assess which layout spaces provided advantages for certain tasks and information mappings, and in what display contexts those advantages hold.

After identifying the tasks and display circumstances for which particular layout spaces have advantages, we now turn our attention to the many possible techniques within a layout space. Our first series of experiments showed that Object space was advantageous for certain task-information mappings and also overall on large displays; Viewport space was advantageous for certain task-information mappings and also overall on desktop displays. These results raise the question of 'What makes an effective Object space technique?' and 'What makes an effective Viewport space technique?'. To answer these questions, we conducted the following evaluations.

The first evaluation examined Object Space layouts on a large screen (front wall of the VT CAVE) and focused on the two principal tradeoffs in IRVE design: the Occlusion / Association tradeoff and the Relative Size / Legibility tradeoff. To test the Occlusion / Association tradeoff, we asked if an IRVE layout algorithm could maintain the benefits of proximity between annotation and referent while reducing occlusion in the scene, thereby making more annotations visible at once. To test the Relative Size / Legibility tradeoff, we asked if annotation scaling methods have an impact on performance. Here, the layout concern is how to strike an effective balance between providing annotation Legibility versus providing the consistent Depth cue of Relative Size between annotation and referent. Therefore, this first evaluation measured the strength of various Depth cues in Object Space. Videos depicting the techniques tested are listed in Appendix H.

The second experiment was conducted on a regular desktop system and focused on the role of the Gestalt cues in Viewport Space layouts. Specifically, we asked if a Viewport Space layout algorithm could aid user performance by providing various levels of Proximity and Connectedness cues between annotations and their referents. Both experiments described below used the same data and task set; we conclude the chapter with a post-hoc comparison of the layout techniques. Videos depicting the techniques tested are listed in Appendix H.

7.1 Experiment 3: Object Space

Our research has shown there are situations where tight coupling of annotation and referent support advantageous user performance. By rendering an annotation in the scene and in the same coordinate system as the referent object, the abstract information is highly conformal to the object in the virtual space. This is a result of the fact that Object Space annotations provide the consistent depth cues of occlusion, relative size, motion parallax, and the additional Gestalt association cue of image-plane proximity with their referents.

In the evaluations described in the last chapter we found that in large screen display situations, this tight spatial coupling between annotation and referent can be advantageous because it does not require users to perform head movements or large saccades when judging the relations between abstract or spatial attributes of objects. We also saw that Object Space can be advantageous for Comparison tasks where the criteria is spatial- the additional cues can help users discriminate spatial relations and locations.

There are some serious drawbacks to Object Space layouts however. For example, there is the limited visibility of annotations. This problem arises for two reasons – first, if an object is not in the viewing frustum, it's annotation will not be visible; second, when annotations are drawn in Object Space, they are subject to occlusion problems from other nearby objects or annotations. In addition there is the limited legibility of annotations – if the annotation is drawn with a consistent Relative Size cue as its referent object, it may not be legible from far away. This requires users to navigate through the virtual environments and get closer to the object in order that the annotation is large enough to read.

In order to better understand these design tradeoffs of Object Space and improve Object Space display techniques, we posed the following question and designed an experiment to test it:

Tight coupling between Annotation and Referent (e.g. Object Space)
is advantageous in some cases, especially on large screen displays;
What techniques are effective to increase the visibility and legibility of annotations?

This experiment examined a set of Object Space configurations to determine their relative effectiveness for Search and Comparison tasks. First we wanted to address the occlusion problem in Object Space layouts, so we looked at how an annotation management approach could reduce occlusion while maintaining Gestalt Proximity. Second, we wanted to address the legibility problem and see if changing annotation scaling factors (providing the depth cue of Relative Size) can affect task performance on a large display (e.g. Figure 7.1).

Figure 7.1: Experimental setup for the Object Space Experiment

Specifically, we asked, 'Can we design an economic layout technique that can reduce occlusion over the range of environments?' and 'How does the annotation's scaling (Legibility vs. the depth cue of Relative Size) impact the IRVE usability?'.

- *Hypothesis 1:* In Object Space, emergent or constraint-based layout techniques are advantageous over deterministic techniques for a range of spatial configurations and task mappings. The emergent layouts should reduce occlusion in the scene by having annotations avoiding each other and other objects in the scene.
- *Hypothesis 2:* In Object Space, we expect that Continuous Scaling will be advantageous because it provides consistent legibility. This would mean that Legibility is more important than the depth cue of Relative Size.

7.1.1 Information Design

In this section we describe the specific Object Space layout techniques examined in this evaluation: the Force-Directed and the ScreenBounds layouts (Figure 7.2). Both of these layouts were tested in conjunction with scaling properties of the annotation (described below). In both of these layout techniques, a screen-aligned bounding region is defined for the Semantic Object. Annotations are then positioned at some location just along this bounds.

The two techniques differ in how the location on the bounds is determined. The manipulation of this variable is intended to test the assumption that less occlusion among information in the scene is advantageous. Consider that in crowded IRVE scenes, occlusion between information types can hinder visual access and thus hinder performance. If we can devise a layout algorithm that seeks to minimize occlusion, more information may be visible and apprehended at once. Table 7.1 shows the combinations of cues presented by the display techniques in this experiment.

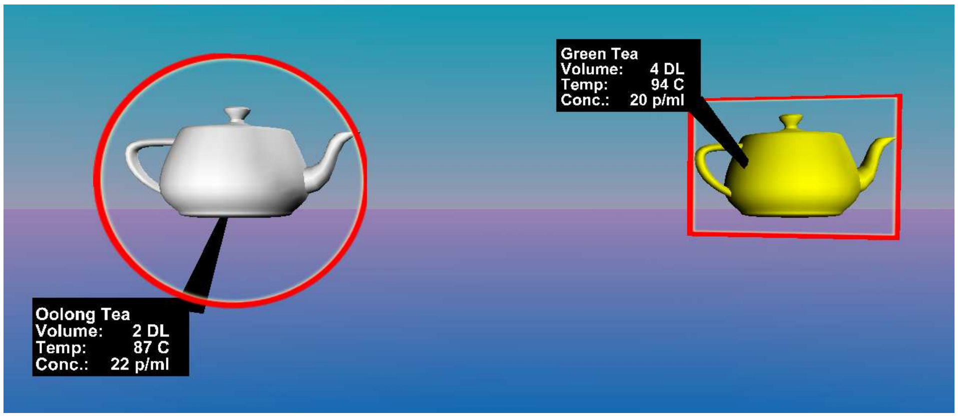


Figure 7.2: Force-directed (left) and ScreenBounds (right) Object Space layouts

	Proximity	**Connectedness**	**Common Fate**	**Similarity**	***None***
Occlusion	**SB**	**SB**	**SB**		
Motion Parallax	**SB F**	**SB F**	**SB F**		
Relative/Size / Perspective	***SB F***	***SB F***	***SB F***		
None					

Table 7.1: Range of Depth and Gestalt Cues presented by Object ScreenBounds (SB) and ForceDirected (F) Space layouts used in Experiment 3; italics denotes the secondary independent variable

Annotation Layouts

Layout: ScreenBounds

The 'ScreenBounds' technique is intended to provide a set of layout locations that maintain association between the annotation and its referent. The rationale behind having a set of locations (as opposed to one location like the FixedPosition or one relative location as in RelativePosition) is that multiple locations can provide options for layout in crowded situations. In this version of the ScreenBounds technique, four points are defined as the screen-aligned bounding box. Typically these are positioned at the far extents of the rendered object, but may be placed anywhere. The annotation will snap to a different point and the label will shift appropriately depending on the viewing angle of the user (Figure 7.2, right).

Layout: Force-Directed

The force-directed algorithm as applied to IRVEs is intended to reduce need for the scene designer to explicitly manage the location of annotations in Object Space for occlusion. This is done with an emergent, constraint-based layout. Our Force-Directed algorithm is a simple ruleset distributed inside each Semantic Object that takes into account the location of obstacles in the scene. The algorithm attempts to minimize occlusion in the scene by creating a repulsion force between itself and other annotations and objects. The algorithm projects these obstacles' force to a screen-aligned bounding circle, and moves the annotation along that circle in order to minimize the force (Figure 7.2, left). The force-directed layout maintains the Gestalt association cue of Proximity, but not the same discrete spatial configuration as the Screen Bounds technique.

Annotation Scaling

This evaluation also tested a secondary independent variable of annotation scaling. The manipulation of this variable was based on the Relative Size / Legibility tradeoff. This tradeoff can be summarized as follows: if the annotation is rendered with a consistent depth cue of Relative Size, it will appear smaller when further away; at small sizes annotations may not be legible.

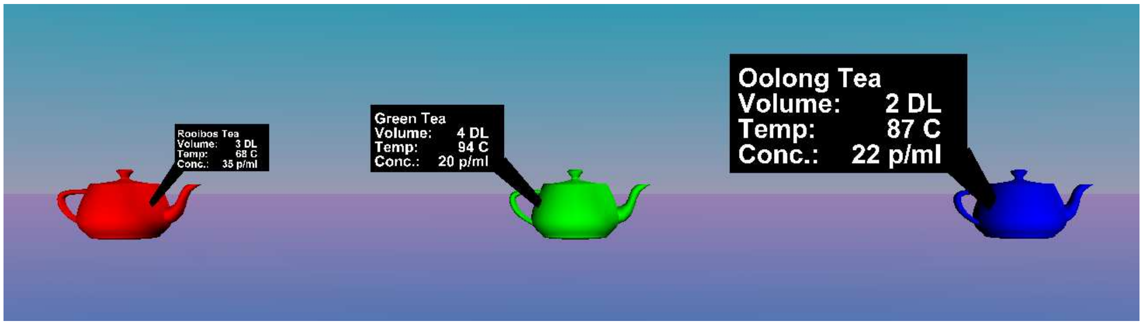


Figure 7.3: Relative Size vs. Legibility; from right to left- No scaling, Periodic scaling, and Continuous scaling

We generated three designs to address this tradeoff (Figure 7.3). The first, **No scaling**, maintains the depth cue of Relative Size between the annotation and referent by fixing the size of the annotation. The second, **Continuous scaling**, effects the annotations scale by a constant multiple of the distance. This multiple guarantees a constant size (and legibility) regardless of user distance. This approach provides no Relative Size depth cue.

The third approach, **Periodic Scaling**, is a compromise design. Periodic scaling defines a distance interval that is used as a conditional to scale the annotation up or down. This has the advantage of providing legibility across a range of distances. While the periodicity can be used a judge of distance traveled during navigation, it can also confound the Relative Size depth cue in situations where two objects are located within the distance interval of each other.

7.1.2 Method

Experimental Conditions

We designed a full-factorial within-subjects experiment to test our layout algorithms and annotation scaling techniques (Table 6.2). Fourteen subjects from the undergraduate pool were tested and 13 completed the experiment: 12 females and 1 male. The subjects sat on a high stool and used a wireless mouse and keyboard, which was set on a flat plexiglass podium at chest height. The environments were back-projected mono-scopically on the front wall of the CAVE (10'x10', 1280x1024 pixels). Cortona VRML client was used with OpenGL in full-screen mode at 32 bit color. The Software Field-Of-View (SFOV) of the IRVE renderings equaled the Display Field-Of-View (DFOV) from the user's position (90 degrees).

Technique (Occlusion vs Association)	**Scaling (Relative Size vs Legibility)**
Screen Bounds	No Scaling (consistent)
Force Directed	Periodic (confounded)
	Continuous (none)

Table 7.2. Experimental design for the Object Space occlusion experiment: 2 x 3 = 6 within-subjects conditions.

A set of cell environments were generated and populated with CML molecules. Each cell environment contained 13 molecules and 8 possible structure locations (nucleus, nucleolus, mitochondria (3), lysosome (2), cytosol). The locations and attributes of molecules was shuffled for all trials in order to guarantee naïve search and comparison. All annotations were visible by default and labels could be toggled on or off by selection. All conditions included Gestalt Connectedness with a polygonal connector (rather than just a line) and a HUD gyroscope was included to provide bearing information. Figure 7.4 shows a CML cell example illustrating each display technique evaluated in this experiment.

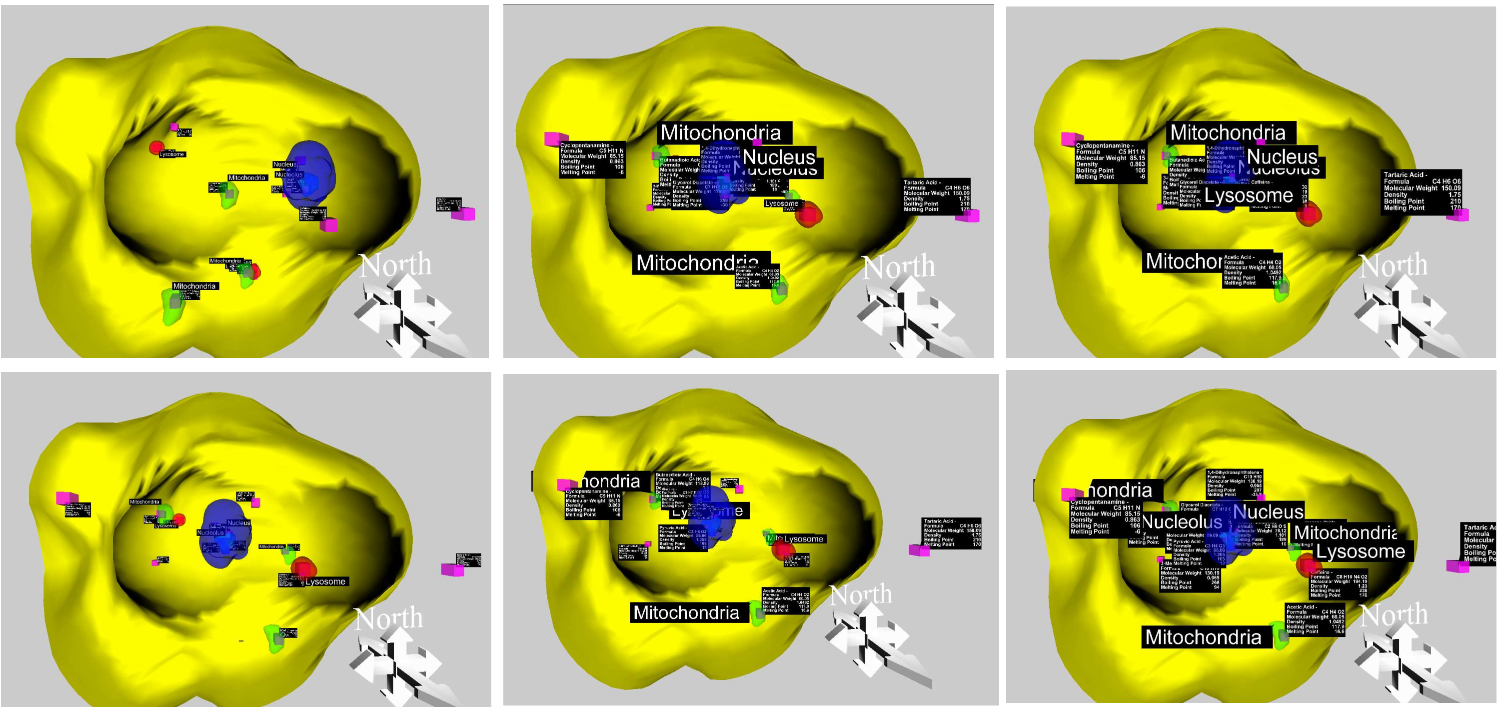


Figure 7.4: Layout: (ScreenBounds = top row; ForceDirected = bottom row) by Scaling (from left to right: None, Periodic, and Continuous)

Protocol

As in our experimental comparisons between Layout Spaces, subjects were introduced to the VRML navigation controls in the Kelp Forest. They learned the use of the mouse-drag for VRML FLYing and the combination with the 'ALT' key to pan or slide. Once user affirmed they were comfortable navigating the virtual environment, a random example cell environment was loaded and the selection and LOD functions of Semantic Objects were demonstrated.

Subjects were given a 5"x7" printed schematic of cell landmarks with a printed depiction of the difficulty and satisfaction scales. Each Display technique was presented four times with a different task (task type x task-information mapping) for a total of 24 trials. Trials were administered in a random order and we measured accuracy, time to completion (seconds), navigation distance, difficulty (1-6), and satisfaction (1-6). In addition, Cognitive battery tests for Hidden Patterns and Number Comparison were administered to each participant before the trials. The tasks, task-information mappings, and example questions are shown in Table 7.3. Experimental materials and result tables are included in Appendices D and E.

Task	Notation	Example
Search for Spatial information based on Abstract information	Search: A->S	Where in the cell is the molecule with a molecular weight of xy.wz?
Search for Abstract information based on Spatial information	Search: S->A	What molecule is right next to the nucleus?
Compare Abstract Information and derive Spatial information	Compare: A-> S	Where in the cell is the molecule with the lowest boiling point?
Compare Spatial information and derive Abstract information	Compare: S->A	What molecule is furthest North?

Table 7.3: Task-information types used in the Object Space experiment

7.1.3 Detailed Results

We constructed a General Linear model of the experimental data and for each dependent variable and performed an ANOVA among the conditions. Where significant interactions were found, t-tests were performed to parse out which combinations caused significant effects. These results illuminate important design issues with Object Space layouts on large displays. Many were unexpected or the result of complex interactions between scaling and layout. Some were the result of individual differences or personal background such as experience with computer games. The results are detailed in this section and summarized in the next section.

Accuracy

There were no significant main effects for Layout across tasks. There were however, significant effects for Annotation Scaling across tasks ($F_{2, 11}$= 6.754; p = .012). Figure 7.5 depicts this relation. Pairwise t-tests show that the significant difference is between No Scaling (90.3%) and Periodic Scaling (80.7%) by t_{12}= 3.825; p = .013. Continuous scaling was not significantly different than either of the other two techniques. The depth cue of Relative size is consistent with the referent in the No Scaling condition, but is confounded in the Periodic Scaling condition.

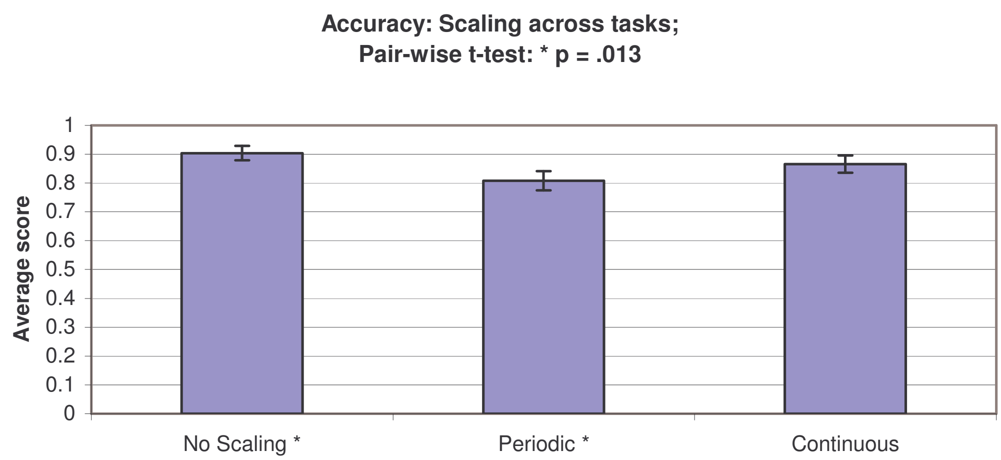


Figure 7.5:Effect of annotation scaling on accuracy

For Comparison tasks, The Screen Bounds technique was significantly more accurate ($F_{1, 12}$= 5.855; p = .032). This is shown in Figure 7.6 where the average score for ScreenBounds was 83.3% and ForceDirected was 71.7%. This effect is primarily due to comparisons of abstract information ($F_{1, 12}$= 5.333; p = .04). This result suggests that ScreenBounds layout technique is better when users have to compare abstract information in labels.

One explanation for this result is that annotations are more stable in the ScreenBounds technique since they undergo discrete changes (from corner to corner) rather that moving all the time along a circle. We suggest this as a problem with the ForceDirected technique: labels that move all the time so they are harder to read *and* they may not be in the same relative location as the last time they were seen. This is a downside of the ForceDirected layout: labels are continuously moving to adjust their layout. In addition, this dynamic nature mean that users might not find the label in the last location they saw it.

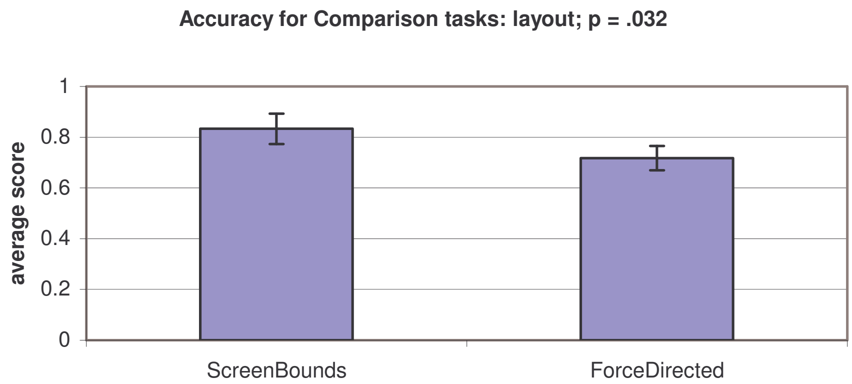


Figure 7.6: Effect of Layout Technique on comparison task accuracy

For tasks of the A->S information mapping, ScreenBounds was more accurate than ForceDirected layout (96.2% vs. 89.7% correct) by $F_{1,\ 12}$= 7.50; p = .018. For tasks of the S->A mapping, Scaling caused a significant effect ($F_{2,\ 11}$= 5.577; p = .021). Pairwise comparisons reveal that No Scaling was better than both Periodic (p = .006) and Continuous (p = .027), which were comparable (88.5%, 71.2%, 76.9% respectively).

Time

We analyzed raw time to completion for each task. Over all tasks, there was no main effect for Layout. There was, however, a main effect for annotation Scaling ($F_{1,\ 12}$= 4.318; p = .040) where No Scaling (98.2 seconds) took significantly longer than the other two techniques (Periodic = 83.3 seconds and Continuous = 83.4 seconds). Figure 7.7 depicts this relation. Pairwise, No Scaling took longer than Periodic Scaling (t_{12} = 3.083; p = .009) and No Scaling took longer than Continuous Scaling (t_{12}= 2.151; p = .053). Similar to accuracy, overall time performance with Periodic and Continuous Scaling was comparable.

The liability of No Scaling for completion time is primarily due to its shortcomings for Search tasks ($F_{2,\ 11}$= 4,284; p =.042) where again No Scaling (95.5 seconds) is significantly worse than Periodic Scaling (69.7 seconds) by t_{12}= 3.00 ; p = .011. This discrepancy in time-to-completion is what we might expect because in the No Scaling condition, users have to navigate in the virtual environment in order to make distant annotations legible. Combined with the results for accuracy reported above, we can see this is a classic speed / accuracy tradeoff: users were more accurate with No Scaling but No Scaling took longer.

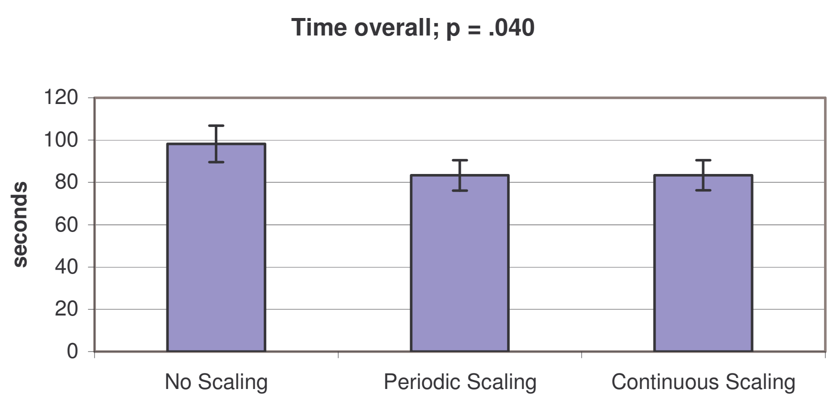


Figure 7.7: Scaling effects on completion time overall

There was a main effect for layout on time performance depending on the task type. For Search tasks, the ForceDirected layout was faster (69.3 sec.) than the ScreenBounds layout (93.1 sec.). This was significant by $F_{1,12}$= 13.089; p = .004. The effect was reversed for Comparison tasks however. For

Comparison tasks, the ScreenBounds layout was significantly faster than ForceDirected layout (87.9 vs. 102.9 sec) at $F_{1,12}$= 5.107; p = .043. These effects are shown in Figures 7.8 and 7.9 respectively. These results suggest that our ForceDirected algorithm did improve visibility of information; however, the lack of occlusion and the dynamic positioning of annotations hindered made comparisons.

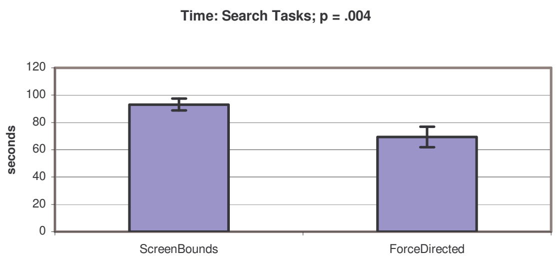


Figure 7.8: Layout effects on Search task time

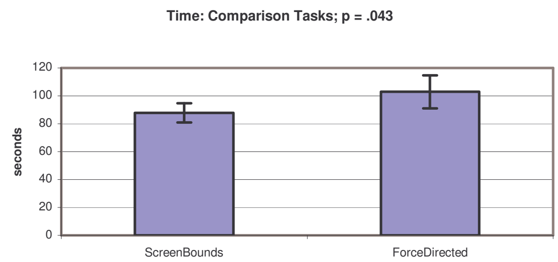


Figure 7.9: Layout effects on Comparison task time

When tasks were of the abstract to spatial (A->S) information mapping, the ForceDirected layout was faster than the ScreenBounds (93.3 vs. 113.9 seconds) and $F_{1,\ 12}$= 8.778; p = .012. From the interaction between layout and scaling for this information mapping, we can see that in both the No Scaling and Continuous Scaling conditions, the ForceDirected layout was significantly faster ($F_{2,\ 11}$)= 5.789; p = .019. This interaction is shown in Figure 7.10. There were no significant differences between layout and scaling conditions when tasks were of the spatial to abstract (S->A) information mapping.

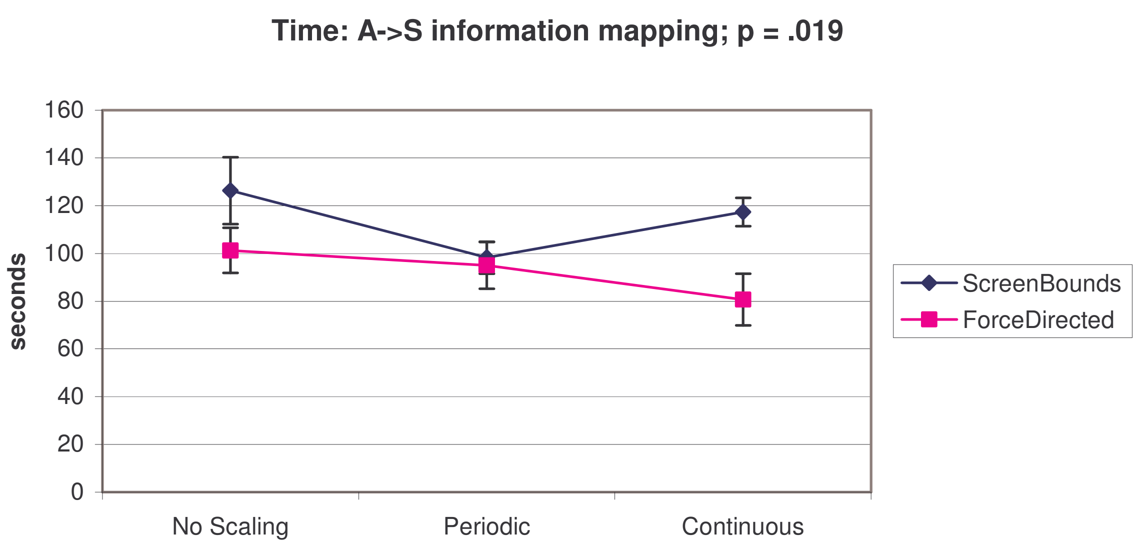


Figure 7.10: Interaction of Layout and Scaling for A->S information mapping

Finally, when we consider the demographic data collected from the subjects, we note that there were no differences overall for gender or use of glasses. There was a significant difference between those who had played computer games (121.4 seconds) before and those that hadn't (78.3 seconds). The strong effect, apparent in Univariate ANOVA and t-test analyses (t_{11} = -3.728; p = .003), is counter-intuitive, but may be explained by the following:

- the games they were experienced with were not 1st person 3D games or

- their experience with WALKing navigation or joysticks interfered with their ability to transfer to FLYing navigation with mouse or
- their experience with game play biased them for a conservative criterion across tasks… for example making sure that they did not miss a piece of information along the way

Navigation Distance

For each frame, the testbed components collected position data of the user's camera. When the user completed the trial (by clicking on the HUD compass), this information was dumped to a text file. This data was analyzed by a Perl script to sum the distance each user traveled in their completion of each task.

Over all tasks, we see significant results for Layout (F(1,12) = 4.972 ; p = .046). This effect (where ForceDirected layout results in more user navigation) is due to performance on Comparison tasks Layout (ScreenBounds = 946.8 world units vs. ForceDirected = 1529.3 world units) where $F_{1,\ 12} = 11.591$; p = .005 and also due to performance on S->A information mappings (ScreenBounds = 791.3 world units vs. ForceDirected = 1414.6 world units) where $F_{2,\ 11}= 6.138$; p = .027. In both of these cases, more user navigation is required to disambiguate spatial criteria or make comparisons when the depth cue of occlusion is not present.

In terms of annotation Scaling, there were also significant effects for distance navigated. These might be expected from our initial observation that No Scaling requires more spatial navigation in order to gain legibility of an annotation. The effect was true overall $F_{2,\ 11}= 4.179$; p = .045 and for A->S information mappings ($F_{2,\ 11}= 4.668$; p = .034). Pairwise t-tests reveal that in both these cases, Periodic Scaling and Continuous Scaling are comparable and No Scaling requires significantly more spatial navigation than the other two conditions. Overall, No Scaling was 27.2% slower and for A->S it was 29.8% slower).

Difficulty

Over all tasks, there were no significant effects of layout or scaling. However, there were effects for both task type and task information mapping. As in the objective measure of time-to-completion, there was a significant difference between layouts depending on the task type: for Search tasks, the ForceDirected layout was rated less difficult than the ScreenBounds (2.90 vs. 3.27) ($F_{1,\ 12} = 6.050$; p = .030) and Comparison tasks ForceDirected layout was rated more difficult than the ScreenBounds (3.85 vs. 3.50) at ($F_{1,\ 12} = 7.259$; p = .020). These effects, which are shown in Figures 7.11 and 7.12 are primarily due to the fact that Layout effected user ratings of difficulty depending on the information mapping. When tasks were of the abstract to spatial (A->S) information mapping, ForceDirected was rated less difficult than ScreenBounds (3.12 vs. 3.72) at $F_{1,12} = 11.919$; p = .005. When the mapping was spatial -> abstract (S->A), Force-directed was more difficult than ScreenBounds (3.62 vs. 3.05) at $F_{1,12} = 14.063$; p = .003.

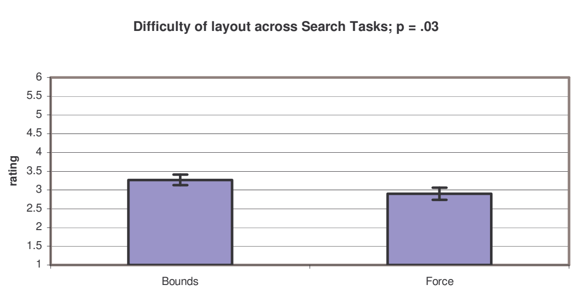


Figure 7.11: Effect of Layout on user difficulty for search tasks

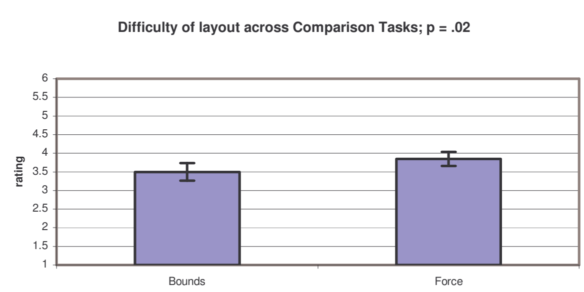


Figure 7.12: Effect of Layout on user difficulty for comparison tasks

Annotation scaling also showed significant effects on user difficulty ratings depending on the information mapping. When tasks were of the abstract to spatial (A->S) information mapping, $F_{2, 11}= 4.221$; $p = .044$; The significant difference is between No Scaling and Periodic Scaling (3.58 vs. 3.17) by $t_{12} = 2.66$; $p = .021$. When the mapping was spatial -> abstract (S->A), a nearly-significant effect $F_{2, 11}= 3.773$; $p = .057$ shows that there was a significant difference between ratings for Periodic Scaling and Continuous Scaling (3.58 vs. 3.02) by $t_{12}= 2.87$; $p = .014$. These effects are shown in Figures 7.13 and 7.14 illustrate these effects for Scaling x information mapping.

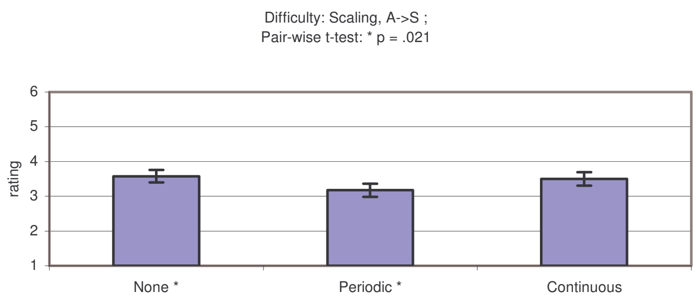


Figure 7.13: Effect of Scaling on user difficulty for tasks of A->S mapping

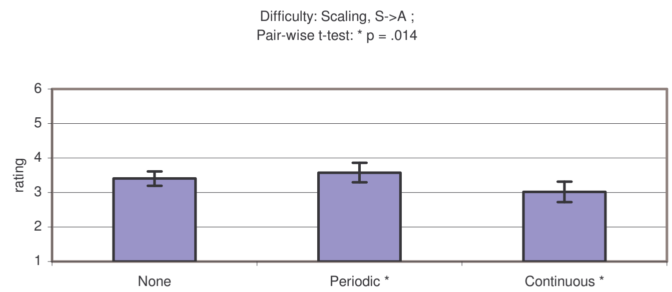


Figure 7.14: Effect of Scaling on user difficulty for tasks of S->A mapping

Satisfaction

There were no main effects across all tasks for either Layout or Scaling. However, over all tasks, the interaction of Layout and Scaling was almost significant ($F_{2, 11}= 3.992$; $p = .052$). While Periodic scaling was nearly equivalent under both layout techniques, the Force-directed layout was preferred for Continuous scaling and Screen Bounds layout was preferred for no scaling. This trend is mostly due to significant interactions by task and task mapping. For example, the same pattern is significant for Search tasks, shown in Figure 7.15 ($F_{2, 11}= 4.686$; $p = 0.034$).

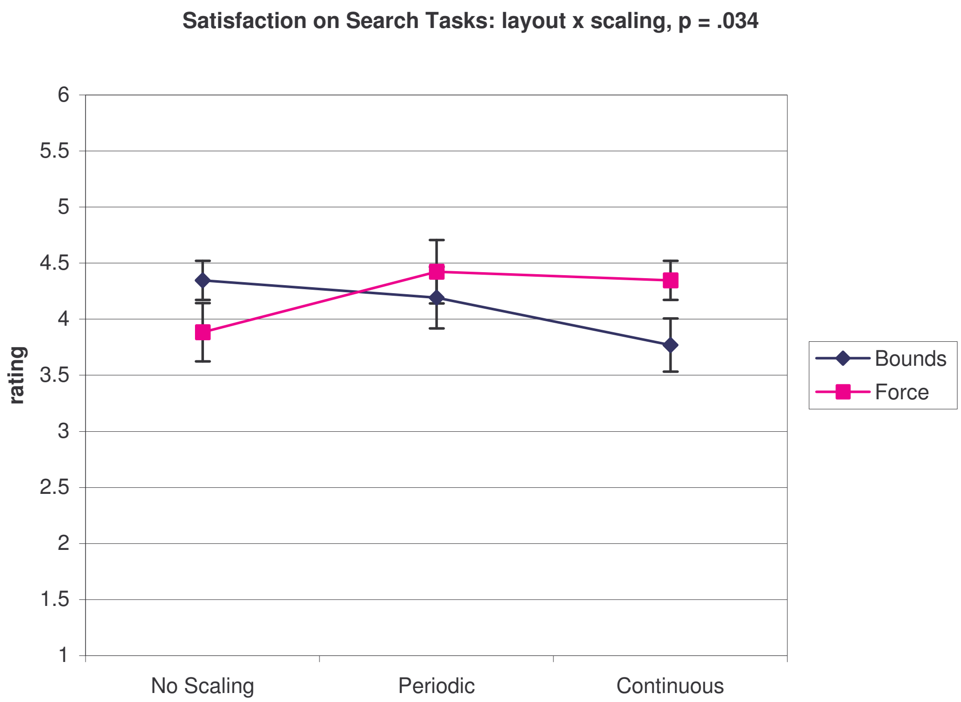


Figure 7.15: Interaction of Layout and Scaling for user satisfaction on Search tasks

For information mapping, there was a barely-significant difference between Layouts for information mapping: for A->S task mappings, ScreenBounds was more satisfactory than ForceDirected (4.30 vs. 3.97) at $F_{1,\ 12}$= 4.771; p = .050. For the S->A information mapping, there was a significant interaction between Layout and Scaling ($F_{2,\ 11}$= 9.362; p = .004). Pairwise t-tests reveal that the effect of No Scaling depends on Layout (ScreenBounds = 4.5 vs. ForceDirected = 3.69) at t_{12}= 6.062; p < .001. In addition, No Scaling is significantly better than ForceDirected + Periodic Scaling (3.92) at t_{12}= 2.961; p = .012.

Cognitive battery scores

We computed Pearson correlations between the user test scores and the objective and subjective performance metrics on our Object Space IRVE design combinations. Most correlations were not significant: there were none for time or for satisfaction ratings. For Accuracy, there was a significant correlation between user's Numbers Comparison test performance and two specific interface and task combinations *(both R = .561, p = .046)*:

- Search tasks with the ForceDirected layout and Continuous Scaling
- Comparison tasks with the ScreenBounds layout and No Scaling

The correlation to accuracy performance in the first case could be related to the user's perceptual aptitude for quick recognition of numeric stimuli. In the second case, the annotations were small, which made it less ambiguous as to which object the annotation referred, but required more time to navigate to the object to make the annotation large enough to be legible. Here, the ability to remember a set of digits would be beneficial for Comparison tasks.

For difficulty, there was a significant correlation with user's Hidden Patterns scores and rating of:

- Screen Bounds layout with Continuous scaling (R = .7, p = .008)
- Search tasks with the Screen Bounds layout (R = .566, p = .044)
- Search tasks with Screen Bounds layout with Continuous scaling (R = .654, p = .015)

In this case, a higher score on the Hidden Patterns test was (positively) correlated to higher difficulty ratings on these conditions. We leave the explanation for future work.

7.1.4 Results Summary

There are a number of rich results from this experiment. Table 7.4 summarizes the significant effects of Layout and Scaling by tasks and by information mapping.

Layout: ScreenBounds

The ScreenBounds technique performed very well in terms of accuracy, especially on Comparison tasks. The ScreenBounds technique provides a proximity relation that is one of 4 discrete positions relative to the referent object; the discrete positioning was helpful in that users could scan the scene and find a label in the same location of the visual field where it was last seen.

However, the ScreenBounds technique was problematic for Search tasks because of occlusion: nested objects or objects sharing a line of sight typically have occluding labels. For the case of molecules nested inside cell structures, users adopted the strategy of minimizing the labels when the occlusion was impeding to task performance.

Layout: ForceDirected

We designed the ForceDirected algorithm to reduce occlusion between objects in the scene. This did provide an advantage for Search tasks, where the ForceDirected technique received significantly more positive ratings. This can be attributed to the algorithm's success in making more labels visible at a given time.

However the poor performance in Comparison tasks point to two specific aspects of our technique that had problematic consequences. First, by reducing occlusion between labels and objects in the scene, we remove the strongest depth cue. The lack of this cue between labels makes spatial comparisons more challenging. Second, the ForceDirected layout compute and update label's position every timestep, resulting in a high amount of visual movement. In the ForceDirected layout, labels change their location in the visual field. When the user is attempting to read or compare two labels, they may have to re-find the label.

Annotation Scaling

Overall, No Scaling enabled better accuracy than Periodic Scaling. Continuous Scaling provided a middle ground and was comparable with both No and Periodic Scaling in terms of accuracy. The advantage of No Scaling is especially apparent on A->S information mapping accuracy. However, No Scaling was the worst performer in terms of time and distance navigated. This speed / accuracy tradeoff could be due to two reasons. The first possibility is that when the depth cue of Relative Size is maintained, there is less ambiguity as to which annotations belong to which referent. The second possibility results from the fact that No scaling also had the worst time and navigation distance scores; this raises the likelihood that users got a better understanding of the space by navigating it.

Periodic Scaling was rated less difficult than No Scaling on S->A information mapping tasks and like accuracy, Continuous Scaling provided a middle ground. For A->S information mapping tasks, Continuous Scaling was rated less difficult than Periodic Scaling and No Scaling was comparable to both. Taken together, these two results support the value of Legibility over Relative Size for subjective ratings.

Layout x Scaling Interactions

For completion time, there was a significant interaction for A->S information mappings. In the Continuous and No Scaling conditions, ForecDirected layout was faster than ScreenBounds layout. Also, for Subjective ratings there was an interesting interaction for satisfaction on Search tasks. When annotations were Continuously scaled, ForceDirected layout was rated more satisfying than ScreenBounds layout; when annotations were Not Scaled, ScreenBounds layout was rated more satisfying than ForceDirected layout.

Task	Accuracy	Time	Navigation Distance	Difficulty / Satisfaction
Overall	*No scaling better than Periodic scaling; Continuous scaling comparable to both*	*Worst: No scaling*	*ForceDirected worse than ScreenBounds* *Worst: No scaling*	
Search		*ForceDirected better than ScreenBounds* *Worst: No scaling*		*Force-directed less difficult than ScreenBounds* *ForceDirected more satisfying than ScreenBounds (w/ Continuous scaling)* *ScreenBounds more satisfying than ForceDirected (w/ No scaling)*
Comparison	*ScreenBounds better than Force-directed*	*ScreenBounds better than ForceDirected*	*ScreenBounds better than ForceDirected* *Worst: No scaling*	*ScreenBounds less difficult than Force-directed*
S->A	*Best: No scaling*		*ScreenBounds better than ForceDirected* *Worst: No scaling*	*No scaling more difficult than Periodic scaling*
A->S	*ScreenBounds layout significantly better than Force-directed*	*ForceDirected better than ScreenBounds for No and Continuous scaling*		*ScreenBounds more satisfactory than ForceDirected* *Periodic more difficult than Continuous*

Table 7.4: Summary of significant results in the Object Space experiment

7.1.5 Conclusions

The results of this evaluation lend us insight into important issues in IRVE Object Space design. One important point to note is the task and mapping specificity of advantages. This specificity points to crucial design tradeoffs in IRVEs and the fact that successful interfaces must mitigate.

Occlusion vs. Association

Our first hypothesis was: in Object Space, emergent or constraint-based layout techniques would be advantageous over deterministic techniques for a range of spatial configurations and task mappings. The emergent layouts should reduce occlusion in the scene by having annotations avoiding each other and other objects in the scene.

Our results indicate that reducing occlusion is positive for Search task time and subjective ratings, but can negatively impact Comparison performance (accuracy, time, navigation distance, and difficulty). This is a partial confirmation of our first hypothesis. The first reason for this is that occlusion is a strong depth cue that can aid spatial comparisons. The second is that in the ForceDirected condition as tested, labels were continuously moving to some degree; this was particularly detrimental in comparison tasks because the spatial configuration of annotation to object would change, requiring users to continuously re-find labels and re-associate them with their proper referent. Due to the nature of our stimuli, we are not able to tell if this problem on comparisons with ForceDirected layouts is due to: a) ambiguity caused by reducing occlusion thus making discrimination of spatial criteria more difficult or b) due to the lookup of abstract target information in its dynamic location.

While the cause of ForceDirected layout's poor performance on comparison tasks must be left to future work, we can say that less occusion (more visibility of abstract information) was advantageous for Search tasks. This result suggests that while visibility is crucial for some tasks, removing occlusion (a strong depth cue) can lead to ambiguity in comparisons. We can also make a point that in the design of emergent or constraint based layouts, designers should slow down dynamic layouts to provide fixation time on annotation text. This could be accomplished by only updating annotation position while the user is navigating for example.

Relative Size vs. Legibility

Our second hypothesis was: In Object Space, we expected that Continuous Scaling would be advantageous because it provides consistent legibility. This would mean that Legibility is more important than the depth cue of Relative Size.

In the scaling techniques we tested, we note that No Scaling maintains the cue of Relative Size, Continuous Scaling does not provide it, and Periodic Scaling confounds it. In the effect of No Scaling, we see a classic speed/accuracy tradeoff. Over all tasks, No scaling provided the best accuracy but worst time and navigation requirements. This was also true of S->A tasks. No Scaling was the worst scaling condition for Search time.

Over all tasks, the time performance of Continuous scaling was comparable with Periodic scaling. Thus our second hypothesis was not confirmed by objective performance measures. In some cases however, such as A->S information mappings, Continuous Scaling was rated significantly less difficult than Periodic Scaling. The second hypothesis also is somewhat supported by subjective measures – users liked the immediately legible labels when they needed to look up or access to abstract information (S->A mapping).

7.2 Experiment 4: Viewport Space

In our comparison of Object and Viewport Space (Section 5.1), we showed that Viewport Space layouts such as the HUD BorderLayoutManager were advantageous overall and especially on desktop displays or under high Software Field-Of-Views (SFOV). The guaranteed visibility and legibility of annotations in Viewport layouts is one of their strongest advantages. However, we do not know how annotations should be organized in a HUD and what role the Gestalt Association cues play in user performance.

We therefore devised an experiment to test the value of Gestalt cues in Viewport layouts for common desktop situations. We chose two dimensions to test: Proximity and Connectedness. We designed two levels of Proximity and three levels of Connectedness (described below). Based on the literature discussed in Chapter 2, we formulated the following hypotheses:

- *Hypothesis 1:* Proximity has been claimed to be one of the strongest association cues of Gestalt. The Proximity HUD will provide a net advantage.
- *Hypothesis2:* Connectedness has been claimed to be one of the strongest association cues. Semitransparent polygon connectors should provide a middle-ground advantage between association and occlusion.

7.2.1 Information Design

In order to test the power of Proximity and Connectedness cues in Viewport Space layouts, we designed two versions of the Viewport HUD: A static BorderLayout and a dynamic ProximityLayout. We also varied the representation of Connectors between annotation and referent where we used a Line, a Polygonal shape or Semi-transparent Polygonal shape. Table 7.5 shows the combination of Depth and Gestalt cues presented by the stimuli in this experiment. All annotations were visible by default and labels could be toggled on or off by selection.

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion					
Motion Parallax					
Relative/Size / Perspective					
None	***P***	***S P***	**S P**		

Table 7.5: Depth and Gestalt Cues presented by the Semantic (S) and Proximity (P) HUD techniques in Viewport Space; italics denotes the secondary independent variable

Viewport Border Layouts

To add the Proximity cue to our Viewport layouts, we adapted our BorderLayout HUD from our Object vs. Viewport experiment. The first prototype for this experiment does not provide the Proximity relation between the annotation and referent: annotations are arranged semantically with landmarks across the top (N) and bottom (S) and then molecules alphabetically down the left (W) and right (E) hand sides finally filling in the bottom (S); see Figure 7.16 top row. In this way, an annotation's location in the HUD is static and determined by the structure of the abstract information.

One problem with the semantic layout structure to the HUD is that there is no conformal relation between spatial and abstract information. As a result, it is a common case that connector lines cross the field of view, both occluding the scene and adding ambiguity to referential relations. To address this issue, we devised a LayoutManager that would locate annotations in the nearest border container to their object's projection.

We created a second version of the BorderLayout that listens to each Semantic Object's computation of its screen projection. The LayoutManager uses this information to compute which container (N,S,W,E) is the closest to the object's projection and adds the annotation to that container. If a container is full, the next-closest container is tried until an open slot is found. The rule for determining the nearest-container is based on quadrants whose axis direction is from screen corner to screen corner. The LayoutManager only updates the layout while the while the user is traveling – when an object's projection changes quadrant. This technique is illustrated in Figure 7.16, bottom row.

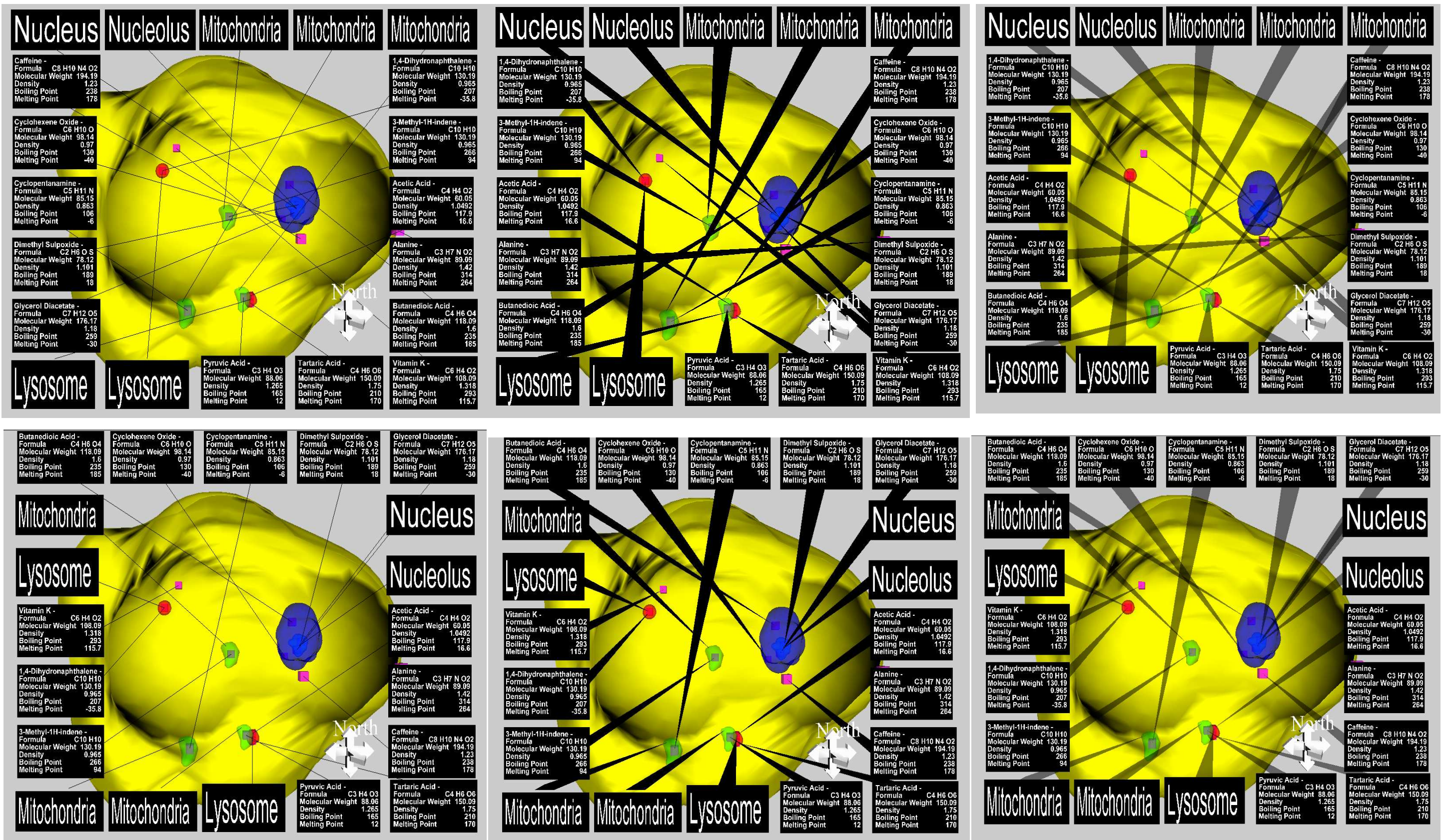


Figure 7.16: Example stimuli for each condition used in this experiment: Semantic Layout and Proximity layout are top and bottom rows respectively. From left to right, the columns show Line Connector, Polygonal Connector, and Semi-Transparent Polygonal Connector.

Connector Geometry and Appearance

We designed three different types of visual connectors to relate annotation and referent. The first type of connector is similar to that used in the Object and Viewport Space shown in prior chapters and simply consists of a colored line drawn between the SemanticObject's lineOrigin in the 3D scene and the annotation's slot on the 2D HUD border. This connector type is shown in the first column of Figure 7.16.

The second type of visual connector was an IndexedFaceSet consisting of 3 triangles drawn between 4 points. The first two points are defined as with the Line Connector above (one at the Semantic Object and one at the annotation's border slot). The two points remaining points are defined as offset from the annotation's border slot. Faces were drawn between these points to provide a strong visual Connectedness. In the conditions tested, polygons were drawn with a flat black material. The third type of connector (third column of Figure 7.16) was the same polygonal geometry and material, but the material was set to a transparency of 0.7. Depending on which HUD border container an annotation is assigned, its connector geometry may need to be redrawn on a new side.

7.2.2 Method

Experimental Conditions

In this experiment, we are interested in the effectiveness of Gestalt cues for IRVE information design. The experimental design is shown in Table 7.6. The experiment was conducted in a quiet room with low overhead lighting and a standard desktop machine with and monitor. Nineteen subjects participated in the experiment and were drawn from the VT undergraduate research pool. There were 8 males and 11 females; 13 had experience with computer games and 6 did not. The monitor CRT resolution was set to 1280x1024; Cortona VRML client was used with OpenGL in full-screen mode at 32 bit color.

Layout technique	**Connectedness**
Proximity BorderLayout	Line
Semantic BorderLayout	Polygon
	Semi-transparent Polygon

Table 7.6. Experimental design for the Viewport Space association experiment: 2 x 3 = 6 within-subjects conditions.

The same experimental protocol was followed as was used in Object Space experiment (Section 7.1.2). This includes the same training regimen and the same stimuli set. In addition, the same tasks and task types as the Object Space experiment were used (Table 7.3). Trials were administered in a random order and we measured accuracy, time to completion (seconds), navigation distance, difficulty (1-6), and satisfaction (1-6). In addition, Cognitive battery tests for Hidden Patterns and Number Comparison were administered to each participant before the trials. Experimental materials and result tables are included in Appendices D & F.

7.2.3 Detailed results

We constructed a General Linear model of the experimental data and for each dependent variable and performed an ANOVA among the conditions. Where significant interactions were found, t-tests were performed to parse out which combinations caused significant effects. These results illuminate important design issues with Viewport Space layouts on desktop displays. The results are detailed in this section and summarized in the next section.

Accuracy

Over all tasks, there were no significant effects on accuracy for either Layout or Connectedness. For Search tasks, there was a significant interaction between Layout and Connectedness ($F_{2, 17}$= 6.870; p = .007); this interaction is shown in Figure 7.17. Pairwise t-tests reveal that under the Semantic layout, the low accuracy of the Line connector (81.6%) is significantly less than Polygon (t_{18} = -2.364; p = .030) or Semi-transparent Polygon (t_{18}= -2.882; p = .010.), both of which resulted in scores of 97.4%. For Polygonal connectors, Semantic layout was significantly more accurate than Proximity layout (81.6%) (t_{18}= -2.364; p = .030). Finally, the difference between Semantic layout with Semi-transparent connector and Proximity layout with Polygonal connector was almost significant (t_{18}=-2.051; p = .055).

In addition, there is a significant interaction for Search tasks if gender is considered a between-subjects variable. Males preformed better with the Proximity layout over the Semantic layout (90.5% vs. 86.9%), but female participants did worse with the Proximity layout (83.9% vs. 94.2%) ($F_{2, 14}$= 7.368; p = .007). Visual inspection with error bars shows that the greatest difference is due to females' use of the layouts – they were much better with the Semantic HUD than with the Proximity HUD. But this difference was not significant by t-test (t_{10} = -1.641; p = .132).

For accuracy on Comparison tasks, there was a significant effect of Connectedness ($F_{2, 17}$= 3.677; p = .047). Statistically, Polygonal (75%) and Semi-transparent (72.4%) connectors are comparable. The Line connector (84.2%) is significantly better than Polygonal (t_{18}= 2.11; p = .049) and significantly better than Semi-transparent (t_{18}= 2.673; p = .016).

The interaction between Layout and Connector (shown in Figure 7.18) was significant ($F_{2, 17}$= 3.677; p = .002) for Comparison tasks. There are many significant pairwise differences in this interaction. In fact, all combinations are significant *except*: Line and Semi-transparent connectors under Proximity layout, Proximity Line vs. Semantic Semi-transparent, Proximity Polygon and Semantic Line, Also, there is no statistical difference between layouts when the Semi-transparent polygonal connector was used and no statistical difference between Semi-transparent and Polygonal connector when the layout was structured Semantically.

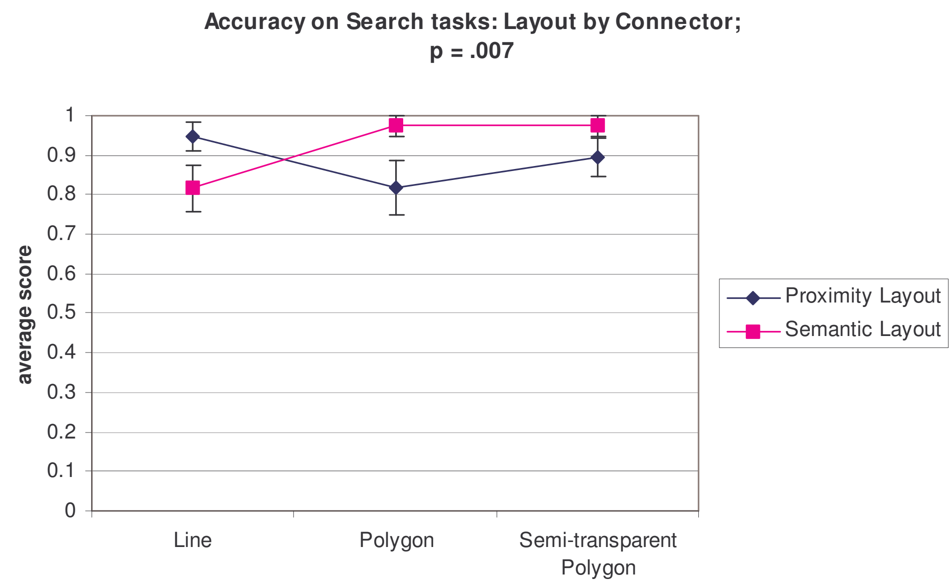


Figure 7.17: Interaction of Layout and Connectedness for Search task accuracy.

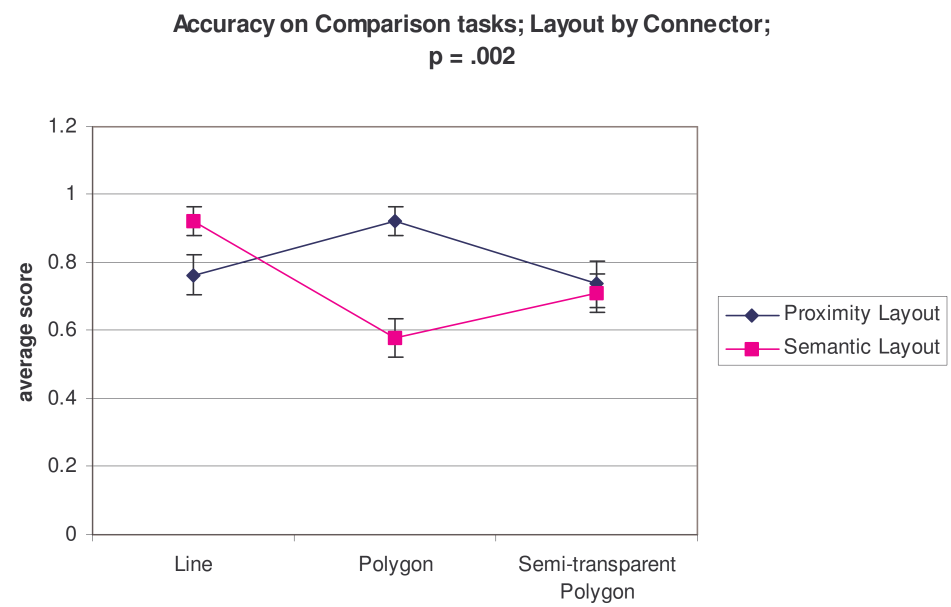


Figure 7.18: Interaction of Layout and Connectedness for Comparison task accuracy.

In addition, there was an effect of Layout when experience with computer games was considered a between-subjects variable ($F_{2,\ 14}$= 5.360; p = .035). For game players, there was no difference in the effectiveness of the layouts. For non-gamers however, the Proximity layout was a significant advantage giving correctness scores of 98.3% vs. 78.3% on the Semantic layout (t_5= 6.325; p = .001) and significantly outperforming gamers in any condition.

There were no significant effects of Layout or Connectedness depending on the information mapping. Also, there were no significant effects of gender, or game experience.

Time

We analyzed raw time to completion for each condition. Over all tasks, there was a main effect for Layout on completion time. Semantic was faster than Proximity by 38.5 vs. 48.3 seconds ($F_{1, 18}$= 20.815; p < .001). This is significant primarily due to Search Tasks performance where again Semantic layout was significantly faster (35.8 seconds vs. 49.2) by $F_{1, 18}$= 10.919; p = .004.

Also for Search tasks, there was a significant effect of Connectedness ($F_{2, 17}$= 7.687; p = .004). In this case, the Semi-transparent polygon (44.5 sec.) and the Line connectors (47.2 sec.) were comparable. Polygonal connector (36.0) was significantly faster than both Line (t_{18}= 3.211; p = .005) and Semi-transparent (t_{18}= -1.974; p = .064) connectors (Figure 7.19).

For Comparison tasks, there was a significant effect of Connectedness ($F_{2, 17}$= 11.390; p = .001). In this case, the Semi-transparent polygon (46.8 sec.) and the Polygonal connector (49.4 sec.) were comparable. The Line connector (36.6 sec.) was faster than the Polygonal connector Line (t_{18}= -3.890; p = .001) and faster than the Semi-transparent connector Line (t_{18}= 3.910; p = .001); see Figure 7.20.

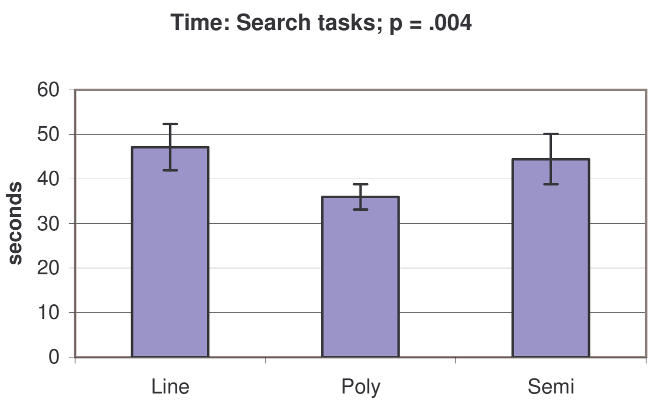


Figure 7.19: Effect of Scaling on completion time for Search tasks

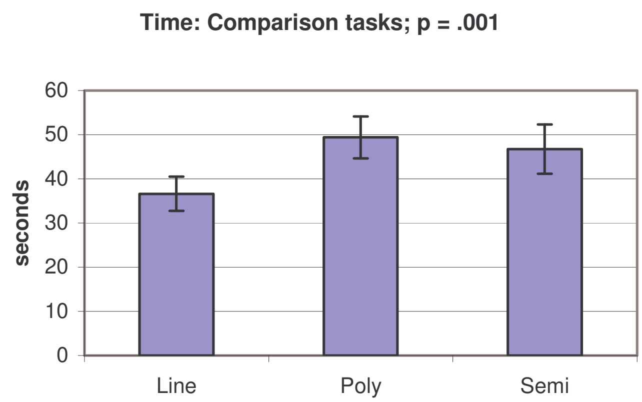


Figure 7.20: Effect of Scaling on completion time for Comparison tasks

For task information mapping, there was a significant effect of Layout when the task was abstract to spatial (A->S) at $F_{1, 18}$= 23.391; p < .001. On average for the A->S information mapping, the Proximity layout was significantly slower (59 sec.) than the Semantic layout (36.1 sec.). For spatial to abstract task time (S->A), the same pattern was almost significant ($F_{1, 18}$= 4.152; p = .057). There were no effects of gender or computer game experience.

Navigation Distance

There were no effects of gender or computer game experience on distance navigated over all trials. Over all tasks, there was also no significant main or interaction effect for Layout or Connectedness on distance navigated; this was also true for Search tasks. For Comparison tasks however, Connectedness was significant: ($F_{2, 17}$= 9.386; p = .002). Polygonal connector (646.4 world units) was comparable to Semi-transparent polygon (642.6 world units). The Line Connector (490.2 world units) required significantly less navigation than the Polygonal connector (t_{18}= -3.474; p = .003) and the Semi-transparent polygon connector (t_{18}= -3.950; p = .001).

When the task information mapping was abstract to spatial (A->S), there was a significant effect to Connectedness ($F_{2, 17}$= 11.936; p = .001). Here the Polygonal connector and the Semi-transparent polygon were comparable (421.9 vs. 434.2 world units respectively). The Line connector (304.4 world units) required significantly less navigation than the Polygonal connector (t_{18}= -3.654; p = .002) and the Semi-transparent polygon connector (t_{18}= -3.275; p = .004).

When the task was A->S, there was also a significant interaction between Connectedness and Layout $F_{2, 17}$= 5.952 P = .011. This interaction is shown in Figure 7.21. In this interaction, the Semantic Layout performed consistently across connector type. For the Proximity Layout however, the Line connector conditions resulted in significantly less navigation than Polygonal connector (t_{18}= -4.482; p < .001) and Semi-transparent polygon connector (t_{18}= -2.679; p < .015). In addition, the Proximity Layout with Line connector was also better than the Semantic layout with Semi-transparent polygon connector (t_{18}= 2.782; p < .012). Finally, the Semantic Layout with Line connector was better than the Proximity Layout with Polygon connector (t_{18}= 2.721; p = .014). When the tasks were of the S->A information mapping, there were no effects of Layout or Connectedness.

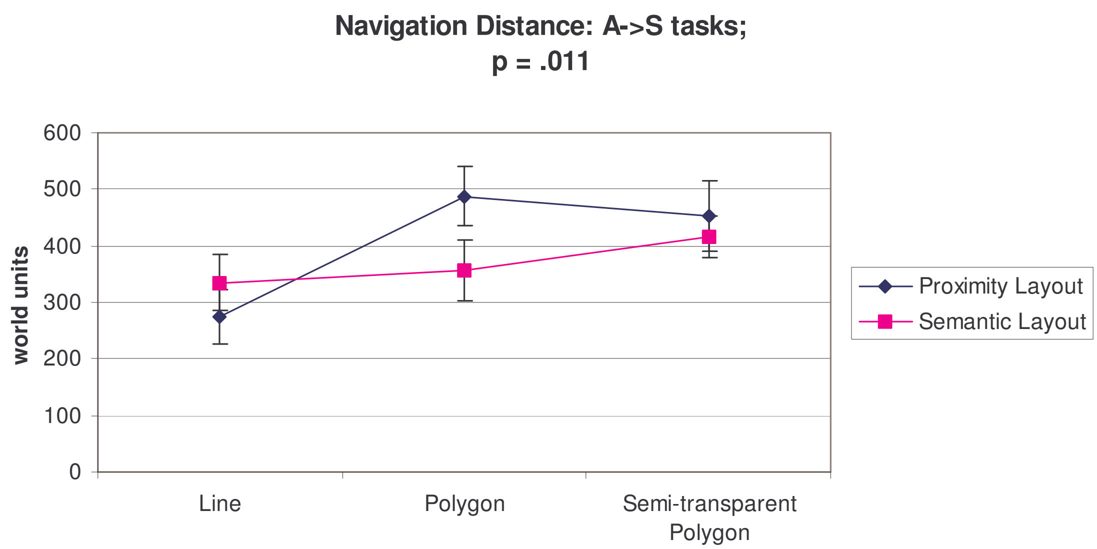


Figure 7.21: Interaction of Layout and Connector on Distance Navigated for A->S tasks

Difficulty & Satisfaction

Over all tasks, the Semantic Layout was rated least difficult (2.20 vs. 2.55) by $F_{1, 18}$= 46.002; p < .001 and most satisfying (4.62 vs. 4.29) by $F_{1, 18}$= 14.682; p = .001. This pattern of subjective ratings was primarily due Search tasks. For Search tasks, the Semantic Layout was rated least difficult (2.10 vs. 2.67) by $F_{1, 18}$= 31.117; p < .001) and most satisfying (4.67 vs. 4.22) by $F_{1, 18}$= 10.778; p = .004.

For Search tasks, Connectedness was a significant influence on user ratings for difficulty ($F_{2, 17}$= 5.523; p = .014) and for satisfaction ($F_{2, 17}$= 5.086; p = .019). The Line and the Semi-transparent polygon connectors were comparable for difficulty rating (2.54 and 2.44 respectively). However, the Polygonal connector was rated significantly less difficult (2.18) than both Line (t_{18}= 2.721; p = .014) and Semi-transparent (t_{18}= -2.371; p = .029) connectors. Connectedness was also a significant influence on user ratings of satisfaction for Search tasks ($F_{2, 17}$= 5.086; p = .019). In this case, the significant difference is that users rated the Polygonal connector more satisfying than Semi-transparent polygon connector (4.67 vs. 4.26), t_{18}= 2.925; p = .009.

Connectedness was also a significant influence on user ratings for difficulty on Comparison tasks ($F_{2, 17}$= 12.929; p < .001). In this case, Line connector and Semi-transparent polygon connector were comparable for difficulty rating (2.10 and 2.36 respectively). However, the Polygonal connector was rated significantly more difficult (2.59) than both Line (t_{18}= -4.949; p < .001) and Semi-transparent (t_{18}= -2.454; p = .025) connector. For Comparison tasks, the Semantic Layout was more rated as more satisfying than the Proximity Layout (4.57 vs. 4.37), $F_{1, 18}$= 4.954; p = .039.

When the task information mapping was abstract to spatial (A->S), there was a significant effect of layout. As we might expect, the Semantic layout was rated less difficult (1.92 vs. 2.53) ($F_{1, 18}$= 19.661; p < .001) and more satisfying (4.78 vs. 4.34) ($F_{1, 18}$= 15.990; p = .001) than the Proximity layout. When the tasks were of the S->A information mapping, there were no main effects of Layout or Connectedness for difficulty or satisfaction ratings.

Cognitive battery scores

We calculated correlations between subject's cognitive battery scores using Pearson's R. Similar to our General Linear Model, we computed combinations for both task types and task information mapping. For accuracy, we found two significant negative correlations (better test score -> worse correctness). For Accuracy measures, we found a significant correlation between user's Number Comparisons score and their performance on Search tasks A->S (R = -.542; p = .017). User's Number Comparisons score also negatively correlated with performance on Semi-transparent connectors under S->A tasks (R = -.476; p = .039).

For our dependent measure of time-to-completion, we found a significant positive correlation between user's Hidden Figures scores and their performance with the Proximity Layout with Line connector (R = .456; p = .049).

Numbers Comparison test also had two significant correlations. The first was between Numbers Comparison and Difficulty ratings on the Semantic Layout with A->S information mappings (R = -.579; p = .009). Since this is a negative correlation, this means that better scores relate to lower difficulty ratings. The Second, for Satisfaction is also negative to the Semantic Layout on S->A information mappings (R = -.466; p = .045), which means that better scores relate to lower satisfaction ratings.

7.2.4 Results Summary

Proximity

We can see that over most measures (time overall and for Search tasks as well as difficulty and satisfaction overall) the Semantically-structured border layout provides significant advantages over the Proximity Layout. For example, it is faster overall and for Search tasks. In addition it is rated less difficult and more satisfying over all tasks. For A->S tasks, the Semantic Layout was faster and considered less difficult and more satisfying.

The games effect is significant where non-gamers were significantly better with the Proximity Layout in terms of Accuracy. Not only were they better with Proximity Layouts, but their performance with Proximity Layouts beat gamers under both layout conditions. The gender effect, where females are better with the Semantic Layout rather than the Proximity layout, is interesting since performance across layouts was comparable for male subjects; however, it was not shown to be statistically significant by t-test.

In the interactions of Layout and Connectedness for Accuracy, we see some advantages for the Proximity layout depending on the Connector type used. For Search Tasks, Proximity Layout plus Line connector is a top performer and is significantly better than Semantic Layout plus Line connector. For Comparison tasks, Proximity Layout plus Polygonal connector is one of the top performers and is significantly better than Semantic Layout plus Polygonal connector.

Layout and Connectedness also interacted with Navigation Distance for A->S information mappings. Proximity Layout was one of the best performers when it was used in combination with the Line connector; it was one of the worst when it was used in combination with the Polygon connector.

Connectedness

In general, Polygonal connectors were better for Search tasks and Line connectors were better for Comparisons. This was true for measures of accuracy, time, navigation distance, difficulty, and satisfaction. Line connectors were also better in terms of Navigation Distance when the task was of the A->S information mapping.

7.2.5 Conclusions

From the results reported above, we can make a number of conclusions regarding the effectiveness of Association cues in Viewport information design.

First, regarding the Proximity cue, we note that the advantage of the Semantic Layout structure was fairly comprehensive. This was counter to our first hypothesis since we believed that a Proximity relation would result in cleaner layouts and less ambiguity of reference. As in the Object Space experiment described in Section 7.1, we note problems with the dynamic aspect of the layout. Unlike the continuous re-arrangement used in the ForceDirected Layout, in this experiment annotations were only re-arranged while the user was navigating. While this reduced motion in the visual field while the user was stationary, the (managed) re-arrangement method is still a net liability.

We can interpret this result in light of our observations of user strategy. Naïve users (who are not familiar with the data set) do not seem to store abstract information in their head. Instead, they use the annotation's location in the visual field as an index to access abstract information as needed. This is an understandable strategy since it is more economical in terms of cognitive load and probably less error-prone. The results also support observations by Zhang and Norman [Zhang, 1994] that users employ the visual display as an external working memory store.

The significant and near-significant effects of Gender and Game may also indicate different user strategies. For example, the gender differences on accuracy for Search tasks suggests that females use a more verbally-oriented strategy to task completion and they are better served by the Semantic Layout. In the case of non-gamers making Comparisons, they are well-served by the Proximity Layout. We speculate that this is because this user group has fewer assumptions about a how a HUD readout 'should' work and so used the Proximity cue to lookup information rather than using a location in the visual field as an index.

The results for Connectedness support prior evidence that the resolution to the Occlusion-Association tradeoff is specific to different IRVE tasks and task information mappings. For Search tasks, the strong Gestalt association cue of Connectedness provided by the Polygonal connector is advantageous. For Comparison tasks, a weaker Connectedness cue via the Line connector was advantageous. Therefore we claim that for Search tasks, it is better for display techniques to provide strong Association, even though this introduces more Occlusion. In contrast, less Occlusion is better than more Association for Comparison tasks. Also for the abstract to spatial (A->S) information mapping, less Occlusion and Association is advantageous (Line connector).

There are interesting cases shown in the interaction effects that are also worth noting in this light. First, in cases where high Association is advantageous (such as Search tasks), it appears that the Proximity Layout can compensate for the low Connectedness of the Line connector in terms of Accuracy performance. Second, in situations were low Association is favored (such as Comparison tasks), the Proximity Layout can compensate for the strong Connectedness of the Polygonal connector (again for Accuracy).

Third, we have observed that when the task criteria is abstract and the information target is spatial (A->S), low Association such as the Line connector is advantageous. For the Navigation Distance on tasks of the A->S information mapping, the Proximity Layout was a poor performer with Polygonal connectors. In this task information mapping, the same algorithm that gives the Proximity Layout the ability to compensate for strong Connectedness is also reduces its effectiveness.

7.3 Post-hoc Comparisons

7.3.1 Design

The biological cell stimuli and task set used in the Object Space and Viewport Space experiments were identical. Overall, the two experiments differed in two important respects: first, the layout techniques used, and second, the screen size. Therefore, we consider the question of if we can use particular conditions to generally compare the aggregated effect of techniques and screen sizes overall.

In order to consider these two experiments comparable, we note the following:

- In the Object Space experiment, all conditions used an opaque polygon connector between the annotation and the referent. In 1 condition, annotations were not scaled for guaranteed legibility.
- In 2 conditions of the Viewport Space experiment, polygonal connectors were used; in all conditions the annotations were always legible.

If we omit one specific condition from each of the experiments (Object space with no scaling and Viewport space with line connector) from the analysis as non-comparable, we can still reasonably compare conditions across all the task information mappings. We aggregate the conditions by technique and then compare across subjects. Analyzing this on a between-subjects basis, we hope to better understand the performance spectrum of these techniques for the Association – Occlusion tradeoff.

We have defined a 2-dimensional design space for IRVE Layout Attributes that provides a rational framework to assess the effect of perceptual cues on information throughput. We categorized our 4 display techniques from experiments 3 and 4 along the diagonal Association-Occlusion dimension of our IRVE information design space. This dimension spans High Association and High Occlusion (which reside in the top left corner) and Low Association and Low Occlusion (which reside in the lower left corner).

Object Space ScreenBounds and Viewport Space Proximity Layouts were combined into 'High Association' category; ObjectSpace ForceDirected and Viewport Space Semantic were combined into 'Low Association' category. In these aggregated display techniques (SB & P) and (F & P), different combinations of consistent cues disambiguate the relation between annotation and referent. The cues present in each technique is shown in Table 7.6 and Figure 7.22. High Association and Low Association are similar except that High Association includes more Occlusion and more Proximity.

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	***H***	***H***	***H***		
Motion Parallax	***H L***	***H L***	***H L***		
Relative/Size / Perspective	***H L***	***H L***	***H L***		
None	***H***	***H L***	***H L***		

Table 7.6: Depth and Gestalt Cues presented by the aggregated Low (L) and High (H) Association techniques in the post-hoc analysis of Experiments 3 and 4; italics denotes a cue whose effect is diluted by averaging (F & S) and (SB & P).

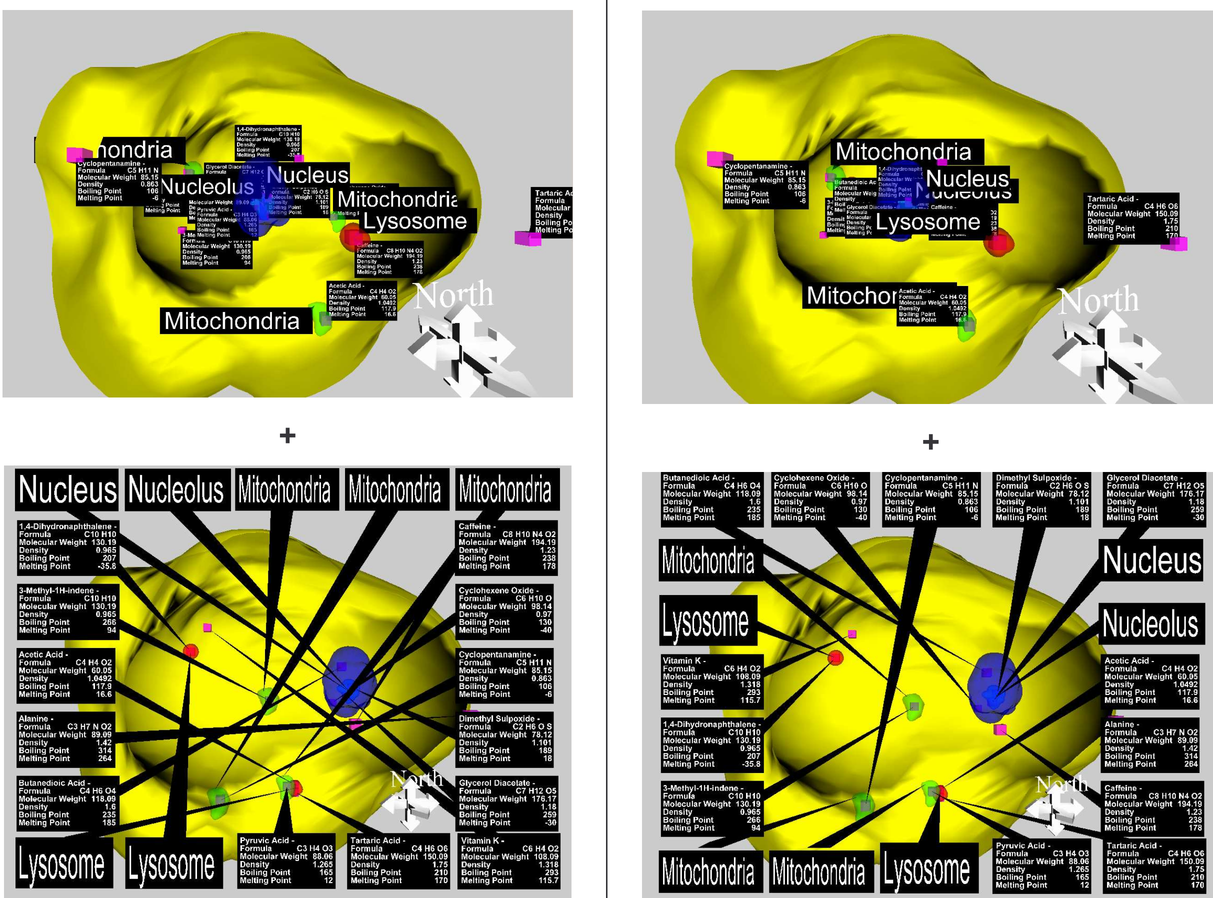
Thus, the post-hoc model is analyzed with a mixed 2 x 2 design. All subjects used both High and Low Association Display Techniques. These techniques were used on either a large-screen or a desktop display.

Based on the experimental results reported in the last two chapters, we hypothesize that:

1. High Association techniques will be:
    a. Advantageous for Search tasks
    b. Advantageous for S->A task information mappings
    c. Advantageous for the large display

2. Low Association techniques will be:
    a. Advantageous for Comparison tasks
    b. A->S task information mappings
    c. Advantageous for the desktop displays

We analyzed all objective measures (Accuracy, Time, and Distance Navigated) using a General Linear Model and ANOVA. Within-subjects conditions were aggregated to the Low and High Association techniques. Screen size and Spatial resolution are combined as a single between-subjects factor we will call ‘Display Context’. When significant effects are shown, t-tests are reported for specific combinations.



Low Association = ForceDirected + Semantic HUD	High Association = ScreenBounds + Proximity HUD

Figure 7.22: IRVE Display Techniques merged for post-hoc analysis

7.3.2 Results

Descriptive statistics for overall results from this analysis are also included in Appendix G.

Accuracy

Over all task types and information mappings, there were no significant differences between Display Contexts. But, there was a significant main effect overall for Display Technique where the High Association techniques were more accurate (84.4% vs. 77.9% correct) where $F_{1, 30}$= 5.466, p = .026. This effect arises from the advantage of High Association for Comparison tasks. For Comparison tasks, Low Association Display Techniques were more accurate (High = 81.8% Low = 97.7% correct) at $F_{1, 30}$= 10.787, p = .003.

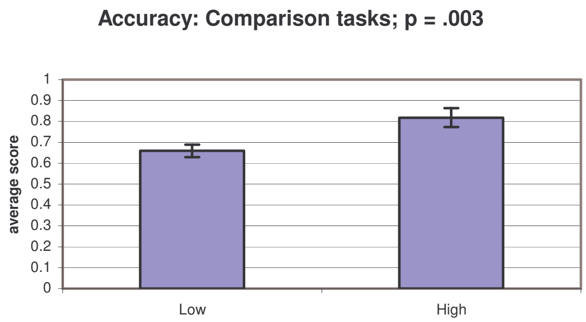


Figure 7.23: Effect of Association Technique for Comparison task Accuracy

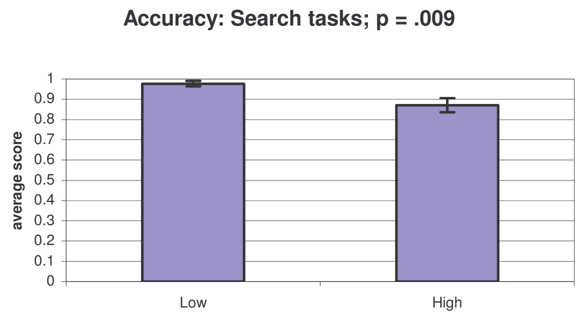


Figure 7.24: Effect of Association Technique for Search task Accuracy

The effect of Display Technique on accuracy was reversed for Search tasks. The High Association Display Techniques (87.0%) were better than the Low Association Display Techniques (65.9%) by $F_{1, 30}$= 7.820, p = .009. This contrast is shown in Figures 7.23 and 7.24 respectively. There were no significant effects of Display Technique or Display Context for either A->S or S->A information mappings.

Time

There was also a significant main effect for Association Display technique for Time over all tasks ($F_{1, 30}$= 7.837, p = .009). Over all tasks, Low Association (59.4 sec.) was significantly faster than High Association (68.12). This is likely due to performance on Search tasks where there was a significant effect for Association Display technique: $F_{1, 30}$= 21.409, p < .001. For Search tasks, Low Association (47.0 sec.) was significantly faster than High Association (67.4 sec.). This relation is depicted in Figure 7.25. For Comparison tasks, the there was no main effect for Association Display technique.

There was also a significant effect of Display technique for abstract to spatial (A->S) information mappings ($F_{1, 30}$= 18.453, p < .001). For A->S tasks, Low Association (62.3 sec.) was significantly faster than High Association (81.039 sec.). This relation is depicted in Figure 7.26. There was no main effect of Association Display technique for S->A mapping.

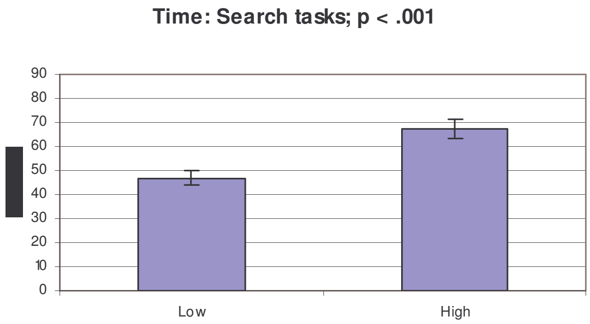


Figure 7.25: Effect of Association Technique for Search task Time

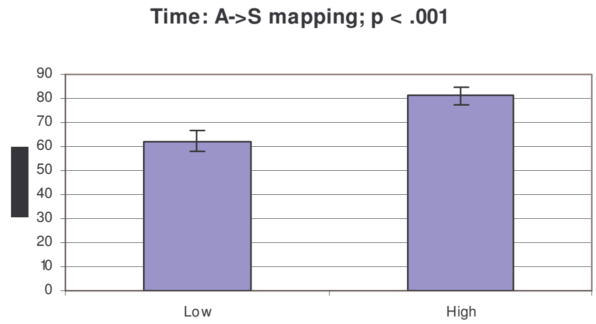


Figure 7.26: Effect of Association Technique for A->S Time

Display Context was a significant between-subjects factor overall ($F_{1,\ 30}$= 28.536; p <.001) where the Large screen was significantly slower than the desktop (83.4 vs. 44.151 seconds). This pattern was true for all task types and information mappings:

- Display Context provided a significant effect for Search tasks ($F_{1,\ 30}$= 31.267; p < .001) where the Large screen was significantly slower than the desktop (74.1 vs. 40.2 seconds).
- Display Context provided a significant effect for Comparison tasks ($F_{1,\ 30}$= 19.878; p < .001) where the Large screen was significantly slower than the desktop (92.6 vs. 48.1 seconds).
- For A->S information mapping, Display Context was significant ($F_{1,\ 30}$= 64.359; p < .001) where the Large screen was significantly slower than the desktop (97.8 vs. 45.5 seconds).
- For the S->A mapping ($F_{1,\ 30}$= 7.036; p =.013) where again the Large screen was significantly slower than the desktop (69.0 vs. 42.8 seconds).

Distance Navigated

There were no effects for Association strength overall or for Search tasks specifically. For Comparison tasks, there was a significant effect of Association ($F_{1,\ 30}$= 6.317; p =.018) where High Association required less navigation than Low Association techniques (777.3 vs. 1030.8 world units).

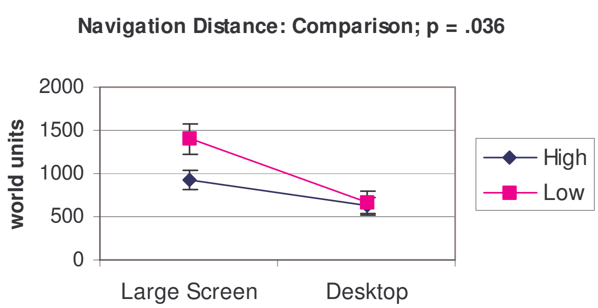


Figure 7.27: Effect of Association Technique and Display Context on Navigation Distance for Comparison Tasks

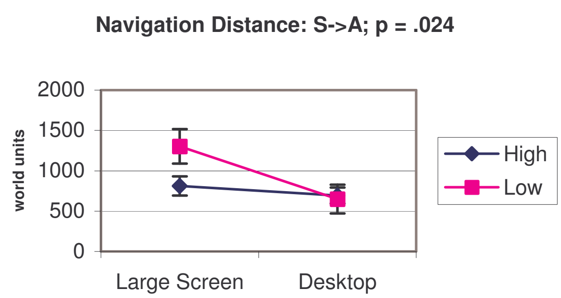


Figure 7.28: Effect of Association Technique and Display Context on Navigation Distance for S->A Tasks

Finally, there are two significant interactions between Association and Display Context that must be reported. They both follow the same pattern. First there was a significant interaction for Comparison tasks ($F_{1,\ 30}$= 4.832; p =.036) and second there is an interaction for S->A information mapping ($F_{1,\ 30}$= 5.678; p =.024). This interaction, shown in Figures 7.27 and 7.28 arises from the fact that High Association layouts are relatively stable across Display Contexts. While Low Association is comparable to High Association in both contexts, on the large display Low Association is clearly the worst performer.

Display Context was a significant between-subjects factor overall ($F_{1, 30}$= 12.822; p =.001) where the Large screen caused significantly more travel distance than the desktop (1007.5 vs. 549.1 world units). This pattern was true for the following task types and information mappings:

- Display Context provided a significant effect for Search tasks ($F_{1, 30}$= 12.002; p =.002) where the Large screen caused significantly more travel distance than the desktop (851.5 vs. 453.6 world units).
- Display Context provided a significant effect for Comparison tasks ($F_{1, 30}$= 11.016; p =.002) where the Large screen caused significantly more travel distance than the desktop (1163.5 vs. 644.5 world units).
- For A->S information mapping, Display Context was significant ($F_{1, 30}$= 45.364; p <.001) where the Large screen caused significantly more travel distance than the desktop (959 vs. 428 world units).

7.3.3 Summary & Conclusions

The experimental evaluations described in this chapter have contributed to our understanding of principal tradeoffs in IRVE information design. This section reviews the implications from our post-hoc analysis of IRVE display techniques and objective performance measures. We reflect on our hypotheses and consider how the evidence can be captured and summarized for information design guidelines.

In order to compare the data from the two experiments described in this chapter, we aggregated conditions that provided different numbers of visual cues relating annotation and referent into 'High' and 'Low' Association techniques (Table 6.6). The visual cues of our design space serve to reduce ambiguity about the relation between abstract and spatial information. In the terminology of Information Theory [Shannon, 1963], the Depth and Gestalt cues could each be considered bits of information.

Connectedness and Common Fate were the two cues fully represented in both aggregated Association conditions. In addition, we aggregated differences in Display size, and Display spatial resolution into a between-subjects variable called 'Display Context'. The Display Context has two levels: Large and Desktop depending on which experiment the data is from (Experiment 3 or 4).

Association

Over all tasks and information mappings, High Association was more accurate. This was a result we did not predict. Most of the evidence compiled to this point indicated that strong Association was not necessary for good performance. The result may be interpreted in conjunction with the Time results where Low Association was faster overall, especially for Search tasks and A->S tasks mappings. This shows there may be a speed–accuracy tradeoff parallel to the Association–Occlusion tradeoff. Overall, High Association layouts are more accurate but take longer to use.

For our hypotheses about the effects of Association and task-type, we had made our hypothesis based on the Connectedness data of Experiment 4. In that case, the increased Association from a strong Connectedness cue was advantageous for Search and a liability for Comparison. Contrary to our hypothesis, Low Association was better for Search and High Association was better for Comparison (in terms of accuracy). Also in this pattern are other significant results: Low Association was faster overall and for Search tasks but High Association required less navigation for Comparison tasks.

These results suggest that perceptual cues relating abstract and spatial information are not treated uniformly by the users perceptual and working memory systems. If they were, we would expect a more symmetric pattern of results. Consider for example that these post-hoc results indicate that there are other cues, such as Occlusion and Proximity that are stronger than Connectedness and Common Fate (which were common to both aggregated layouts).

In prior experiments, low Association was sufficient to be advantageous for most tasks. However in this case, the post-hoc analysis shows that Occlusion and Proximity (higher Association) are significantly beneficial to performance of Comparison tasks. In contrast, the High Association from these cues are liabilities for Search tasks. We defer a full interpretation of these results to the next chapter. For now, it is sufficient to claim that this post-hoc analysis complements our previous empirical investigations by providing additional data as to when particular cues may be more or less important depending on the task or the task mapping.

Display Context

First, we note that Accuracy performance was not affected by the Display Context used between Experiments 3 and 4. Second, we note that time and navigation distance were significantly effected by the Display Context between-subjects condition. Overall and for most task types and information mappings, the Large Screen Display Context was worse than the Desktop context. This was a strong effect and was not predicted based on Experiment 1 (Object vs. Viewport portability) where display size did not have such an effect.

This effect could be attributed to three possible reasons. First, the ergonomics of the stool and podium made the mechanics of mouse navigation and selection unfamiliar or difficult. Second, the lower spatial resolution of the Large Screen condition made text harder to read. Related to this difference in spatial resolution is the third possibility that the mouse cursor was harder to track on the large screen.

While the ultimate cause of this discrepancy is left for future investigation and explanation, designers are encouraged to address these issues when considering usability of IRVE information designs on large, low-resolution displays. For example, they must consider the usability cost of time and navigation distance when porting IRVE applications from desktops to large projection displays. We speculate that these costs may be mitigated by adding resolution and/or stereoscopy to the display itself or by improving the ergonomics of input devices.

8. Conclusions and Recommendations

8.1 Conclusions

In this research, we focused on development and testing of layout algorithms for IRVEs. The display techniques and empirical evaluations undertaken in this research provide the basis for IRVE developers to understand the usability consequences of their design choices. By understanding the perceptual properties and the performance impacts of various IRVE display techniques, interface designers can customize their presentation to various platforms such as single, tiled, or large-screen displays.

Specifically, we present a systematic program to understand how fundamental perceptual properties affect user performance. In this work, we detail a methodology to assess, design, and deliver appropriate IRVE information displays per task and data type. This program includes an IRVE design space, an IRVE task taxonomy, IRVE display components, and empirical data regarding the effectiveness of various design choices. This section summarizes the novel insights and design guidelines that result from this research.

In the course of this research we ran a number of studies to assess how successfully various display techniques resolve the Association-Occlusion tradeoff and the Legibility-Relative Size tradeoff. We began with one initial user study to qualitatively survey IRVE information design issues. The survey demonstrated the feasibility and issues involved in constructing usable IRVE interfaces in immersive environments such as the CAVE. We then ran four formal usability studies to collect quantitative performance data by objective and subjective measures.

The empirical studies manipulated the design dimensions of Layout Space, Association, and Display Size in order to understand how various Depth and Gestalt cues interact in IRVEs. We asked "What are the best ways to manage layout space and visual associations so that perceptual and abstract information can be understood together and separately?".

For the requirement of relating annotation and referent in an IRVE interface, we tested Search and Comparison tasks for both: spatial criteria and abstract targets and also abstract criteria with spatial targets:

- In the first two formal experiments, we compared the effectiveness of Layout Spaces in various situations such as desktop, large screen displays, and projection distortions (Software Field-Of-View). The first set of evaluations provided an overall picture on the relative value of Layout spaces for different display situations; we tested Object Space vs. Display Space and Object Space vs. Viewport Space.
- In the second two formal experiments, we sought to understand what factors make an effective technique in particular Layout Spaces. The second set of evaluations compared techniques within Layout Spaces. For an understanding of significant design parameters in Object Space, we tested on a large screen display; we then compared two Viewport techniques on a desktop display.

8.1.1 Experiment Summary

From the first experiment, we can see that overall the Viewport Space was advantageous over Object Space - the tight coupling (via Gestalt and Depth cues) provided by Object Space was not advantageous enough to overcome the occlusion problem. However, the tight-spatial coupling (via Gestalt and Depth cues) that is provided by Object Space was advantageous for Comparison tasks, especially when the display size was large and/or the Software Field of View was high.

In the second experiment, we can see that the tight coupling (via Gestalt and Depth cues) that is provided by Object Space was advantageous when the task mapping was Comparison: S->A. However, on most other counts, a loosely coupled Display Space was advantageous over Object Space - the loose coupling provided by Display Space was more than compensated for by the minimal occlusion and the addition of attribute-centric visualizations.

Both experiments show there are indeed tasks and situations where high Association is advantageous enough to overcome the liabilities of high Occlusion. However, overall the benefits of minimal occlusion (such as the Viewport or Display Space techniques tested) seem strong enough to compensate for the ambiguities of minimal Association.

In Experiment 3, we varied Depth cues presented by Object Space techniques. Here we note that Occlusion may be the strongest depth cue, but it is a hindrance for Search tasks. While there are cases were occlusion can be a helpful cue (such as making spatial discriminations easier), it is overall a liability.

In Experiment 4, we examined the Gestalt association cues provided by two Viewport Space layouts: Proximity and Connectedness. Again, the pattern of results is task type or information mapping -specific. The general result is that the Connectedness cue can significantly affect performance depending on the task type. Any advantage gained by increased association through Proximity was negated by the costs of re-finding annotations in motion.

The meta-analysis conducted post-hoc on Experiments 3 and 4 supports the general theme that overall performance can be improved by providing ‘just enough and no more’ Association between annotation and referent. By comparing different combinations of cues, we got results that were unexpected given the cumulative evidence. For example overall, High Association was better for task accuracy especially due to its benefits for Comparison tasks. However for Comparison tasks, High Association required more navigation. For Search tasks, the aggregated Low Association layouts performed better than the High Association layouts for accuracy and time as well as for time on A->S task mappings.

8.1.2 Association and Occlusion

We have articulated two important tradeoffs in IRVE information design: the Association-Occlusion Tradeoff and the Legibility-Relative Size Tradeoff. The *Association-Occlusion Tradeoff* can be summarized as:

> Tighter Association between annotation and referent results in more occlusion in the scene.
> More consistent Depth cues and Gestalt cues between annotation and referent (i.e. more Association):
>
> **(+)** May convey more information about the relation between annotation and referent
>
> (i.e. less ambiguity)
>
> **(-)** May cause more occlusion between scene objects and therefore less visibility
>
> of information

Recalling our guiding research question from Chapter 1, we are interested in how various IRVE display techniques can aid users in understanding the relationships between abstract and spatial information. However, we do not want layout techniques to interfere with access to either individual information type. This is where the Occlusion–Association tradeoff occurs: if more information regarding this Association between information types is conveyed by the layout, the more Occlusion can occur among and between information types. We have reported rich and varied results that suggest there are many interactions among the dimensions of the design space.

In general, we showed that insuring visibility of both spatial and abstract information types is one of the most important design concerns. By the empirical data, we have shown that advantageous user performance can be achieved with very few cues (the less Association) in an IRVE. There are, however, particular circumstances where visual configurations of high Association and high Occlusion can be advantageous. Specifically, cases where the Depth cue of Occlusion and the Gestalt cur of Proximity can be beneficial. For example, on large displays, high Software Field-Of-Views, and tasks that require accuracy in Comparisons. These impacts of the perceptual cues in our layout techniques are collected in our IRVE design guidelines (listed in section 8.2).

8.1.3 Legibility-Relative Size

The *Legibility-Relative Size Tradeoff* can be summarized as:

> If annotations are rendered with the consistent depth cue of Relative Size, they may not be legible from a distance:
>
> **(+)** Relative Size provides an additional, disambiguating cue relating annotation and referent
>
> **(-)** Relative size may require more spatial navigation to recover abstract information from the scene

This work shows that overall, the legibility of annotations is more important than the Depth cue of Relative Size. Experiment 3 (Object Space) confirms this specifically. It also presents a classic user interface tradeoff of speed and accuracy. The results show that when annotations are scaled for Legibilty, users are faster to complete the tasks but also less accurate. This also suggests that users can gain valuable spatial information by the act of navigation (to achieve Legibility).

For textual information, we showed that it is important to provide stability for fixation and reading time. In situations with naïve users, we also noticed that users rely on their memory of the location of information in the visual field rather than maintaining any declarative information in working memory. If users adopt this strategy, annotations that guarantee immediate Legibility will be faster. Finally we note that special care should be taken when rendering text in 3D environments, particularly in low resolution and stereoscopic systems.

8.1.4 Dynamic Annotation Location

The results relating to the dynamic layout algorithms are also important. It would appear that naive subjects do not store abstract information declaratively in working memory when performing comparison tasks. Instead, they rely on the location in the visual field to repeatedly 'look-up' information; they keep the information in the world rather than in their head. If the annotation has changed position, this introduces a time delay while they re-find and re-read the annotation.

While our dynamic layout techniques may portray more Association between annotation and referent, the benefit of additional Association (and less Occlusion) is not enough to overcome the liabilities of annotations in motion. It is not clear if users familiar with the data set would show the same performance profile as the naïve users who seem to rely on a stable visual layout. The results with non-gamers regarding the Viewport Layout are also interesting in this regard (they were the best with Proximity HUD). Users without computer game experience may not have the same assumptions about HUD stability and were thus able to use the Proximity cue more effectively.

8.1.5 Information Architectures

For both durability and maintenance reasons, it is desirable to store data in an expressive, machine-readable model. XML is designed for this purpose. By marking up data sets with our IRVE display component syntax, we have demonstrated that a single data source can be transformed into multiple representations. The Pipeline or Hybrid publication paradigms we have described provide the means to deliver these views dynamically over the network. The IRVE syntax and transformation framework can be extended with additional display techniques and components to further enrich online integrated information spaces.

With an engineered approach and concrete usability data, we can make a compelling case to the graphics and informatics communities to innovate and improve the technological foundations for IRVEs. This includes further development of Visualization Services and their supporting standards. For example, this research has enumerated a number of important functionalities both as new capabilities for the specification, and through component libraries such as annotations and Semantic Objects.

We have found a particularly powerful combination of International Standards and open-source server and scenegraph runtimes through declarative, networked resources such as XML and X3D, and server-side visualization services such as Cocoon, Perl, and XSLT. A number of technical capabilities for IRVE techniques are converging. One important contribution of this work is improving the X3D standard to provide better layout and rendering facilities in the target runtime.

Through the PathSim IRVE application and publishing paradigms detailed in Chapters 4 and 5, we have also described and implemented a number of server-side components to transform static and timeseries data to IRVE Visualization (e.g. Perl: txt -> VRML; Java: XML -> X3D). Some exciting frontiers for IRVE application design are seen in the trend toward better naming schemes for networked data, increased integration with rich metadata schemes, and web services that deliver IRVE content on demand. The full potential for integrated information spaces is yet to be realized as interactive 3D media and assets become first-class members of the WWW and a crucial part of the IT enterprise.

8.2 Recommendations

8.2.1 Implications for Information Design

This work begins the crucial effort of justifying IRVE information design features with empirical usability data. First, we have enumerated the design space of IRVEs and formulated a research program to understand how layout algorithms (a.k.a. display techniques) support users in integrating heterogeneous data types. Second, we have demonstrated the value of the IRVE approach in providing multiple complementary representations of information while providing a unified environment for exploration and analysis.

This work has shown that the perceptual cues provided by IRVE layouts may have different impacts on performance depending on the task, the task-information mapping, and the display size. This fact requires that designers approach IRVE design with a clear set of requirements for user activities and tasks, as well as knowledge of the display platform where the application will eventually be used. Our results contribute to the field's knowledge of how to mitigate IRVE design tradeoffs and produce effective integrated information spaces.

Readers will note that these guidelines capture the effects layout techniques and display and are organized by the task and information mapping. Because of the specificity of performance effects, we can say that one IRVE display technique does not fit all. Rather, it may be fruitful for IRVE applications to provide some logic or control that can adapt or change the layout technique depending on the user's task and information criteria/target.

There are many interesting design issues in the prototypes we have developed in this work. Because the nature of usability in IRVEs is a new area of research, we must be cautious in providing guidelines and recommendations. For example, we have done most of our empirical tests with textual annotations, but the benefits of including multiple, alternate visualizations is certainly promising (e.g. PathSim, Experiment 2). In addition, there was an equivalent amount of information presenting in all the experiments. While this allowed us to focus on a challenging volume of data and relate experiments in an initial theoretical model, it tells us little about the scalability of IRVE visualizations to different sizes and kinds of data. Both of these areas will be fruitful for future IRVE research. At this early juncture, we will only formulate guidelines for aspects of information design that our empirical research addressed.

8.2.2 IRVE Design Guidelines

Through our investigation of the IRVE information design space, we formulate the following IRVE Information Design guidelines into 2 categories: Techniques and Displays. To apply these guidelines effectively, designers should have a detailed understanding of their task and knowledge structure requirements.

Techniques

Overall

- *Choose Visibility over Occlusion & Association*
- *Increase Proximity of annotation and referent*
- *Minimize relocation of annotations*
- *For Speed, choose Legibility; For Accuracy, choose Relative Size;*
- *Reduce requirements for spatial navigation*
- *Display global attributes in a visible display area such as Viewport or Display space*
- *Collect common object attributes in visible display area such as Viewport or Display space and connect multiple views with Common Fate (e.g. brushing and linking interaction)*

Search Tasks

- *Choose Visibility over Occlusion*
- *Choose strong Connectedness*

Comparison Tasks

- *Choose Minimal Connectedness*

A->S

- *Choose Legibility*
- *Choose Minimal Connectedness*

S->A

- *For Speed, choose Legibility; For Accuracy, choose Relative Size;*

Displays

Overall

- *When the display size is large, increase Proximity*
- *For Speed, consider the challenges presented for large displays such as input mappings and framerate*
- *Pay special attention to text rendering on large screens and in stereoscopic renderings*
- *Reduce interaction modes*
- *Provide visibility of mode status*

Search Tasks

- *Increase Software Field of View (SFOV)*

Comparison Tasks

- *Decrease Software Field Of View (SFOV)*

8.2.3 PathSim IRVE

Based on the evidence compiled in this research program, we may suggest further improvements to the PathSim IRVE interface. To review, PathSim v0.2 uses both Object and Viewport Layout Spaces (Figures 5.7 – 5.10). For Object Space layouts of Macro-scale tonsil and connective tissue, the RelativeRotation technique is used with a Line connector. In the Micro-scale visualization, FixedPosition layout technique is used (Figures 5.8 and 5.10 respectively). PathSim's Viewport Space layouts represent multi-scale information, and so are not visually associated to their referents. They are also stable in terms of their location in the visual field across scales.

Because the end users of PathSim know their anatomy quite well, most of the activity in PathSim has to do with abstract information targets (*->A), i.e. the comparison of spatially-registered abstract information (S->A). For these reasons, we propose to keep the Line connector. The requirement of Visibility also seems well served for the Waldeyer's ring through the use of the RelativePosition Semantic Object Layout. While Periodic and Continuous Scaling were comparable for the most part, Continuous Scaling did provide consistent performance across the majority of tasks. Therefore we recommend changing the scaling technique from Periodic to Continuous.

8.3 Descriptive Models

In this section, we explore an initial interpretive framework to understand the relative power of Depth and Gestalt cues in IRVEs. This initial descriptive model is based on applying Information Theory to the problems of Human Information Processing (HIP) for IRVE perception. Through this theoretical inquiry, we may build a better framework to understand the perceptual properties and usability impacts of IRVE information design techniques.

8.3.1 Initial (naïve) Model

In IRVEs, we can consider our layout techniques as presenting information about the (referential) relation between abstract and spatial information types. Each consistent Depth and Gestalt cues between the annotation and referent reduces the ambiguity of this relation. In this first model, we apply Information Theory to the problem. Information Theory defines a unit of information (a bit) as any signal that reduces uncertainty. In this section, we building an initial explanatory model that attempts to quantify the amount of information conveyed by the visual configuration of the display technique.

Since this is an unexplored problem and we are interested in the usability impacts of various combinations of Depth and Gestalt cues in IRVE information design, we will initially assume the Null Hypothesis, which is that all bits are 'created equal'. That is, they each contribute equally some information that serves the disambiguation process (relating annotation and referent). Therefore, each cue would convey one (1) bit of information to the true/false question *"Are the annotation and referent related?"*. In addition, we assume that there is perfect transmission of the bit between sender (IRVE system) and receiver (IRVE user). These assumptions allow us to effectively create a 1-Dimensional scale that quantifies the degree of Association in the IRVE layout.

Consider the examples tested in our Experiments (Tables 6.2, 6.3, 7.1, 7.4, 7.6). To aid this exposition, we have reprinted the tables below. For each IRVE Display Technique, we can describe the amount of information it conveys about the relationship between annotation and referent by adding up the bits provided by each cue present. In order to do this, there are two special cases to consider: first, how to compute a bit value for a technique when one of the cues was varied as a dependent variable, and second, when two techniques are aggregated into one representative technique (as was done in our post-hoc analysis of Experiments 3 and 4).

In the first case, consider Experiments 3 and 4. In Experiment 3, we varied the cue of Relative Size as a three-level independent variable. In one of the three conditions, the cue was present ($P = 0.33$). By the equations summarized in [Wickens, 2000] pgs 44-50, we can compute the information conveyed by this cue (H) based on its probability. According to Information Theory, a less probable event conveys more information than a common event. By Equation 1, the Relative Size depth cue in Experiment 3 conveys 1.58 bits. In Experiment 4, the cue of Connectedness was present across all three variations (line, semi-transparent polygon, and opaque polygon connectors). For our initial Information Theory model, the cue is present in any case and therefore all conditions contribute 1 bit of information.

$$H = \log_2(1/P)$$

Equation 1: Information conveyed (H) in bits by an event with probability P

By the same logic, when we aggregate techniques that provide different cues (as in our post-hoc analysis), we consider the probability that cue was present. For the High and Low conditions we created, two techniques are averaged. If a cue is present in only one of those techniques, the probability of that cue event is (P = 0.5). By Equation 1, these cues convey 1 bit of information. If we sum the bits present in each technique tested, we have a quantitative expression of how much information an IRVE display technique conveys about the relationship of an annotation and its referent.

8.3.2 Summary of the Initial Model

Using Information Theory, we have computed Association Information Values (AIV = bits present) for all the techniques tested in this work. They are summarized in Table 8.1. All four experiments used the same basic data set; therefore they were all comparable in terms of the amount of information present in the environment. The natural question is, 'how does the descriptive model line up with the observed results?'.

First, we note in many cases a higher value is not necessarily advantageous for user performance. For example, the Viewport and Display techniques tested did enable better performance with very little Association information; the pattern was generally similar for other experiments. We reflect this general trend favoring visibility over Association in our cumulative design guidelines (see Section 8.2.2).

AIV (Bits Present)	1	2	3	5	5.58	6	6.58
Display Technique	Display (D)	Viewport (V) Semantic HUD (S)	Proximity HUD (P)	Object (O) Low (L)	ForceDirected (FD)	High (H)	ScreenBounds (SB)

Table 8.1: Sum Bits (AIV) conveying the relation between annotation and referent in the IRVE conditions tested in this research

If naïve users were equally sensitive to all cues (all bits were considered equally), we might have expected to see a positive linear relationship between more information conveyed and better performance. However this is not the case. The relationship is inverted from what we might expect assuming more information is better: the general results show that layout techniques with lower AIV values are generally more advantageous.

But, certain combinations of cues can make a higher AIV value advantageous as we have seen via the post-hoc analysis. This suggests is there are interferences or interactions among the cues that a simple summation approach cannot capture. Indeed, the richness of our results suggests that some cues are more important than others and that naïve users employ cues differently depending on the task.

If we consider our data in light of Information Theory, we can plot significant objective metrics of user performances by AIV value. Where one AIV provided two or more data points, those points were averaged. If we plot accuracy performance and add linear trend lines, we see that for accuracy, high AIVs (more bits depicting the referential relation) are better for Search tasks and for both information mappings. However, low AIVs (fewer bits depicting the referential relation) are better for Comparisons. These trends are shown in Figure 8.1.

If we look at time performance, we also see task specificity. The pattern for time is that high AIVs are faster overall and for Comparison tasks. Low AIVs, in contrast, are faster for Search tasks and for A->S information mappings. These trends are shown in Figure 8.2. The compiled data for this analysis is shown in Table 8. 2.

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	**O**	**O**	**O**		
Motion Parallax	**O**	**O**	**O**		
Relative/Size / Perspective					
None		**V**	**V**		

Table 6.2: Depth and Gestalt Cues presented by Object (O; AIV=5) and Viewport (V; AIV =2) Space layouts used in Experiment 1

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	**O**	**O**	**O**		
Motion Parallax	**O**	**O**	**O**		
Relative/Size / Perspective					
None				**D**	

Table 6.3: Depth and Gestalt Cues presented by Object (O; AIV =5) and Display (D; AIV =1) Space layouts used in Experiment 2

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	**SB**	**SB**	**SB**		
Motion Parallax	**SB F**	**SB F**	**SB F**		
Relative/Size / Perspective	***SB F***	***SB F***	***SB F***		
None					

Table 7.1: Range of Depth and Gestalt Cues presented by Object ScreenBounds (SB; AIV=6.58) and ForceDirected (F; AIV =5.58) Space layouts used in Experiment 3; italics denotes the secondary independent variable

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion					
Motion Parallax					
Relative/Size / Perspective					
None	**P**	***S P***	**S P**		

Table 7.4: Depth and Gestalt Cues presented by Semantic (S; AIV =2) and Proximity (P; AIV =3) HUD techniques used in Experiment 4; italics denotes the secondary independent variable

	Proximity	Connectedness	Common Fate	Similarity	None
Occlusion	***H***	***H***	***H***		
Motion Parallax	***H L***	***H L***	***H L***		
Relative/Size / Perspective	***H L***	***H L***	***H L***		
None	***H***	***H L***	***H L***		

Table 7.6: Depth and Gestalt Cues presented by the aggregated High (H; AIV =6) and Low (L; AIV =5) Association techniques in the post-hoc analysis of Experiments 3 and 4; italics denotes a cue whose effect is diluted by averaging (SB & P), and (F & S).

Gestalt x Depth cues provided by IRVE display techniques tested in this research program

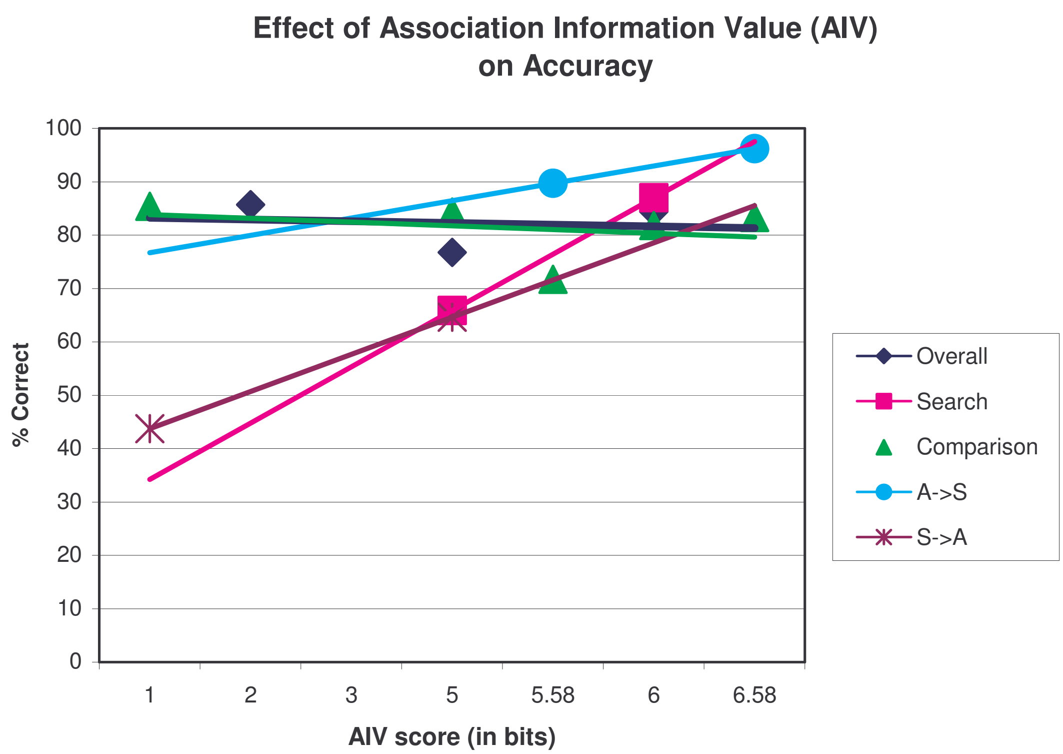


Figure 8.1: Significant performances by AIV (accuracy)

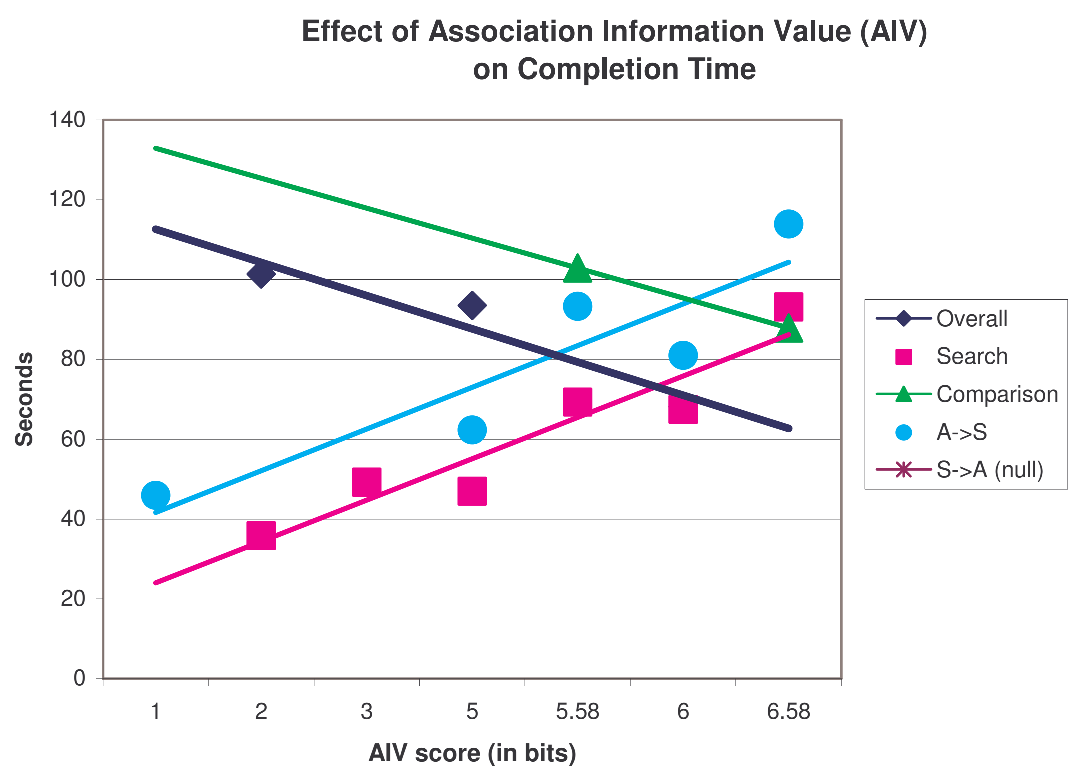


Figure 8.2: Significant performances by AIV (time)

AIV	1	2	3	5	5.58	6	6.58
Accuracy (% Correct)							
Overall		85.7		76.75		84.4	
Search				65.9		87	
Comparison	85.4			84.3	71.7	81.8	83.3
A->S					89.7		96.2
S->A	43.7			64.6			
Time (seconds)							
Overall		101.4		93.55		68.12	
Search		35.84	49.232	47	69.3	67.4	93.1
Comparison					102.9		87.9
A->S	45.9			62.3	93.3	81	113.9
S->A (null)							

Table 8.2: Averaged data of significant performances by AIV value

These observations yield two important points. First, that the Null Hypothesis we posed is not supported: users seem to be more sensitive to some cues than to others. Second, that the relative importance of cues is determined by the task (both task-type and information mapping). These facts lead us to reflect on our initial model.

8.3.3 Speculations on Revised Models

Weighted, Additive Cues

While our simple additive model of perceptual cues (Association Information Value, AIV) does allow us to quantify the Association information conveyed by an IRVE layout, it falls short in many important respects. First, in Information Theory, there is no distinction for signals or bits of different 'power'. This is problematic because the bits do not have any intrinsic strength. Therefore, this model cannot represent any differences between the information transmitted by our line connectors or our polygonal connectors for example.

Our results show that all information presented by an IRVE layout is not considered equally by the user: different cues seem more important for particular tasks. Our work expands on previous research into the power of 2D or 3D cues by considering the combination of 2D and 3D cues that is typical of IRVEs. First, consider that prior work in 2D stimuli has shown that Gestalt cues vary in their relative power for grouping. While Ware has claimed that Connectedness and Proximity are the strongest Gestalt cues for static images [Ware, 2000], we found that in IRVEs the contribution of these bits is not as significant as we might have believed. Consider that in some display contexts such as desktop displays, Connectedness has a stronger performance effect than Proximity. In addition, Common Fate (e.g. brushing and linking) seems sufficient to convey Association information about annotations and referents.

Second, consider that work in 3D stimuli has shown that Depth cues also might be ranked by relative power. Cutting & Vishton [Cutting, 1995] showed that Occlusion is consistently the strongest cue over multiple depths of field. For IRVEs, we found that Occlusion aids spatial comparison tasks, but it is generally detrimental to user performance. Strong Connectedness can be advantageous for Search tasks but detrimental for Comparison tasks. Overall, visibility seems to be the most important IRVE design criteria.

Based on the pattern of results reported here, we expect that the profile of advantageous cue weights will be specific to the task and the information mapping. Consider some of our specific results: Occlusion can aid spatial discrimination and strong Connectedness can be advantageous for Search tasks but detrimental for Comparison tasks. Full exploration of this quantitative model and the determination of weights is an area for future experimentation and analysis.

Thus, our results support the weighted, additive-cue model of Bruno and Cutting [Bruno, 1988]. This model claims that perceptual cues do not contribute equally to decision-making, but rather in a weighted fashion. Based on the pattern of the results reported here, we expect that the profile of advantageous cue weights will be specific to the task and the information mapping. In order to capture users' relative sensitivity to different cues, we suggest introducing a weight term for each cue (i) per task (t) and per display context (d) into the AIV scoring method as shown in Equation 2.

$$\mathbf{AIV} = \sum (\mathbf{weight}_{i,\,t,\,d} * \mathbf{H}_i)$$

Equation 2: Proposed weighting term reflecting user's differential sensitivity to Depth and Gestalt cues in IRVEs

Alternative Models

Indeed, there are other possibilities for capturing the richness of observed effects besides introducing a set of weights. For example if we kept in the lineage of Information Theory, we might distinguish our cue combinations through other sign systems. For example, we might devise another coding scheme that does allow multiple states for a given cue. In this way, we could describe Association in a relative ordinal scale rather than an absolute scale. For example, it may be fruitful to continue this analysis using the metric of Hamming Distance and Hamming Weights [Hamming, 1950] to describe the degree of difference between various layouts techniques by the cues presented. In this way, we may be able to better understand the more subtle impacts different cues may have depending on their power. It is an open challenge to find a sufficient explanatory model. We believe this is a worthwhile effort as developing such a model would be valuable for IRVE designers to have some tools to assess novel designs.

Most significantly, we have recast our IRVE information-association problem as an optimization problem for information throughput. The optimization problem can be stated as: "Optimize Depth and Gestalt cue combinations for IRVE task and display context". This research has shown that users leverage cues differently depending on the task, information mapping, and display context. The strengths and effects of specific cues are described in our design guidelines (Section 8.2.2).

Finally, we note that if IRVE information design is considered an optimization problem, it may be possible to develop a model based on soft constraints rather than an elaborate set of weights. In our case, the constraint satisfaction problem is: "Given this display context and task, provide a combination of perceptual cues with the least performance cost". Soft constraints are the cost functions used to evaluate solutions in a constraint satisfaction problem. Soft constraints manifest a *preference* toward some cost goal rather than a hard goal that must be satisfied. Such a model could be flexible enough to describe and deliver IRVE layouts reliably across displays and task situations.

These speculations on how our naïve model might be improved are initial signposts on the way to a fuller understanding if IRVE information design tradeoffs. The purpose of the initial (naive) model was to collect all the data from this research program into a larger framework. While the naïve model we proposed above is insufficient in a number of respects, it has helped to begin the discussion about how IRVE layout techniques can increase the throughput of information to the user. We discuss the many opportunities for future IRVE research in the next section.

8.4 Future Work

This research has not only clarified and supplemented prior work, it has also opened up new directions for Cognitive Science and HCI. The role of perceptual feature binding and Visual Working Memory in comprehension is a growing area of research that directly impacts information analysis and communication across many fields and domains. This research and the further studies it suggests will advance our understanding of how rich perceptual media may help alleviate human capacity, translation, and retrieval limitations through new techniques of visual rendering.

With our design guidelines in hand, next-generation IRVE applications can take advantage of various screen resolutions and sizes to deliver effective visualization tools to researchers, practitioners, and students. Research and development of IRVEs will continue to have an impact across industries and domains. The volume and complexity of heterogeneous data continues to grow, and scientists, engineers, and designers will continue to require better analytic and visualization tools to manage it in a useful way.

Fertile directions of basic IRVE research for the future will be further exploration of the perceptual and cognitive impacts of IRVE interface designs across desktop, large-scale, and immersive displays. This involves continued iteration of designs and experimentation through the methods of Usability Engineering, specifically toward multi-modal, embodied, and 3D user interfaces. The rich variations in IRVE display techniques raise additional questions.

We have provided an initial framework through which to consider IRVE information design. In addition, we have demonstrated interesting relationships in the design space and their relation to users performance. Still, there is much research to be done regarding how to quantify the effectiveness of perceptual cues in IRVE information design. The first problem is further investigating quantitative models of cue effectiveness that can include flexible reliance on different cues, depending on the task, information mapping, display, and content context.

For example, we have just scratched the surface of how the IRVE layouts and user performance are related across: display sizes, resolutions, and SFOV. Nearly all of our subjects were naïve to the domain of cellular biology and it remains to be determined how expert strategies might differ. In our pilot studies, we observed some intruiging relationships for between SFOV and DFOV but did not have enough statistical power to make any conclusions. Also, temporal and visual tolerances for dynamic layouts and vection is a research worthy of further attention.

Beyond fundamental perceptual cues, there are additional usability questions for IRVE display techniques. For example, all of the IRVEs tested had the same number of targets and distractors, but how effectively do these various techniques scale for larger data sets? What about other annotations beyond text and numbers? The myriad combinations of aggregation and representation possible are yet to be explored and hold great promise to increase insight and productivity in rich integrated information spaces.

Finally, in this experimental series, we collected the scores for factor-referenced cognitive aptitudes such as Perceptual Speed and Closure Flexibility. We did not have any concrete hypotheses regarding these measures, only a hunch they would be interesting. However the results are not convincing in any direction. What we can say about this lack of strong effect is that we doubt these tests are measuring factors that are predictive of performance on dynamic IRVE displays. It might be more productive to examine more recent tests, for example those for spatial ability in dynamic perceptual situations (i.e. [Bradshaw, 2003]).

However, it is crucial not only to improve our understanding of design principles and user abilities, but also to make them practical. For this reason, future research should also include application development with researchers in other domains. There are a number of specific applications where the benefit of IRVEs can be seen. For example, in the fields of biology and medicine, scientists examine the properties and relationships of structures, from cells to tissues to gross anatomy. Similarly, in chemistry, astronomy or architecture, understanding the spatial nature of processes is crucial for insight - using IRVEs can reduce the cognitive distance between the investigator and their data. Additionally, the principles and techniques of IRVEs could be beneficially applied to educational spaces, as in the multimedia software and curricula that train and educate new scientists and practitioners.

Collaborations and development with medical and biochemical experts will be especially fruitful for IRVEs. In these domains it is especially important to unify spatial and abstract information. These domains also have rich semantics and additional requirements such as accuracy over speed. Such multi-disciplinary collaboration will lead to next-generation information tools that further: leverage XML for data interchange, provide web-services to high-performance computing systems, integrate IRVE assets with the ontologies of the Semantic Web, and push the visualization and interface capabilities of open standards such as X3D.

Lastly, IRVE researchers should continue to track and contribute to the VE and InfoVis toolkits in the open-source and open-standards software movements; these technologies provide a powerful means to develop and deploy new tools with robust functionality and low financial cost. Continuing IRVE research in the context of Web-connected 3D graphics will provide increased interoperability and re-use. In this way, progress for 3D user interfaces for IRVE might begin to significantly increase and improve in much the same way as 2D UI information-rich hypermedia interfaces began rapid & ongoing evolution with advent of the HTML/XML Web. The future holds great promise for the development and deployment of IRVE display techniques and components for portable, integrated information spaces.

9. References

Ahlberg, C., and Wistrand, E. (1995). IVEE: an Information Visualization and Exploration Environment. IEEE InfoVis, (Spotfire): www.spotfire.com.

Ames, A. L., Nadeau, David R, Moreland, John L (1997). VRML Sourcebook. New York, John Wiley & Sons.

Anderson, J. R. (1983). The Architecture of Cognition. Cambridge, MA, Harvard University Press.

Baddeley, A. (2003). "Working memory: Looking back and looking forward." Nature Reviews Neuroscience **4**: 829-839.

Baddeley, A., and Lieberman, K. (1980). Spatial working memory. Attention & Performance. R. S. Nickerson. Hillsdale, NJ, Lawrence Erlbaum Associates Inc. **VIII**.

Baddeley, A. and R. Logie (1999). Working Memory: the multiple component model. Models of working memory: mechanisms of active maintenance and executive control. A. M. P. Shah. New York, Cambridge University Press**:** 28-61.

Baddeley, A. D., and Hitch, G. (1974). Working memory. Recent advances in learning and motivation. G. Bower. New York, Academic Press. **8**.

Baldonado, M., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for using Multiple Views in Information Visualization. Advanced Visual Interfaces (AVI).

Bederson, B. B., Hollan, J.D., Perlin, K., Meyer, J., David, B., and Furnas, G (1996). "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics." Journal of Visual Languages and Computing **7**(1): 3-32.

Bederson, B. B., Meyer, J., and Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. ACM Symposium on User Interface Software and Technology.

Bell, B., Feiner, S., and Hollerer, T. (2001). View Management for Virtual and Augmented Reality. ACM Symposium on User Interface Software and Technology.

Bell, B. A., and Feiner, S.K. (2000). Dynamic Space Management for User Interfaces. ACM Symposium on User Interface Software and Technology.

Bertin, J. (1983). Semiology of Graphics. Madison, WI, University of Wisconsin Press.

Biederman, I. (1987). "Recognition by components: A theory of human-image understanding." Psychological Review(94): 115-147.

Biederman, I., and Gerhardstein, P.C. (1993). "Recognizing depth-rotated objects: Evidence for 3-D viewpoint invariance." Journal of Experimental Psychology: Human Perception and Performance(19): 1162-1182.

Bogart, B., Nachbar, Martin, Kirov, Miro, McNeil, Dean (2001). VR Anatomy Courseware. NYU School of Medicine, NYU School of Medicine, http://endeavor.med.nyu.edu/courses/anatomy/courseware/vranat/

Bolter, J., Hodges, L.F., Meyer, T., & Nichols, A. (1995). "Integrating Perceptual and Symbolic Information in VR." IEEE Computer Graphics and Applications **15**(4): 8-11.

Boukhelifa, N., Roberts, Jonathan C., Rodgers, Peter (2003). A Coordination Model for Exploratory Multi-View Visualization. International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2003), IEEE.

Bowman, D., D. Johnson, and L. Hodges (2001b). "Testbed Evaluation of VE Interaction Techniques." Presence: Teleoperators and Virtual Environments **10**(1): 75-95.

Bowman, D., Hodges, L., and Bolter, J. (1998). "The Virtual Venue: User-Computer Interaction in Information-Rich Virtual Environments." Presence: Teleoperators and Virtual Environments **7**(5): 478-493.
Bowman, D., J. Gabbard, and D. Hix. (2002). "A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods." Presence: Teleoperators and Virtual Environments **11**(4).
Bowman, D., D. Koller, et al. (1997b). Travel in Immersive Virtual Environments: an Evaluation of Viewpoint Motion Control Techniques. Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'97), IEEE Press.
Bowman, D., Kruijff, E., Joseph, J., LaViola., Poupyrev, I. (2001a). "An introduction to 3-D User
Interface Design." Presence: Teleoperators and Virtual Environments **10**(1): 96-108.
Bowman, D., Kruijff, E., LaViola, J., and Poupyrev, I. (2004). 3D User Interfaces: Theory and Practice. Boston, Addison-Wesley.
Bowman, D., North, C., Chen, J., Polys, N., Pyla, P., and Yilmaz, U. (2003a). Information-Rich Virtual Environments: Theory, Tools, and Research Agenda. Proceedings of ACM Virtual Reality Software and Technology, Osaka, Japan, ACM SIGGRAPH.
Bowman, D., Setareh, M., Pinho, M., Ali, N., Kalita, A., Lee, Y., Lucas, J., Gracey, M., Kothapalli, M., Zhu, Q., Datey, A., and Tumati, P. (2003b). "Virtual-SAP: An Immersive Tool for Visualizing the Response of Building Structures to Environmental Conditions." Proceedings of IEEE Virtual Reality: 243-250.
Bowman, D. W., J., Hodges, L., and Allison, D. (1999). "The Educational Value of an Information-Rich Virtual Environment." Presence: Teleoperators and Virtual Environments **8**(3): 317-331.
Bradshaw, G., and Giesen, J. M. (2003). Dynamic Measures of Spatial Ability, Executive Function, and Social Intelligence, Storming Media: http://www.stormingmedia.us/40/4074/A407414.html.
Brown, M. (2002). XML Processing with Perl, Python, and PHP. San Francisco, Sybex.
Bruno, N., & Cutting, J.E. (1988). "Minimodularity and the perception of layout." Journal of Experimental Psychology: General **177**: 161-170.
Brutzman, D. (2002). Teaching 3D modeling and simulation: virtual kelp forest case study. Proceedings of Web3D, Tempe, AZ, ACM SIGGRAPH.
Card, S., J. Mackinlay, and B. Shneiderman (1999). Information Visualization: Using Vision to Think. San Francisco, Morgan Kaufmann.
Card, S. K., Moran, T.P., and Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ, Lawrence Erlbaum Assoc.
Carroll, J. B. (1993). Human Cognitive Abilities: a survey of factor-analytic studies. New York, NY, Cambridge University Press.
Celada, F. S., Philip (1992). "A model of cellular interactions in the immune system." Immunology Today **13**(2): 56-62.
Chandler, P., and Sweller, J. (1991). "Cognitive Load Theory and the Format of Instruction." Cognition and Instruction **8**: 293-332.
Chen, J., Narayan, M.A., and Perez-Quinones, M.A. (2005). The Use of Hand-held Devices for Search Tasks in Virtual Environments. IEEE VR2005 workshop on New Directions in 3DUI, Germany, IEEE Press.

Chen, J., P. Pyla, and D. Bowman (2004). Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment. Virtual Reality, Chicago, IL, IEEE.

Collete, F., and Van der Linden, M. (2002). "Brain imaging of the central executive component of working memory." Neuroscience and Biobehavior Reviews(26): 105-25.

Convertino, G., Chen, J., Yost, B., Young-Sam, Ryu, and North, C. (2003). Exploring context switching and cognition in dual-view coordinated visualizations. International Conference on Coordinated and Multiple Views in Exploratory Visualization.

Conway, A. R. A., Kane, M.J., Engle, R.W. (2003). "Working memory capacity and its relation to general intelligence." Trends in Cognitive Sciences **7**(12): 547-52.

Cutting, J. E., & Vishton, P.M. (1995). Perceiving layout: The integration, relative dominance, and contextual use of different information about depth. Handbook of Perception and Cognition. W. Epstein, & S. Rogers. NY, Academic Press. **Vol. 5: Perception of Space and Motion**.

Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G., and Starkweather, G. (2003). Toward characterizing the productivity benefits of very large displays. INTERACT.

Dachselt, R., Hinz, M., Meissner, K , (2002). CONTIGRA: An XML-Based Architecture for Component-Oriented 3D Applications. Proceedings of the Web3D 2002 Symposium, ACM SIGGRAPH

Dachselt, R. and E. Rukzio (2003). Behavior3D: an XML-based framework for 3D graphics behavior. Proc. of the Web3D 2003 Symposium, ACM SIGGRAPH.

Darken, R. P., and Peterson, Barry (2002). Spatial Orientation, Wayfinding, and Representation. Handbook of Virtual Environments. K. Stanney, Lawrence Erlbaum**:** 493-518.

Darken, R. P., and Silbert, John L. (1996). Wayfinding Strategies and Behaviors in Large Scale Virtual Worlds. CHI, ACM.

Diaper, D. (1989). Task analysis for knowledge-based descriptions (TAKD); The method and an example. In Task Analysis for Human-Computer Interaction. D. Diaper. Chichester, England, Ellis-Horwood**:** 108-59.

Domos, S. (2004). "4D Builder Suite." http://www.domos.be/site/4D/_eng/index_eng.htm.

Dos Santos, C., Gros, P, Abel, P, Loisel, D, Trichaud, N, and Paris, JP (2000). Mapping Information onto 3D Virtual Worlds. Proceedings of IEEE International Conference on Information Visualization, London, England.

Doubleday, A., Ryan, Michele, Springett, Mark, Sutcliffe, Alistair (1997). A comparison of usability techniques for evaluating design. DIS, Amsterdam, NL, ACM.

Draper, M. H., Viirre, E.S., Furness, T.A. & Gawron, V.J. (2001). "Effects of image scale and system time delay on simulator sicjness within head-coupled virtual environments." Human Factors **43**(1): 129-146.

Dykstra, P. (1994). "X11 in Virtual Environments: Combining Computer Interaction Methodologies." j-X-RESOURCE **9**(1): 195-204.

Ekstrom, R. B., French, J.W., Harman, H.H. (1976). Manual for Kit of Factor Referenced Cognitive Tests. Princeton, NJ, Educational Testing Service.

Ericsson, K. A. (2003). "Exceptional memorizers: made, not born." Trends in Cognitive Sciences **7**(6): 233-235.

Ericsson, K. A., and Kintsch (1995). "Long Term Working Memory." Psychological Review **102**(2): 211-45.

Eysenck, M. W., and Keane, M.T. (2000). Cognitive Psychology: A student's handbook. Philadelphia, PA, Psychology Press.

Faraday, P. (1995). Evaluating Multimedia Presentations for Comprehension. CHI, Companion, Denver, CO, ACM.

Faraday, P., and Sutcliffe, Alistair (1996). An Empirical study of Attending and Comprehending Multimedia Presentations. ACM Multimedia, Boston, MA, ACM.

Faraday, P., and Sutcliffe, Alistair (1997). Designing Effective Multimedia Presentations. CHI, Atlanta, GA, ACM.

Faraday, P., and Sutcliffe, Alistair (1998). Making Contact Points between Text and Images. Multimedia, Bristol, UK, ACM.

Farah, M. J., Hammond, K.M., Levine, D.N., and Calvanio, R. (1988). "Visual and spatial mental imagery: Dissociable systems of representation." Cognitive Psychology(20): 439-462.

Farrell, E. J. and R. A. Zappulla (1989). "Three-Dimensional Data Visualization and Biomedical Applications." Critical Reviews Biomedical Engineering **16**(4): 323-326.

Feiner, S., Macintyre, B., Haupt, M., and Solomon, E. (1993). Windows on the World: 2D Windows for 3D Augmented Reality. Symposium on User Interface Software and Technology (UIST), ACM.

FMA (2004). Foundational Model of Anatomy (FMA), http://sig.biostr.washington.edu/projects/fm/index.html.

Foley, J. D., van Dam, Andries, Feiner, Steven K, Hughes, John F (1995). Computer Graphics: Principles and Practice in C. Boston, Addison-Wesley.

Frank, A., and Timpf, S. (1994). "Multiple Representations for Cartographic Objects in a Multi-Scale Tree - An Intelligent Graphical Zoom." Computers & Graphics **18**(6): 823-829.

Friedhoff, R., and Peercy, Mark (2000). Visual Computing. New York, Scientific American Library.

Furnas, G., W. (1981). The FISHEYE View: A New Look at Structured Files. Murray Hill, NJ, AT&T Bell Laboratories.

Furnas, G., W. (1986). Generalized Fisheye Views: Visualizing Complex Information Spaces. CHI, ACM.

Gabbard, J., D. Hix, et al. (1999). "User-Centered Design and Evaluation of Virtual Environments." IEEE Computer Graphics & Applications **19**(6): 51-59.

Goguen, J. (2000). Information Visualizations and Semiotic Morphisms. http://citeseer.ist.psu.edu/goguen00information.html, UCSD.

Gopher, D. (1996). "Attention control: explorations of the work of an executive controller." Cognitive Brain Research **5**: 23-38.

Gray, W. D., and Salzman, Marilyn C. (1998). "Damaged Merchandise: a review of experiments that compare usability evaluation methods." Human-Computer Interaction **13**(3): 203-61.

Green, C., and Bavelier, D. (2003). "Action videogame modifies visual selection attention." Nature Reviews Neuroscience **423**: 534-537.

Grilo, A., Caetano, Artur, Rosa, Agostinho (2001). "Agent Based Artificial Immune System Genetic and Evolutionary Computation." Conference Late Breaking Papers, citeseer.nj.nec.com/442254.html **San Francisco**

Gutwin, C. a. S., A. (2003). Fisheye Views are Good for Large Steering Tasks. CHI.

Hamming, R. W. (1950). "Error-detecting and error-correcting codes." Bell System Technical Journal **29**(2): 147-160.

Harris, S. (2004). "PathSim: Scientists model interaction of viruses and immune system." Virginia Tech Research Magazine Fall, http://www.research.vt.edu/resmag/fall2004/PathSim.html

Hibbard, W., Dyer, Charles R., and Paul, Brian E. (1995). Interactivity and the Dimensionality of Data Displays. Perceptual Issues in Visualization. G. Grinstein, and Levkoitz, H. New York.

Hibbard, W., Levkowitz, H., Haswell, J.,Rheingans, P., and Schoeder, F. (1995). Interaction in Perceptually-based Visualization. Perceptual Issues in Visualization. G. Grinstein, and Levkoitz, H. New York, Springer.

Hochheiser, H., Shneiderman, B. (2004). "Dynamic Query Tools for Time Series Data Sets, Timebox Widgets for Interactive Exploration." Information Visualization, Palgrave-Macmillan **3**(1): 1-18.

Hoellerer, T., S. Feiner, et al. (1999). "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System." Computers and Graphics **23**(6): 779-785.

Irani, P., and Ware, C. (2000). Diagrams based on structural object perception. Advanced Visual Interfaces, Palermo.

Just, M. A., Carpenter, P.A., Keller, T.A. (1996). "The Capacity Theory of Comprehension: New Frontiers of Evidence and Arguments." Psychological Review **103**(4): 773-80.

Kaptelinin, V., and Nardi, B. (1997). The Activity Checklist, Report. D. o. Informatics. Sweden, Umeå University.

Kay, M. (2001). XSLT. Birmingham UK, Wrox Press.

Keller, P. R. (1993). Visual Cues: Practical Data Visualization. Piscataway, NJ, IEEE Computer Society Press.

Kim, T. a. F., Paul (2002). A 3D XML-Based Customized Framework for Dynamic Models. Proceedings of the Web3D 2002 Symposium, ACM SIGGRAPH

Kling-Petersen, T., R. Pascher, et al. (1999). "Virtual Reality on the Web: The Potential of Different Methodologies and Visualization Techniques for Scientific Research and Medical Education." Stud Health Technol Inform **62**: 181-186.

Kossyln, S. M. (1994). Image and Brain: The Resolution of the Imagery Debate. Cambridge, Mass, MIT Press.

Larimer, D. a. B., D. (2003). VEWL: A Framework for Building a Windowing Interface in a Virtual Environment. Proceedings of INTERACT: IFIP TC13 International Conference on Human-Computer Interaction.

Larkin, J. H., and Simon, Herbert A. (1987). "Why a Diagram is (Sometimes) Worth Ten Thousand Words." Cognitive Science **11**: 65-99.

Logie, R. H. (1995). Visuo-spatial working memory. Hove, UK, Psychology Press.

Lohse, J. (1991). A Cognitive Model for the Perception and Understanding of Graphs. Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, New Orleans, LA, ACM.

Macedonia, M. R., and Zyda, Michael J. (1997). "A Taxonomy for Networked Virtual Environments." IEEE MultiMedia: 48-56.

Macguire, E. A., Valentine, Wilding, Kapurs (2003). "Routes to remembering: the brains behind superior memory." Nature Neuroscience(6): 90-95.

Mackinlay, J. (1986). "Automating the design of graphical presentations of relational information." ACM Trans. Graph. **5**(2): 110-141.

Mackinlay, J., and Heer, J. (2004). Wideband Displays: Mitigating Multiple Monitor Seams. CHI, Vienna, Austria, ACM.
Marr, D. (1982). Vision : a computational investigation into the human representation and processing of visual information. San Francisco, W.H. Freeman.
Mayer, R. E. (2002). "Cognitive Theory and the Design of Multimedia Instruction: An Example of the Two-Way Street Between Cognition and Instruction." New Directions for Teaching and Learning **89**: 55-71.
McClean, P., Saini-Eidukat, Bernie, Schwert, Donald. Slator, Brian, and White, Alan (2001). Virtual Worlds in Large Enrollment Biology and Geology Classes Significantly Improve Authentic Learning. Selected Papers from the 12th International Conference on College Teaching and Learning (ICCTL-01). J. A. Chambers. Jacksonville, FL, Center for the Advancement of Teaching and Learning**:** 111-118.
McLaughlin, B. (2001). Java & XML. Cambridge, O'Reilly.
Miller, G. A. (1956). "The magic number seven, plus or minus two: Some limits on our capacity to process information." Psychological Review **63**: 81-93.
Miyake, A., Friedman, N.P., Rettinger, D.A, Shah, P., Hegarty, M. (2001). "How are Visuospatial Working Memory, Executive Functioning, and Spatial Abilities Related? A Latent-Variable Analysis." Journal of Experimental Psychology: General **130**(4): 621-640.
Munro, A., Breaux, Robert, Patrey, Jim, Sheldon, Beth, (2002). Cognitive Aspects of Virtual Environment Design. Handbook of virtual environments: design, implementation, and applications. Mahwah, N.J., Lawrence Erlbaum Associates.
Murray-Rust, P., Rzepa Henry S., and Wright, M. (2001). "Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content." New J. Chem.: 618-634.
Nardi, B. (1992). Context and Consciousness: Activity Theory and Human-Computer Interaction. Cambridge, MIT Press.
Newman, W. (1997). Better or Just Different? On the Benefits of Designing Interactive Systems in Terms of Critical Parameters. DIS '97, Amsterdam, Netherlands, ACM.
Nielsen, J. and R. Molich (1992). Heuristic Evaluation of User Interfaces. Proceedings of the 1992 ACM Conference on Human Factors in Computing Systems (CHI'92), ACM Press.
NIST Chemistry WebBook, http://webbook.nist.gov/chemistry.
Norman, D. A. (1986). Cognitive Engineering. User Centered System Design. D. A. Norman and S. D. Draper. Hillsdale, NJ, Lawrence Erlbaum Associates**:** 31-61.
North, C. (2001). Multiple Views and Tight Coupling in Visualization: A Language, Taxonomy, and System. CSREA CISST Workshop of Fundamental Issues in Visualization.
North, C., and Shneiderman, B. (2000). "Snap-Together Visualization: Can Users Construct and Operate Coordinated Views?" Intl. Journal of Human-Computer Studies **53**(5): 715-739.
North, C., Conklin, N., Idukuri, K., and Saini, V. (2002). "Visualization Schemas and a Web-based Architecture for Custom Multiple-View Visualization of Multiple-Table Databases." Information Visualization, Palgrave-Macmillan **December**.
NPS, N. P. S. (2003). SAVAGE content archive, http://web.nps.navy.mil/~brutzman/Savage/contents.html , .
Pausch, R., Snoddy, Jon, Taylor, Robert, Watson, Scott, and Haseltine, Eric (1996). Disney's Aladdin: first steps toward storytelling in virtual reality. International Conference on Computer Graphics and Interactive Techniques.

Payne, S., and Green, T. (1986). "Task-action grammars: A model of the mental representation of task languages." Human-Computer Interaction(2): 93-133.

Physiome Project, T. (2003). CellML, AnatML, FieldML, http://www.physiome.org.

Pickett, R. M., Grinstein, G., Levkowitz, H., Smith, S. (1995). Harnessing Preattentive Perceptual Processes in Visualization. Perceptual Issues in Visualization. G. Grinstein, and Levkoitz, H. New York, Springer.

Pierce, J. S., Forsberg, Andrew S., Conway, Matthew J., Hong, Seung, Zeleznik, Robert C., Mine, Mark R. (1997). Image plane interaction techniques in 3D immersive environments. Interactive 3D graphics, ACM.

Plumlee, M., and Ware, C. (2003). Integrating multiple 3d views through frame-of-reference interaction. International Conference on Coordinated and Multiple Views in Exploratory Visualization.

Polys, N., Bowman, D., North, C., Laubenbacher, R., Duca, K., (2004d). PathSim Visualizer: An Information-Rich Virtual Environment for Systems Biology. Web3D Symposium, Monterey, CA, ACM Press.

Polys, N., North, C., Bowman, D., Ray, A., Moldenhauer, M., Dandekar, C. (2004a). Snap2Diverse: Coordinating Information Visualizations and Virtual Environments. SPIE Conference on Visualization and Data Analysis (VDA), San Jose, CA.

Polys, N., L. Shupp, et al. (2006). "The Effects of Task, Task Mapping, and Layout Space on User Performance in Information-Rich Virtual Environments." Technical Report TR-06-12: http://eprints.cs.vt.edu.

Polys, N. F. (2003). Stylesheet Transformations for Interactive Visualization: Towards Web3D Chemistry Curricula. Web3D Symposium, St. Malo, France, ACM Press.

Polys, N. F. (2005b). Publishing Paradigms with X3D. Information Visualization with SVG and X3D. a. V. G. Chanomei Chen, Springer-Verlag.

Polys, N. F., and Bowman, Doug A. (2004b). "Desktop Information-Rich Virtual Environments: Challenges and Techniques." Virtual Reality **8**(1): 41-54.

Polys, N. F., Bowman, Doug A., North, Chris (2004c). Information-Rich Virtual Environments: Challenges and Outlook. NASA Virtual Iron Bird Workshop, NASA Ames, http://ic.arc.nasa.gov/vib/index.php.

Polys, N. F., Duca, K. A., Laubenbacher, R., Bowman, D. A., North, C. (2003). Interactive Visualization of Biological Databases using Information-Rich Virtual Environments. Digital Biology: The Emerging Paradigm, Silver Springs, MD, National Institute of Health.

Polys, N. F., Kim, S., Bowman, D.A. (2005c). Effects of Information Layout, Screen Size, and Field of View on User Performance in Information-Rich Virtual Environments. Proceedings of ACM Virtual Reality Software and Technology Monterey, CA, ACM SIGGRAPH.

Polys, N. F., North, C., Bowman, D. A., Laubenbacher, R., Duca, K. A. (2005a). Information-Rich Virtual Environments for Biomedicine. Computational Cell Biology, Lennox, MA.

Prince, S., Cheok, A., Farbiz, F., Williamson, T., Johnson, N., and Billinghurst, M. (2002). 3-D Live: Real Time Interaction for Mixed Reality. Computer Supported Cooperative Work, ACM.

Puzone, R., Kohler, B., Seiden, P., Celada, F. (2002). "IMMSIM, a flexible model for in machina experiments on immune system responses." Future Generation Computer Systems, Elsevier Science B.V. **18**: 961-972.

Rensink, R. A. (2000). "Seeing, sensing, and scrutinizing." Vision Research(40): 1469-1487.
Ressler, S. (2003). NIST: Open Virtual Reality Testbed Anthropometric Landmarks: http://ovrt.nist.gov/projects/vrml/h-anim/landmarkInfo.html http://ovrt.nist.gov/projects/cardlab/vrmlhead.htm.
Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics (Proceedings of SIGGRAPH), ACM.
Roberts, J. C. (1999). On Encouraging Coupled Views for Visualization Exploration. Visual Data Exploration and Analysis VI, Proceedings of SPIE, IS&T and SPIE.
Rosson, M. B., and Carroll, J. (2002). Usability Engineering: Scenario Based Development of Human-Computer Interaction. New York, NY, Morgan Kauffman.
Rothbaum, B., and Hodges, Larry (1999). "The Use of Virtual Reality Exposure in the Treatment of Anxiety Disorders." Behavior Modification **23**(4): 507-525.
Saiki, J. (2003). "Spatiotemporal characteristics of dynamic feature binding in visual working memory." Vision Research(43): 2107-2123.
Saini-Eidukat, B., Schwert, D.P., and Slator, B.M. (1999). Designing, Building, and Assessing a Virtual World for Science Education. International Conference on Computers and Their Applications, Cancun, Mexico.
Salzman, M. C., Dede, C., Loftin, B. R., and Chen, J. (1999). "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning." Presence: Teleoperators and Virtual Environments **8**(3): 293-316.
SBML (2003). Systems Biology Markup Language. http://www.sbml.org http://www.sbml.org
Shannon, C. E., and W. Weaver (1963). The Mathematical Theory of Communication, University of Illinois Press.
Sheppard, L. M. (2004). "Virtual Building for Construction Projects." IEEE Computer Graphics and Applications(January/February): 6-12.
Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. Proceedings of IEEE Visual Languages, Boulder, CO.
Simmons, D. J. (2000). "Attentional capture and inattentional blindness." Trends in Cognitive Sciences **4**: 147-155.
Smith, E. E., and Jonides, J. (1997). "Working memory: A view from neuroimaging." Cognitive Psychology(33): 5-42.
Strickland, D., Hodges, Larry, Nort, Max, and Weghorst, Suzanne (1997). "Overcoming Phobias by virtual exposure." Communications of the ACM **40**(August): 34-39.
Subramaniam, S. (2003). Scientific Data Integration: Challenges and Some Solutions. Digital Biology: The Emerging Paradigm, Silver Springs, MD, NIH.
Sutcliffe, A. (2003). Multimedia and Virtual Reality: designing multisensory user interfaces. Mahwah, NJ, Lawrence Erlbaum and Assoc.
Sutcliffe, A., and Faraday, P. (1994). Designing Presentation in Multimedia Interfaces. CHI, ACM Press.
Swaminathan, K., Sato, S. (1997). "Interaction Design for Large Displays." ACM Interactions(January).
Tan, D., Gergle, D., Scupelli, P., and Pausch, R. (2003). With Similar Visual Angles, Larger Displays Improve Performance. CHI.
Tardieu, H., and Gyselink, V. (2003). Working Memory Constraints in the Integration and Comprehension of Information in a Multimedia Context. Cognition in a Digital World. H. v. Ooostendorp. NJ, Lawrence Erlbaum & Assoc.

Thomas, J. J., and Cook, Kristin A. (2006). "A Visual Analytics Agenda." IEEE Computer Graphics & Applications(January / February): 10-13.
Töpfer, F., and Pillewizer (1966). "The Principles of Selection, A Means of Cartographic Generalization." Cartographic J **3**(1): 10-16.
Treisman, A., and Gormican, Stephen. (1988). "Feature analysis in early vision: Evidence from search asymmetries." Psychological Review **95**(1): 15-48.
Tufte, E. (1983). The Visual Display of Quantitative Information. Cheshire, CT, Graphics Press.
Tufte, E. (1990). Envisioning Information. Cheshire, Graphics Press.
Tulving, E., and Schacter, D.L. (1990). "Priming and Human Memory Systems." Science **247**(4940): 301-306.
Vogel, E. K., Woodman, G.F., and Luck, S.J. (2001). "Storage of Features, Conjunctions, and Objects in Visual Working Memory." Journal of Experimental Psychology: Human Perception and Performance **27**(1): 92-114.
W3C, W. W. W. C. "XML, XSLT."
Walsh, A., and Sévenier, Mikael (2001). Core Web3D. Upper Saddle River, NJ, Prentice-Hall.
Ware, C. (2000). Information Visualization: Perception for Design. New York, Morgan Kauffman.
Ware, C. (2003). Design as Applied Perception. HCI Models, Theories, and Frameworks: Towards a Multidisciplinary Science. J. M. Carroll. San Franscisco, Morgan-Kaufmann.
Wasserman, A. I., and Shewmake, D.T. (1982). "Rapid prototyping of interactive information systems." ACM Software Engineering Notes **7**(5): 171-80.
Watzman, S. (2003). Visual Design Principles for Usable Interfaces. The Human-Computer Interaction Handbook. J. A. Jack and A. Sears. NJ, Lawrence Erlbaum Associates, Inc.
Web3D, C. "X3D Specification, VRML Specification." ISO **http://www.web3d.org**.
Wei, B., Silva, C., Koutsofios, E., Krishnan, S., and North, S. (2000). "Visualization Research with Large Displays." IEEE Computer Graphics and Applications **20**(4): 50-54.
White, A. R., McClean, Phillip E., and Slator, Brian M. (1999). The Virtual Cell: An Interactive, Virtual Environment for Cell Biology. World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99), Seattle, WA.
White, C. (2002). Mastering XSLT. San Francisco, Sybex.
Wickens, C. D., and Hollands, J.G. (2000). Engineering psychology and human performance. Upper Saddle River, NJ, Prentice Hall.
Woodruff, A., Landay, J., and Stonebraker, M. (1998b). Goal-Directed Zoom. SIGCHI, Los Angeles, ACM.
Woodruff, A., Landay, J., and Stonebraker, M., (1998a). Constant Information Density in Zoomable Interfaces. Advanced Visual Interfaces (AVI), L'Aquila, Italy.
Yost, B. a. N., C. (2006). The Perceptual Scalability of Visualization. IEEE Symposium on Information Visualization, Baltimore Maryland, IEEE Press.
Yumetech (2005). "Xj3D." www.yumetech.com.
Zhang, J., and Norman, D.A. (1994). "Representations in distributed cognitive tasks." Cognitive Science **18**: 87-122.

Appendices

A. XML description of IRVE Display Components

The DTD and Schema that describe our IRVE display components are also included in the digital archive:

irve.dtd | **irve.xsd**

A.1 DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) -->
<!--DTD generated by XMLSPY v5 rel. 4 U (http://www.xmlspy.com)-->
<!ELEMENT BarGraph EMPTY>
<!ATTLIST BarGraph
              batch CDATA #IMPLIED
>
<!ELEMENT BoundingBoxLayout EMPTY>
<!ATTLIST BoundingBoxLayout
              bounds CDATA #IMPLIED
              xsnap CDATA #IMPLIED
              ysnap CDATA #IMPLIED
>
<!ELEMENT Connector EMPTY>
<!ATTLIST Connector
              type NMTOKEN #REQUIRED
              color CDATA #IMPLIED
>
<!ELEMENT FieldValuePanel EMPTY>
<!ATTLIST FieldValuePanel
              hasBackground CDATA #IMPLIED
              backgroundColor CDATA #IMPLIED
              backgroundTransparency CDATA #IMPLIED
              title CDATA #IMPLIED
              fields CDATA #IMPLIED
              values CDATA #IMPLIED
              batch CDATA #IMPLIED
>
<!ELEMENT FixedPositionLayout EMPTY>
<!ATTLIST FixedPositionLayout
              infoPositon CDATA #IMPLIED
>
<!ELEMENT ForceDirectedLayout EMPTY>
<!ATTLIST ForceDirectedLayout
              radial CDATA #REQUIRED
              index CDATA #REQUIRED
              indexGeo CDATA #REQUIRED
              xsnap CDATA #REQUIRED
              ysnap CDATA #REQUIRED
>
<!ELEMENT GenericHUD ANY>
<!ELEMENT HUDBorderLayout EMPTY>
<!ATTLIST HUDBorderLayout
              capacity CDATA #IMPLIED
              fillOrder CDATA #IMPLIED
```

```
>
<!ELEMENT HUDGyro EMPTY>
<!ELEMENT IRVE (HUDGyro, ViewPositionOrientation, X3DChildNode,
X3DGroupingNode, SemanticObject+)>
<!ELEMENT ImagePanel EMPTY>
<!ATTLIST ImagePanel
              url CDATA #REQUIRED
              scaleS CDATA #REQUIRED
              scaleT CDATA #REQUIRED
>
<!ELEMENT InfoLOD (#PCDATA | TextPanel | FieldValuePanel | BarGraph |
LineGraph | ImagePanel | MoviePanel | X3DChildNode | X3DGroupingNode)*>
<!ATTLIST InfoLOD
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
              range CDATA #IMPLIED
>
<!ELEMENT LayoutTechnique (FixedPositionLayout | RelativePositionLayout |
BoundingBoxLayout | ScreenBoundsLayout | ForceDirectedLayout | GenericHUD |
HUDBorderLayout)>
<!ATTLIST LayoutTechnique
              layoutSpace NMTOKEN #IMPLIED
              scaling NMTOKEN #REQUIRED
>
<!ELEMENT LineGraph ANY>
<!ELEMENT MoviePanel EMPTY>
<!ATTLIST MoviePanel
              url CDATA #IMPLIED
>
<!ELEMENT ObjLOD (#PCDATA | X3DChildNode | X3DGroupingNode)*>
<!ATTLIST ObjLOD
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
              range CDATA #IMPLIED
>
<!ELEMENT RelativePositionLayout EMPTY>
<!ATTLIST RelativePositionLayout
              infoPosition CDATA #REQUIRED
              axisOfRotation CDATA #IMPLIED
>
<!ELEMENT ScreenBoundsLayout EMPTY>
<!ATTLIST ScreenBoundsLayout
              maxBoundingBox CDATA #REQUIRED
              xsnap CDATA #REQUIRED
              ysnap CDATA #REQUIRED
>
<!ELEMENT SelectedObj (X3DGroupingNode | X3DChildNode)>
<!ATTLIST SelectedObj
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
>
<!ELEMENT SemanticObject (Connector, LayoutTechnique, InfoLOD, ObjLOD,
SelectedObj)>
<!ATTLIST SemanticObject
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
              position CDATA #IMPLIED
              rotation CDATA #IMPLIED
              load CDATA #IMPLIED
              select CDATA #IMPLIED
              hasConnector CDATA #IMPLIED
>
<!ELEMENT TextPanel EMPTY>
<!ATTLIST TextPanel
```

```
              hasBackground CDATA #IMPLIED
              backgroundColor CDATA #IMPLIED
              backgroundTransparency CDATA #IMPLIED
              textColor CDATA #IMPLIED
              message CDATA #IMPLIED
>
<!ELEMENT ViewPositionOrientation EMPTY>
<!ATTLIST ViewPositionOrientation
              enabled CDATA #IMPLIED
>
<!ELEMENT X3DChildNode EMPTY>
<!ATTLIST X3DChildNode
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
>
<!ELEMENT X3DGroupingNode (X3DChildNode)>
<!ATTLIST X3DGroupingNode
              DEF ID #IMPLIED
              USE IDREF #IMPLIED
>
```

A.2 Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) -->
<!--W3C Schema generated by XMLSPY v5 rel. 4 U (http://www.xmlspy.com)--
>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
              <!--
==========================================================================--
>
              <!-- Simple types are used as Field types for X3D
attributes. -->
              <!-- Special-case range restrictions on regular base types
also provided. -->
              <xs:simpleType name="SFBool">
               <xs:annotation>
                     <xs:appinfo>
SFBool is a logical type with possible values (true|false) to match the
XML boolean type.
Hint:  X3D SFBool values are lower case (true|false) in order to
maintain
compatibility with other XML documents.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFBool"/>
               </xs:annotation>
               <xs:restriction base="xs:boolean"/>
              </xs:simpleType>
              <xs:simpleType name="MFBool">
               <xs:annotation>
                     <xs:appinfo>
MFBool is an array of Boolean values.
Type MFBool was previously undefined in the VRML 97 Specification, but
nevertheless needed for event utilities and scripting.
Example use: MFBool is useful for defining a series of behavior states
using a BooleanSequencer prototype.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFBool"/>
               </xs:annotation>
               <xs:list itemType="xs:boolean"/>
              </xs:simpleType>
              <xs:simpleType name="SFColor">
               <xs:annotation>
                     <xs:appinfo/>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFColor"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-
9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+))))?"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="MFColor">
               <xs:annotation>
                     <xs:appinfo>Array values are optionally separated
by commas.</xs:appinfo>
```

```
                        <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFColor"/>
                 </xs:annotation>
                 <xs:restriction base="xs:string">
                        <xs:whiteSpace value="collapse"/>
                        <xs:pattern value="((((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-
9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))( )?(,)?( )?)*"/>
                 </xs:restriction>
                </xs:simpleType>
                <xs:simpleType name="SFColorRGBA">
                 <xs:annotation>
                        <xs:appinfo/>
                        <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFColorRGBA"/>
                 </xs:annotation>
                 <xs:restriction base="xs:string">
                        <xs:whiteSpace value="collapse"/>
                        <xs:pattern value="((((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-
9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))))?"/>
                 </xs:restriction>
                </xs:simpleType>
                <xs:simpleType name="MFColorRGBA">
                 <xs:annotation>
                        <xs:appinfo/>
                        <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFColorRGBA"/>
                 </xs:annotation>
                 <xs:restriction base="xs:string">
                        <xs:whiteSpace value="collapse"/>
                        <xs:pattern value="((((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-
9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+))) (((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+)))( )?(,)?( )?)*"/>
                 </xs:restriction>
                </xs:simpleType>
                <xs:simpleType name="SFDouble">
                 <xs:annotation>
                        <xs:appinfo>SFDouble is a double-precision
floating-point type.  Array values are optionally separated by commas.
See GeoVRML 1.0 Recommended Practice, Section 2.3, Limitations Of
Single-Precision for rationale.</xs:appinfo>
                        <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFDouble"/>
                 </xs:annotation>
                 <xs:restriction base="xs:double"/>
                </xs:simpleType>
```

```
              <xs:simpleType name="MFDouble">
               <xs:annotation>
                     <xs:appinfo>MFDouble is an array of Double values,
i.e. a double-precision floating-point array type.
See GeoVRML 1.0 Recommended Practice, Section 2.3, Limitations Of
Single-Precision for rationale.
SFDouble/MFDouble are analagous to SFDouble/MFDouble.  Array values are
optionally separated by commas.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFDouble"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="(((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?( )?(,)?( )?)*"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFFloat">
               <xs:annotation>
                     <xs:appinfo>SFFloat is a single-precision floating-
point type.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFFloat"/>
               </xs:annotation>
               <xs:restriction base="xs:float"/>
              </xs:simpleType>
              <!--   SFFloatNonNegative and SFFloatPositive no longer
needed
              <xs:simpleType name="SFFloatNonNegative">
               <xs:annotation>
                     <xs:appinfo>SFFloat is a single-precision floating-
point type.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFFloat"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((\+)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFFloatPositive">
               <xs:annotation>
                     <xs:appinfo>SFFloat is a single-precision floating-
point type.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFFloat"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((\+)?(0\.(0)*[1-9][0-9]*|([1-
9]([0-9]*)?)?(\.[0-9]*)?((E|e)(\+|\-)?[0-9]+)?))?"/>
               </xs:restriction>
              </xs:simpleType>  -->
              <xs:simpleType name="MFFloat">
               <xs:annotation>
                     <xs:appinfo>MFFloat is an array of SFFloat values,
i.e. a single-precision floating-point array type.  Array values are
optionally separated by commas.</xs:appinfo>
```

```
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFFloat"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="(((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?( )?(,)?( )?)*"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFImage">
               <xs:annotation>
                     <xs:appinfo>The SFImage field specifies a single
uncompressed 2-dimensional pixel image.  SFImage fields contain three
integers representing the width, height and number of components in the
image, followed by widthÃ—height hexadecimal or integer values
representing the pixels in the image.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFImage"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:minLength value="4"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="MFImage">
               <xs:annotation>
                     <xs:appinfo>MFImage is an array of SFImage
values.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFImage"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:minLength value="4"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFInt32">
               <xs:annotation>
                     <xs:appinfo>An SFInt32 field specifies one 32-bit
signed integer.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFInt32"/>
               </xs:annotation>
               <xs:restriction base="xs:integer"/>
              </xs:simpleType>
              <xs:simpleType name="MFInt32">
               <xs:annotation>
                     <xs:appinfo>An MFInt32 field defines an array of
32-bit signed integers.  Array values are optionally separated by
commas.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/technicalinfo/specifications/vrml97/part1/f
ieldsRef.html#SFInt32"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(
)?(,)?( )?)*"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFRotation">
               <xs:annotation>
```

```
                    <xs:appinfo>SFRotation is an axis-angle 4-tuple,
indicating X-Y-Z direction plus angle orientation about that axis.  The
first three values specify a normalized rotation axis vector about which
the rotation takes place. (Thus the first three values must be within
the range [-1..+1] in order to represent a normalized unit vector.
Problem:  scientific notation allows leading digit.)  The fourth value
specifies the amount of right-handed rotation about that axis in
radians.</xs:appinfo>
                    <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFRotation"/>
              </xs:annotation>
              <xs:restriction base="xs:string">
                    <xs:whiteSpace value="collapse"/>
                    <xs:pattern value="((\+|\-)?(((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (\+|\-)?(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(\+|\-)?(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(\+|\-)?(0|[1-9][0-9]*)?(\.[0-9]*)?((E|e)(\+|\-)?[0-9]+)?)?"/>
              </xs:restriction>
             </xs:simpleType>
             <xs:simpleType name="MFRotation">
              <xs:annotation>
                    <xs:appinfo>MFRotation is an array of SFRotation
values.  Array values are optionally separated by commas.</xs:appinfo>
                    <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFRotation"/>
              </xs:annotation>
              <xs:restriction base="xs:string">
                    <xs:whiteSpace value="collapse"/>
                    <xs:pattern value="((\+|\-)?(((\.[0-9]+|0(\.[0-
9]*)?)((E|e)(\+|\-)?[0-9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-
9]*)((E|e)\-[0-9]+))) (\+|\-)?(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(\+|\-)?(((\.[0-9]+|0(\.[0-9]*)?)((E|e)(\+|\-)?[0-
9]+)?)|(1(\.[0]*)?((E|e)\-[0-9]+)?)|([1-9](\.[0-9]*)((E|e)\-[0-9]+)))
(\+|\-)?(0|[1-9][0-9]*)?(\.[0-9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?(
)?)*"/>
              </xs:restriction>
             </xs:simpleType>
             <xs:simpleType name="SFString">
              <xs:annotation>
                    <xs:appinfo>SFString defines a single string
encoded with the UTF-8 universal character set.</xs:appinfo>
                    <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFString"/>
              </xs:annotation>
              <xs:restriction base="xs:string"/>
             </xs:simpleType>
             <xs:simpleType name="MFString">
              <xs:annotation>
                    <xs:appinfo>MFString is an array of SFString
values, each "quoted" and separated by whitespace.  Array values are
optionally separated by commas.</xs:appinfo>
                    <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFString"/>
              </xs:annotation>
              <xs:list itemType="xs:string"/>
             </xs:simpleType>
```

```
              <xs:simpleType name="SFTime">
               <xs:annotation>
                     <xs:appinfo>The SFTime field specifies a single
time value.  Time values are specified as a double-precision floating
point number. Typically, SFTime fields represent the number of seconds
since Jan 1, 1970, 00:00:00 GMT.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFTime"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((-1(.(0)*)?)|((\+)?(0|[1-9][0-
9]*)?(\.[0-9]*)?((E|e)(\+|\-)?[0-9]+)?))?"/>
               </xs:restriction>
               <!-- base type xs:time not usable due to different
nomenclatures for time values. -->
              </xs:simpleType>
              <xs:simpleType name="MFTime">
               <xs:annotation>
                     <xs:appinfo>MFTime is an array of SFTime values.
Array values are optionally separated by commas.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFTime"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((-1(.(0)*)?|(0|[1-9][0-
9]*)?(\.[0-9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?( )?)*)?"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFVec2f">
               <xs:annotation>
                     <xs:appinfo>SFVec2f is a 2-tuple pair of SFFloat
values.
Hint: SFVec2f can be used to specify a 2D single-precision
coordinate.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec2f"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?"/>
               </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="MFVec2f">
               <xs:annotation>
                     <xs:appinfo>MFVec2f is an array of SFVec2f values.
Array values are optionally separated by commas.</xs:appinfo>
                     <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec2f"/>
               </xs:annotation>
               <xs:restriction base="xs:string">
                     <xs:whiteSpace value="collapse"/>
                     <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?( )?)*"/>
               </xs:restriction>
              </xs:simpleType>
```

```
            <xs:simpleType name="SFVec2d">
             <xs:annotation>
                  <xs:appinfo>SFVec2d is a 2-tuple pair of SFDouble
values.  Array values are optionally separated by commas.
Hint: SFVec2d can be used to specify a 2D double-precision
coordinate.</xs:appinfo>
                  <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec2d"/>
             </xs:annotation>
             <xs:restriction base="xs:string">
                  <xs:whiteSpace value="collapse"/>
                  <xs:pattern value="((\+|\-)?([0-9]+(\.[0-
9]*)?|\.[0-9]+)((E|e)(\+|\-)?[0-9]+)? (\+|\-)?([0-9]+(\.[0-9]*)?|\.[0-
9]+)((E|e)(\+|\-)?[0-9]+)?)?"/>
             </xs:restriction>
            </xs:simpleType>
            <xs:simpleType name="MFVec2d">
             <xs:annotation>
                  <xs:appinfo>MFVec2d is an array of SFVec2d values.
Array values are optionally separated by commas.</xs:appinfo>
                  <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec2d"/>
             </xs:annotation>
             <xs:restriction base="xs:string">
                  <xs:whiteSpace value="collapse"/>
                  <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?( )?)*"/>
             </xs:restriction>
            </xs:simpleType>
            <xs:simpleType name="SFVec3f">
             <xs:annotation>
                  <xs:appinfo>SFVec3f is a 3-tuple triplet of SFFloat
values.
Hint:  SFVec3f can be used to specify a 3D coordinate or a 3D scale
value.</xs:appinfo>
                  <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec3f"/>
             </xs:annotation>
             <xs:restriction base="xs:string">
                  <xs:whiteSpace value="collapse"/>
                  <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?"/>
             </xs:restriction>
            </xs:simpleType>
            <xs:simpleType name="MFVec3f">
             <xs:annotation>
                  <xs:appinfo>MFVec3f is an array of SFVec3f values.
Array values are optionally separated by commas.</xs:appinfo>
                  <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec3f"/>
             </xs:annotation>
             <xs:restriction base="xs:string">
                  <xs:whiteSpace value="collapse"/>
                  <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?( )?)*"/>
```

```
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="SFVec3d">
                <xs:annotation>
                      <xs:appinfo>SFVec3d is a 3-tuple triplet of
SFDouble values.
See GeoVRML 1.0 Recommended Practice, Section 2.3, Limitations Of
Single-Precision.
Hint:  SFVec3d can be used to specify a georeferenced 3D
coordinate.</xs:appinfo>
                      <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec3d"/>
                </xs:annotation>
                <xs:restriction base="xs:string">
                      <xs:whiteSpace value="collapse"/>
                      <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?)?"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType name="MFVec3d">
                <xs:annotation>
                      <xs:appinfo>MFVec3d is an array of SFVec3d values.
Array values are optionally separated by commas.
See GeoVRML 1.0 Recommended Practice, Section 2.3, Limitations Of
Single-Precision.
Hint:  MFVec3d can be used to specify a list of georeferenced 3D
coordinates.</xs:appinfo>
                      <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-
X3dAbstractSpecification/Part01/fieldsDef.html#SFVec3d"/>
                </xs:annotation>
                <xs:restriction base="xs:string">
                      <xs:whiteSpace value="collapse"/>
                      <xs:pattern value="((\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)? (\+|\-)?(0|[1-9][0-9]*)?(\.[0-
9]*)?((E|e)(\+|\-)?[0-9]+)?( )?(,)?( )?)*"/>
                </xs:restriction>
              </xs:simpleType>

              <!--
=========================================================================--
>

              <xs:element name="HUDBorderLayout">
                <xs:complexType>
                      <xs:attribute name="capacity" type="SFInt32"/>
                      <xs:attribute name="fillOrder" type="MFString"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="Connector">
                <xs:complexType>
                      <xs:attribute name="type" use="required">
                            <xs:simpleType>
                                  <xs:restriction base="xs:NMTOKEN">
                                        <xs:enumeration value="line"/>
                                        <xs:enumeration value="shape"/>
                                  </xs:restriction>
                            </xs:simpleType>
                      </xs:attribute>
                      <xs:attribute name="color" type="SFColor"/>
```

```
            </xs:complexType>
          </xs:element>
          <xs:element name="FieldValuePanel">
            <xs:complexType>
                  <xs:attribute name="hasBackground" type="SFBool"/>
                  <xs:attribute name="backgroundColor"
type="SFColor"/>
                  <xs:attribute name="backgroundTransparency"
type="SFFloat"/>
                  <xs:attribute name="title" type="MFString"/>
                  <xs:attribute name="fields" type="MFString"/>
                  <xs:attribute name="values" type="MFFloat"/>
                  <xs:attribute name="batch" type="SFInt32"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="ForceDirectedLayout">
            <xs:complexType>
                  <xs:attribute name="radial" type="SFVec3f"
use="required"/>
                  <xs:attribute name="index" type="SFInt32"
use="required"/>
                  <xs:attribute name="indexGeo" type="SFInt32"
use="required"/>
                  <xs:attribute name="xsnap" type="SFFloat"
use="required"/>
                  <xs:attribute name="ysnap" type="SFFloat"
use="required"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="HUDGyro">
            <xs:complexType/>
          </xs:element>
          <xs:element name="IRVE">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="HUDGyro"/>
                        <xs:element ref="ViewPositionOrientation"/>
                        <xs:element ref="X3DChildNode"/>
                        <xs:element ref="X3DGroupingNode"/>
                        <xs:element ref="SemanticObject"
maxOccurs="unbounded"/>
                  </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="ImagePanel">
            <xs:complexType>
                  <xs:attribute name="url" type="MFString"
use="required"/>
                  <xs:attribute name="scaleS" type="SFFloat"
use="required"/>
                  <xs:attribute name="scaleT" type="SFFloat"
use="required"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="InfoLOD">
            <xs:complexType mixed="true">
                  <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element ref="TextPanel"/>
                        <xs:element ref="FieldValuePanel"/>
                        <xs:element ref="BarGraph"/>
                        <xs:element ref="LineGraph"/>
                        <xs:element ref="ImagePanel"/>
                        <xs:element ref="MoviePanel"/>
                        <xs:element ref="X3DChildNode"/>
```

```
                        <xs:element ref="X3DGroupingNode"/>
                  </xs:choice>
                  <xs:attribute name="DEF" type="xs:ID"/>
                  <xs:attribute name="USE" type="xs:IDREF"/>
                  <xs:attribute name="range" type="MFFloat"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="BoundingBoxLayout">
            <xs:complexType>
                  <xs:attribute name="bounds" type="MFVec3f"/>
                  <xs:attribute name="xsnap" type="SFVec3f"/>
                  <xs:attribute name="ysnap" type="SFVec3f"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="LayoutTechnique">
            <xs:complexType>
                  <xs:choice>
                        <xs:element ref="FixedPositionLayout"/>
                        <xs:element ref="RelativePositionLayout"/>
                        <xs:element ref="BoundingBoxLayout"/>
                        <xs:element ref="ScreenBoundsLayout"/>
                        <xs:element ref="ForceDirectedLayout"/>
                        <xs:element ref="GenericHUD"/>
                        <xs:element ref="HUDBorderLayout"/>
                  </xs:choice>
                  <xs:attribute name="layoutSpace">
                        <xs:simpleType>
                              <xs:restriction base="xs:NMTOKEN">
                                    <xs:enumeration value="DISPLAY"/>
                                    <xs:enumeration value="OBJECT"/>
                                    <xs:enumeration value="USER"/>
                                    <xs:enumeration value="WORLD"/>
                              </xs:restriction>
                        </xs:simpleType>
                  </xs:attribute>
                  <xs:attribute name="scaling" use="required">
                        <xs:simpleType>
                              <xs:restriction base="xs:NMTOKEN">
                                    <xs:enumeration
value="CONSTANT"/>
                                    <xs:enumeration value="NONE"/>
                                    <xs:enumeration
value="PERIODIC"/>
                              </xs:restriction>
                        </xs:simpleType>
                  </xs:attribute>
            </xs:complexType>
           </xs:element>
           <xs:element name="ObjLOD">
            <xs:complexType mixed="true">
                  <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element ref="X3DChildNode"/>
                        <xs:element ref="X3DGroupingNode"/>
                  </xs:choice>
                  <xs:attribute name="DEF" type="xs:ID"/>
                  <xs:attribute name="USE" type="xs:IDREF"/>
                  <xs:attribute name="range" type="MFFloat"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="RelativePositionLayout">
            <xs:complexType>
                  <xs:attribute name="infoPosition" type="SFVec3f"
use="required"/>
```

```
                        <xs:attribute name="axisOfRotation"
type="SFVec3f"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="ScreenBoundsLayout">
              <xs:complexType>
                        <xs:attribute name="maxBoundingBox" type="MFVec3f"
use="required"/>
                        <xs:attribute name="xsnap" type="SFFloat"
use="required"/>
                        <xs:attribute name="ysnap" type="SFFloat"
use="required"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="SemanticObject">
              <xs:complexType>
                        <xs:all>
                              <xs:element ref="Connector"/>
                              <xs:element ref="LayoutTechnique"/>
                              <xs:element ref="InfoLOD"/>
                              <xs:element ref="ObjLOD"/>
                              <xs:element ref="SelectedObj"/>
                        </xs:all>
                        <xs:attribute name="DEF" type="xs:ID"/>
                        <xs:attribute name="USE" type="xs:IDREF"/>
                        <xs:attribute name="position" type="SFVec3f"/>
                        <xs:attribute name="rotation" type="SFRotation"/>
                        <xs:attribute name="load" type="SFBool"/>
                        <xs:attribute name="select" type="SFBool"/>
                        <xs:attribute name="hasConnector" type="SFBool"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="TextPanel">
              <xs:complexType>
                        <xs:attribute name="hasBackground" type="SFBool"/>
                        <xs:attribute name="backgroundColor"
type="SFColor"/>
                        <xs:attribute name="backgroundTransparency"
type="SFFloat"/>
                        <xs:attribute name="textColor" type="SFColor"/>
                        <xs:attribute name="message" type="MFString"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="ViewPositionOrientation">
              <xs:complexType>
                        <xs:attribute name="enabled" type="SFBool"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="X3DChildNode">
              <xs:complexType>
                        <xs:attribute name="DEF" type="xs:ID"/>
                        <xs:attribute name="USE" type="xs:IDREF"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="X3DGroupingNode">
              <xs:complexType>
                        <xs:sequence>
                              <xs:element ref="X3DChildNode"/>
                        </xs:sequence>
                        <xs:attribute name="DEF" type="xs:ID"/>
                        <xs:attribute name="USE" type="xs:IDREF"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="SelectedObj">
```

```
            <xs:complexType>
                  <xs:choice>
                        <xs:element ref="X3DGroupingNode"/>
                        <xs:element ref="X3DChildNode"/>
                  </xs:choice>
                  <xs:attribute name="DEF" type="xs:ID"/>
                  <xs:attribute name="USE" type="xs:IDREF"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="LineGraph"/>
           <xs:element name="BarGraph">
            <xs:complexType>
                  <xs:attribute name="batch" type="SFInt32"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="MoviePanel">
            <xs:complexType>
                  <xs:attribute name="url" type="MFString"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="FixedPositionLayout">
            <xs:complexType>
                  <xs:attribute name="infoPositon" type="SFVec3f"/>
            </xs:complexType>
           </xs:element>
           <xs:element name="GenericHUD"/>
</xs:schema>
```

B. Experiment 1

B.1 Materials

Evaluation of Object vs. Viewport Space in Information-Rich Virtual Environments

Justification of Project

This project involves testing human performance on various tasks using Viewport Space in Information-Rich Virtual Environments (IRVEs). IRVEs combines traditional virtual environments (VEs) with visualizations of abstract information. The purpose of the project is to attempt to differentiate between two types of IRVE techniques and their appropriateness for the tasks tested. The two techniques will be viewed using a standard desktop computer and large display system. Because little has been explored in IRVEs, this research has the potential to have a major impact on the development of new IRVEs. Human subjects must be used so that we can quantify how well real users perform on the tasks involved.

Procedures

For this experiment, approximately 20 student subjects will be recruited. The only selection criteria will be near-perfect vision (corrected or uncorrected) and the ability to operate a desktop computer. Subjects of any age (over 18) or gender will be accepted into the study.

Subjects will attend one session of approximately 1 hour in length. The session will include:

1. Read and fill out user questionnaire
2. Spatial visualization test (enclosed)
3. Virtual environment training (mouse and keyboard controls)
4. Read a set of instructions for the experiment (enclosed)
5. Perform various tasks using Viewport Space (enclosed)

The specific technology used in the experiment will be on a standard desktop monitor and large display system. Subjects will be sitting, using a keyboard and mouse to do the tasks. For example, most tasks will involve navigating through a virtual cell. In all tasks, subjects will be sitting. The subjects will view the cell in the desktop monitor or large display system while using a keyboard and mouse to navigate through the environment. Their movements will consist of typing and clicking a mouse. There will be no difference from operating any other desktop computer.

Risks and Benefits

Subjects will be allowed to take breaks during the experiment and to quit the experiment if they feel dizzy or nauseous. See enclosed Informed Consent form for more information.

Confidentiality/Anonymity

Only the investigators will have access to the data. Each subject will be identified by number only on data forms and reports. When reporting data, only aggregate information and statistical analysis of the whole group of subjects will be published. See enclosed Informed Consent form for more information.

Informed Consent

Not Required.

Virginia Tech
VIRGINIA POLYTECHNIC INSTITUTE
AND STATE UNIVERSITY

Institutional Review Board

Dr. David M. Moore
IRB (Human Subjects)Chair
Assistant Vice President for Research Compliance
CVM Phase II- Duckpond Dr., Blacksburg, VA 24061-0442
Office: 540/231-4991; FAX: 540/231-6033
email: moored@vt.edu

DATE: October 19, 2004

MEMORANDUM

TO: Doug A. Bowman Computer Science 0106
Seonho Kim
Nicholas Polys

FROM: David Moore DM by TBM

SUBJECT: **IRB Exempt Approval:** "Viewport space in Information-Rich Virtual Environment Evaluation" IRB # 04-513

I have reviewed your request to the IRB for exemption for the above referenced project. I concur that the research falls within the exempt status. Approval is granted effective as of October 15, 2004.

cc: File

Experiment 1

User Questionnaire

Please help us to categorize our user population by completing the following items.

Gender (circle one): Male Female

Age: _____________

Do you wear glasses or contact lenses (circle one)?
No Glasses Contact Lenses

Occupation (if student, indicate graduate or undergraduate):

____

Major / Area of specialization (if student): _________________________________

Rate your familiarity with computers: (circle one)

•----------------------•----------------------•----------------------•----------------------•

not at all familiar — not very familiar — somewhat familiar — fairly familiar — very familiar

How often do you use computers...

...for work? (circle the best answer)

a. not at all
b. once a month
c. once a week
d. several times a week
e. daily

...for fun? (circle the best answer)

a. not at all
b. once a month
c. once a week
d. several times a week
e. daily

Have you ever used a virtual reality (VR) system? If so, please describe it (what type of display was used, what kind of application (e.g. game, architectural walk-through) was running, how did you interact with the system, etc.).

____

____

Experiment 1

Variable Key:

Layout 1 (LI) = Viewport Space (HUD)

Layout 2 (L2) = Object Space

SFOV (F1) = 60° vertical

SFOV (F2) = 100° vertical

A HUD, FOV1	a	Env 1	*	*	Where in the cell is the Pyruvic Acid molecule? (mitochondria)	S: A->S
	b	Env 2	*	*	**What molecule is just outside of the nucleolus?** (Pyruvic Acid)	S: S->A
	c	Env 3	*	*	**Which is the most northern molecule? (Caffeine)**	C: S
	d	Env 4	*	*	**Of all the molecules in the nucleolus, what is the name of the molecule that has the largest number of atoms?** (caffeine	C: A
B HUD FOV2	a	Env 5	*	*	**Where in the cell is the molecule with a molecular weight of 88.06? (lysosome)**	S: A->S
	b	Env 6	*	*	**Find the lysosome that is closest to a mitochondria. What is the boiling point of the molecule inside that lysosome? (117.9)**	S: S->A
	c	Env 7	*	*	**Where in the cell is the molecule with the highest boiling point?** (lysosome)	C: A
	d	Env 8	*	*	**Find the lysosome that is closest to a mitochondria. What is the melting point of the molecule with a ring?** (115.7)	C: S
C OBJ, FOV1	a	Env 9	*		**Where in the cell is the molecule with the chemical formula C6H10O? (Mitocondria)**	S: A->S
	b	Env 10	*		**What molecule is at the membrane? (pyruvic acid)**	S: S->A
	c	Env 11	*		**Where in the cell is the molecule with the lowest melting point?** (cytoplasm -40)	C: A
	d	Env 12	*		**Of all the molecules in the mitochondria what is the name of the molecule with the most carbons?** (vitamin k)	C: S
D OBJ FOV2	a	Env 13	*		**Where in the cell is the molecule with a molecular weight of 150.09?** (Nucleus)	S: A->S
	b	Env 14	*		**Find the lysosome which not near a mitochondria. What is the density of the molecule inside this lysosome? (.965)**	S: S->A
	c	Env 15	*		**Which is the most northern molecule in the nucleolus? (3 methyl indene)**	C: S
	d	Env 16	*		**Of all the molecules in the lysosomes, what is the name of the molecule that is the most south?** (pyruvic acid)	C: S

B.2 Results

Experiment 1									
Subjects									
User	**Gender**	**Age**	**Vision Aids**	**Occupation**	**Major**	**Computer Familiarity**	**Comp. Usage work)**	**Comp. Usage (fun)**	**VR Experience**
S01	M	24	Glass	CS graduate student	CS	5	5	5	yes
S02	M	22	Glass	CS graduate student	CS	5	5	5	yes, games, glove
S03	F	31	Glass	CS graduate student	CS	5	5	5	yes
S04	F	24	Glass	CS graduate student	CS	5	5	5	yes - game
S05	M	21	No	CS graduate student	CS	5	5	5	yes - archit; hmd
S06	M	22	No	CS graduate student	CS	5	5	4	no
S07	M	31	Glass	CS graduate student	CS / PL	5	5	5	no
S08	F	29	Glass	CS graduate student	CS	5	5	5	yes
S09	F	22	Lense	CS graduate student	CS	5	5	4	no
S10	M	29	No	CS graduate student	CS	4	5	5	yes all
S11	M	33	Glass	CS graduate student	CS	5	5	5	yes, hmd
S12	M	35	Glass	CS graduate student	CS / VLSI	5	5	5	yes
S13	M	22	No	CS graduate student	CS	5	5	5	yes, cave
S14	M	32	No	CS graduate student	CS	5	5	5	yes
S15	M	24	Lense	EE graduate student	EE	5	5	5	yes
S16	F	26	No	CS graduate student	CS	4	5	4	no

	Experiment 1		
Cognitive Battery Scoring: `(Right-Wrong)/(n-1)`			
	Hidden Patterns	**Paper Folding**	**Cube Comparisons**
S01	136	7.75	9
S02	111	8.75	13
S03	71	6.75	6
S04	175	8.75	15
S05	30	6.75	11
S06	146	5.25	9
S07	131	4.25	9
S08	166	6.75	8
S09	115	6.75	-3
S10	34	4.5	7
S11	138	2.75	15
S12	138	3.75	8
S13	106	6.75	3
S14	107	5.5	1
S15	28	7.75	13
S16	124	6.75	2

Experiment 1																	
ACCURACY																	
	Env	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Env**	**Display**
S01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	SM
S02	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	LG
S03	1	1	0	1	1	1	1	1	1	0	1	1	1	0	0	0	SM
S04	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	LG
S05	1	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	SM
S06	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	LG
S07	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	SM
S08	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	LG
S09	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	SM
S10	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	LG
S11	1	1	0	1	1	1	1	1	1	0	1	1	1	0	0	0	SM
S12	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	LG
S13	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	SM
S14	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	LG
S15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	SM
S16	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	LG
IDEAL Time																	
IDEAL	5	3.4	3	4	3.4	4	5	2.9	3.4	5.1	4.3	2.7	3.2	3.3	3.7	3.3	
IDEAL	7	6.5	5.5	6	5.9	6.4	10.5	6.8	5.6	12.5	6.5	8.3	6.2	6.1	7.2	8.9	

TIME SUB IDEAL																
	Env 1	Env 2	Env 3	Env 4	Env 5	Env 6	Env 7	Env 8	Env 9	Env	Env	Env	Env	Env	Env	Env	Display
S01	25	8.6	58	12	111.6	21	65	22.1	113.6	42.9	295.7	67.3	41.8	11.7	43.3	116.7	SM
S02	74	115.5	32.5	22	85.1	4.6	194.5	59.2	45.4	25.5	73.5	146.7	354.8	35.9	28.8	18.1	LG
S03	47	54.6	57	171	71.6	17	227	27.1	142.6	349.9	26.7	171.3	96.8	160.7	76.3	56.7	SM
S04	403	0.1	22.5	32	130.1	5.6	241.5	34.2	190.4	262.5	417.5	48.7	127.8	40.9	88.8	18.1	LG
S05	122	161.6	22	188	48.6	130	169	308.1	51.6	30.9	215.7	59.3	87.8	17.7	25.3	48.7	SM
S06	148	114.5	64.5	29	228.1	3.6	239.5	143.2	271.4	127.5	331.5	269.7	167.8	74.9	243.8	31.1	LG
S07	200	102.6	133	141	78.6	120	226	68.1	250.6	147.9	188.7	59.3	64.8	112.7	51.3	118.7	SM
S08	378	117.5	7.5	78	206.1	137.6	57.5	22.2	164.4	87.5	573.5	143.7	252.8	25.9	101.8	395.1	LG
S09	445	41.6	92	37	140.6	104	381	56.1	206.6	57.9	241.7	141.3	46.8	48.7	73.3	76.7	SM
S10	332	42.5	122.5	65	167.1	11.6	76.5	19.2	139.4	254.5	138.5	141.7	113.8	66.9	44.8	15.1	LG
S11	63	17.6	13	9	128.6	15	1	95.1	93.6	371.9	85.7	25.3	41.8	52.7	7.3	35.7	SM
S12	127	23.5	124.5	152	194.1	192.6	267.5	74.2	241.4	196.5	260.5	58.7	226.8	32.9	200.8	76.1	LG
S13	384	74.6	13	10	59.6	32	89	11.1	27.6	29.9	259.7	99.3	191.8	33.7	35.3	22.7	SM
S14	141	11.5	2.5	5	55.1	8.6	312.5	24.2	115.4	77.5	348.5	149.7	32.8	77.9	37.8	61.1	LG
S15	84	0.6	11	1	38.6	0.1	94	26.1	25.6	30.9	136.7	75.3	74.8	145.7	24.3	108.7	SM
S16	66	9.5	0.1	40	259.1	45.6	10.5	101.2	189.4	232.5	112.5	173.7	190.8	12.9	66.8	16.1	LG

SATISFACTION																	
	Env 1	Env 2	Env 3	Env 4	Env 5	Env 6	Env 7	Env 8	Env 9	Env	Env	Env	Env	Env	Env	Env	Display
S01	7	7	6	7	5	7	4	7	5	7	3	6	5	7	7	3	SM
S02	3	4	5	6	5	6	3	6	4	3	3	5	2	5	5	4	LG
S03	4	5	7	3	3	5	3	6	4	3	3	3	4	5	6	3	SM
S04	5	5	5	6	4	6	3	6	3	2	5	5	4	6	5	4	LG
S05	3	3	6	5	6	2	5	4	4	6	3	5	5	6	7	5	SM
S06	5	3	6	6	5	5	3	3	4	4	4	4	3	6	4	5	LG
S07	3	4	4	5	5	4	6	5	3	2	4	5	5	5	4	4	SM
S08	5	4	7	7	6	6	7	7	5	6	6	6	5	6	7	6	LG
S09	5	7	6	7	5	6	3	7	3	7	4	5	4	7	6	7	SM
S10	5	6	3	6	6	7	7	7	4	1	6	3	5	5	6	7	LG
S11	2	2	3	3	5	3	2	5	4	7	6	2	3	4	3	3	SM
S12	3	7	1	4	4	2	1	6	4	1	2	7	2	5	3	3	LG
S13	5	7	7	7	7	7	7	7	7	6	5	7	7	7	7	7	SM
S14	6	7	7	7	6	7	3	6	6	5	4	6	6	5	6	6	LG
S15	1	3	5	4	3	4	2	5	2	3	2	2	2	3	3	2	SM
S16	6	7	7	6	4	6	7	4	6	5	6	5	6	5	6	7	LG

DIFFICULTY																
	Env 1	Env 2	Env 3	Env 4	Env 5	Env 6	Env 7	Env 8	Env 9	Env	Env	Env	Env	Env	Env	Env	Display
S01	1	1	2	1	3	1	3	1	3	1	5	1	3	1	1	4	SM
S02	5	4	3	2	5	2	5	2	6	7	6	3	6	3	2	5	LG
S03	2	3	1	6	2	1	2	1	7	7	1	2	2	5	4	2	SM
S04	6	1	3	1	5	1	7	3	5	6	6	3	5	4	3	2	LG
S05	3	3	1	5	3	2	4	5	3	3	5	4	4	2	2	3	SM
S06	5	4	3	2	5	2	6	5	6	6	7	5	5	2	5	3	LG
S07	3	2	4	4	3	1	2	3	5	6	5	1	1	4	4	3	SM
S08	5	4	1	2	5	3	1	1	3	5	7	3	5	2	2	5	LG
S09	3	1	2	1	2	2	5	1	3	1	5	3	3	1	2	1	SM
S10	7	1	6	2	5	1	1	1	6	7	5	7	3	4	1	1	LG
S11	3	2	3	3	5	3	2	5	4	7	6	2	3	4	3	3	SM
S12	5	1	7	4	6	6	7	2	5	6	6	1	6	1	6	6	LG
S13	4	1	1	1	1	1	2	1	4	2	5	1	4	2	1	1	SM
S14	3	1	1	1	2	1	5	2	2	4	5	2	1	1	2	2	LG
S15	3	2	3	3	4	2	3	3	3	4	6	3	3	5	5	6	SM
S16	6	2	2	2	3	3	3	4	7	2	5	5	7	2	2	4	LG

B.3 Descriptive Statistics

ACCURACY

Descriptive Statistics - Overall

	DIS	GENDER	Mean	Std. Deviation	N
L1F1	**LG**	F	.7500	.0000	3
		M	.8500	.1369	5
		Total	.8125	.1157	8
	SM	F	.8750	.1768	2
		M	.8333	.1291	6
		Total	.8438	.1294	8
	Total	F	.8000	.1118	5
		M	.8409	.1261	11
		Total	.8281	.1197	16
L1F2	**LG**	F	.7500	.0000	3
		M	.8000	.1118	5
		Total	.7813	8.839E-02	8
	SM	F	1.0000	.0000	2
		M	.9167	.1291	6
		Total	.9375	.1157	8
	Total	F	.8500	.1369	5
		M	.8636	.1306	11
		Total	.8594	.1281	16
L2F1	**LG**	F	.6667	.1443	3
		M	.7500	.1768	5
		Total	.7188	.1602	8
	SM	F	.8750	.1768	2
		M	.7917	.2923	6
		Total	.8125	.2588	8
	Total	F	.7500	.1768	5
		M	.7727	.2360	11
		Total	.7656	.2135	16
L2F2	**LG**	F	.9167	.1443	3
		M	.8000	.2092	5
		Total	.8438	.1860	8
	SM	F	.6250	.5303	2
		M	.6250	.2622	6
		Total	.6250	.2988	8
	Total	F	.8000	.3260	5
		M	.7045	.2454	11
		Total	.7344	.2657	16

TIME SUB IDEAL

Descriptive Statistics - Overall

Display		G	Mean	Std. Deviation	N
L1F1	**LG**	F	96.1833	60.2762	3
		M	87.4500	39.1800	5
		Total	90.7250	43.9963	8
	SM	F	118.1500	50.5581	2
		M	77.2750	57.5957	6
		Total	87.4938	55.6117	8
	Total	F	104.9700	50.9941	5
		M	81.9000	47.9677	11
		Total	89.1094	48.4700	16
L1F2	**LG**	F	104.2667	1.5069	3
		M	118.0500	47.8815	5
		Total	112.8813	36.9001	8
	SM	F	128.0500	59.9273	2
		M	81.5958	50.1535	6
		Total	93.2094	52.6513	8
	Total	F	113.7800	32.6902	5
		M	98.1659	50.3703	11
		Total	103.0453	45.0809	16
L2F1	**LG**	F	216.3583	34.6323	3
		M	170.6750	63.7099	5
		Total	187.8062	56.7547	8
	SM	F	167.2500	7.6014	2
		M	116.0417	35.5073	6
		Total	128.8438	38.3500	8
	Total	F	196.7150	36.5736	5
		M	140.8750	55.3896	11
		Total	158.3250	55.8265	16
L2F2	**LG**	F	111.4833	71.3882	3
		M	97.1000	38.5054	5
		Total	102.4938	48.5668	8
	SM	F	79.5000	25.6326	2
		M	63.1250	22.4338	6
		Total	67.2188	22.6009	8
	Total	F	98.6900	54.9479	5
		M	78.5682	34.0516	11
		Total	84.8563	40.8771	16

DIFFICULTY

Descriptive Statistics - Overall

	DIS	GENDER	Mean	Std. Deviation	N
L1F1	**LG**	F	2.9167	.1443	3
		M	3.3500	1.0840	5
		Total	3.1875	.8530	8
	SM	F	2.3750	.8839	2
		M	2.4583	.7813	6
		Total	2.4375	.7410	8
	Total	F	2.7000	.5420	5
		M	2.8636	.9960	11
		Total	2.8125	.8636	16
L1F2	**LG**	F	3.2500	.7500	3
		M	3.5500	1.3509	5
		Total	3.4375	1.1080	8
	SM	F	2.0000	.7071	2
		M	2.6250	.9585	6
		Total	2.4688	.9008	8
	Total	F	2.7500	.9354	5
		M	3.0455	1.1928	11
		Total	2.9531	1.0963	16
L2F1	**LG**	F	4.7500	.2500	3
		M	5.1000	1.2324	5
		Total	4.9688	.9584	8
	SM	F	3.6250	.8839	2
		M	3.7083	.8279	6
		Total	3.6875	.7763	8
	Total	F	4.3000	.7786	5
		M	4.3409	1.2159	11
		Total	4.3281	1.0713	16
L2F2	**LG**	F	3.5833	.1443	3
		M	3.2500	1.3346	5
		Total	3.3750	1.0264	8
	SM	F	2.5000	1.0607	2
		M	3.0000	.9747	6
		Total	2.8750	.9449	8
	Total	F	3.1500	.8023	5
		M	3.1136	1.0975	11
		Total	3.1250	.9874	16

SATISFACTION

Descriptive Statistics - Overall

	DIS	GENDER	Mean	Std. Deviation	N
L1F1	**LG**	F	5.8333	.6292	3
		M	5.0000	1.1040	5
		Total	5.3125	.9978	8
	SM	F	5.5000	1.0607	2
		M	4.5417	1.7278	6
		Total	4.7813	1.5780	8
	Total	F	5.7000	.7159	5
		M	4.7500	1.4274	11
		Total	5.0469	1.3045	16
L1F2	**LG**	F	5.5000	.9014	3
		M	4.9000	1.3532	5
		Total	5.1250	1.1726	8
	SM	F	4.7500	.7071	2
		M	4.8750	1.3299	6
		Total	4.8438	1.1568	8
	Total	F	5.2000	.8367	5
		M	4.8864	1.2716	11
		Total	4.9844	1.1346	16
L2F1	**LG**	F	5.0000	1.0897	3
		M	4.0000	.7289	5
		Total	4.3750	.9543	8
	SM	F	4.0000	1.0607	2
		M	4.4167	1.3934	6
		Total	4.3125	1.2589	8
	Total	F	4.6000	1.0840	5
		M	4.2273	1.1094	11
		Total	4.3438	1.0796	16
L2F2	**LG**	F	5.5833	.7217	3
		M	4.6500	1.0983	5
		Total	5.0000	1.0351	8
	SM	F	5.2500	1.0607	2
		M	4.7500	1.6733	6
		Total	4.8750	1.4880	8
	Total	F	5.4500	.7583	5
		M	4.7045	1.3730	11
		Total	4.9375	1.2400	16

C. Experiment 2

C.1 Materials

Evaluation of Object vs. Display Space in Information-Rich Virtual Environments

Justification of Project

This project involves testing human performance on various tasks using Information-Rich Virtual Environments (IRVEs) technology. IRVEs combines traditional virtual environments (VEs) with visualizations of abstract information. The purpose of the project is to attempt to differentiate between two types of IRVE techniques and their appropriateness for the tasks tested. The two techniques will be viewed using a standard desktop computer system. Because little has been explored in IRVEs, this research has the potential to have a major impact on the development of new IRVEs. Human subjects must be used so that we can quantify how well real users perform on the tasks involved.

Procedures

For this experiment, approximately 20 student subjects will be recruited. The only selection criteria will be near-perfect vision (corrected or uncorrected) and the ability to operate a desktop computer. Subjects of any age (over 18) or gender will be accepted into the study.

Subjects will attend one session of approximately 1 hour in length. The session will include:

1. Read and fill out user questionnairre
2. Virtual environment training (mouse and keyboard controls)
3. Read a set of instructions for the experiment (enclosed)
4. Perform various tasks using IRVE technology (enclosed)

The specific technology used in the experiment will be on a standard desktop monitor. Subjects will be sitting, using a keyboard and mouse to do the tasks. For example, most tasks will involve navigating through a virtual cell. In all tasks, subjects will be sitting. The subjects will view the cell in the monitor while using a keyboard and mouse to navigate through the environment. Their movements will consist of typing and clicking a mouse. There will be no difference from operating any other desktop computer.

Risks and Benefits

Subjects will be allowed to take breaks during the experiment and to quit the experiment if they feel dizzy or nauseous. See enclosed Informed Consent form for more information.

Confidentiality/Anonymity

Only the investigators will have access to the data. Each subject will be identified by number only on data forms and reports. When reporting data, only aggregate information and statistical analysis of the whole group of subjects will be published. See enclosed Informed Consent form for more information.

Informed Consent

Not Required.

Virginia Tech
VIRGINIA POLYTECHNIC INSTITUTE
AND STATE UNIVERSITY

Institutional Review Board

Dr. David M. Moore
IRB (Human Subjects)Chair
Assistant Vice President for Research Compliance
CVM Phase II- Duckpond Dr., Blacksburg, VA 24061-0442
Office: 540/231-4991; FAX: 540/231-6033
email: moored@vt.edu

DATE: October 22, 2004

MEMORANDUM

TO: Christopher L. North Computer Science 0106
Nicholas Polys

FROM: David Moore

SUBJECT: **IRB Exempt Approval:** "Information-Rich Virtual Environment Evaluation"
IRB # 04-519

I have reviewed your request to the IRB for exemption for the above referenced project. I concur that the research falls within the exempt status. Approval is granted effective as of October 22, 2004.

cc: File

Experiment Description

Thank you for participating in this experiment. Your input in this experiment has the potential to impact how virtual environments are implemented. The information gathered from this experiment will help us understand the key factors in combining traditional virtual environments (VEs) with visualizations of abstract information.

You will be asked to perform a set of tasks using a virtual environment system. These tasks consist of navigating through a 3D environment or manipulating virtual 3D objects. We are not evaluating you; rather, you are helping us to evaluate the system. All information that you help us attain will remain anonymous. The time you take to do each task and other aspects of your interaction with the system will be measured. You may be asked questions during and after the evaluation.

The session will last about one hour. You are welcome to take rest breaks as needed. One scheduled rest break will be given about half-way through the experiment. You may also terminate your participation at any time, for any reason.

You will have the opportunity to explore the program before beginning the experiment. Before each task, you will be given full instructions. It is important that you understand the instructions before beginning each task. If anything is unclear, be sure to ask before beginning the task.

The results of this study will be kept strictly confidential. The information you provide will have your name removed and only a subject number will identify your evaluation.

Experiment 2
Preliminary Questionnaire

Participant # _________

Gender (circle one): Male Female

Age: _____________

Do you wear glasses or contact lenses (circle one)? No Glasses Contact Lenses

Occupation (if student, indicate graduate or undergraduate): ___________________________________

Major / Area of specialization (if student): _________________________________

Rate your familiarity with computers: (circle one)

1	2	3	4	5
not at all familiar	not very familiar	somewhat familiar	fairly familiar	very familiar

How often do you use computers...

...for work? (circle the best answer)

a. not at all
b. once a month
c. once a week
d. several times a week
e. daily

...for fun? (circle the best answer)

a. not at all
b. once a month
c. once a week
d. several times a week
e. daily

Have you ever used a virtual reality (VR) system? If so, please describe what type of display was used, what kind of application (e.g. game, architectural walk-through) was running and how you interacted with the system (e.g. controller, head-mount display, goggles).

____

____

Experiment 2
Task Questionnaire

Participant # _________ Question # _________

Rate how difficult this task was to complete:

1	2	3	4	5	6	7
Least Difficult						Most Difficult

If you used the 3D virtual environment, rate how difficult it was to navigate:

1	2	3	4	5	6	7	N/A
Least Difficult						Most Difficult	

Rate how useful the 3D objects were in completing this task:

1	2	3	4	5	6	7
Least Useful						Most Useful

Rate how useful the 2D objects were in completing this task:

1	2	3	4	5	6	7
Least Useful						Most Useful

Rate how satisfying this interface is for completing this task:

1	2	3	4	5	6	7
Least Satisfying						Most Satisfying

General Feedback:

Experiment 2

Task key

IRVE Layout	Task Type	Information Mapping	Question
Object Space	Search	A -> S	1
			2
			3
		S -> A	4
			5
			6
	Comparison	A -> S	7
			8
			9
		S -> A	10
			11
			12
Display Space	Search	A -> S	13
			14
			15
		S -> A	16
			17
			18
	Comparison	A -> S	19
			20
			21
		S -> A	22
			23
			24

Task Questions

1. Where in the cell is the Caffeine molecule?
2. Where in the cell is the molecule with a molecular weight of 78.13?
3. Where in the cell is the molecule with the chemical formula $C_6H_{10}O$?
4. What molecule is just outside of the nucleus?
5. Find the lysosome that is closest to a mitochondria.
   What is the boiling point of the molecule inside that lysosome?
6. What molecule is inside the membrane?
7. Which is the most northern molecule, Caffeine or Vitamin K?
8. Where in the cell is the molecule with the highest boiling point?
9. Of all the molecules with the chemical formula $C_{10}H_{10}$,
   what is the general shape?
10. Of all the molecules in the nucleolus,
    What is the name of the molecule that has the largest atom in its center?
11. Find the lysosome that is closest to a mitochondria.
    What is the melting point of the molecule shaped like a triangle?
12. Of all the molecules in the cytoplasm,
    What is the name of the molecule with the most rings?
13. Where in the cell is the Alanine molecule?
14. Where in the cell is the molecule with a molecular weight of 85.15?
15. Where in the cell is the molecule with the chemical formula $C_3H_4O_3$?
16. What is the melting point of the molecule inside the nucleolus?
17. Find the lysosome which not near a mitochondria.
    What is the density of the molecule inside this lysosome?
18. What molecule is inside the membrane?
19. Which is the most northern molecule, Dimethyl Sulpoxide or Tartaric Acid?
20. Where in the cell is the molecule with the lowest melting point?
21. Of all the molecules that are Acids,
    What is the shape of the molecule with the lowest boiling point?
22. Find the lysosome closest to a mitochondria.
    Of all the molecules in the lysosome,
    What is the density of the molecule that has the most rings?
23. What is the chemical formula of the molecule located in
    the cytoplasm that is in the shape of a ring?
24. Of all the molecules in the membrane,
    What is the name of the molecule that is in the most southern region
    of the cell?

C.2 Results

Experiment 2									
User	**Gender**	**Age**	**Vision Aids**	**Occupation**	**Major**	**Computer Familiarity**	**Comp. Usage (work)**	**Comp. Usage (fun)**	**VR Experience**
S01	F	23	No	graduate	MBA	5	daily	daily	No
S02	M	22	Contacts	undergraduate	CS	5	daily	daily	No
S03	M	23	Glasses	graduate	CS	5	daily	daily	Yes
S04	M	31	Glasses	graduate	CS	5	daily		Yes
S05	M	25	Glasses	graduate	CS	5	daily	daily	Yes
S06	F	21	No	undergraduate	BioChem/Bio	3	several times a week	once a week	Yes
S07	F	22	No	graduate	CS	5	daily	daily	No
S08	M	26	Glasses	graduate	ISE	4	several times a week	several times a week	No
S09	M	21	Glasses	undergraduate	Comp Eng	5	daily	daily	Yes
S10	M	22	Contacts	graduate	CS	5	daily	daily	Yes
S11	M	23	Contacts	graduate	CS	5	daily	daily	Yes
S12	F	31	Glasses	Faculty	HNFE	4.5	daily	daily	No
S13	F	30	Contacts	graduate	HNFE	4	daily	daily	No
S14	M	25	Glasses	graduate	CS	5	daily	daily	Yes
S15	M	24	Glasses	graduate	CS	5	daily	daily	Yes
S16	F	25	Contacts	graduate	HNFE	3	daily	several times a week	No

Experiment 2

Accuracy

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	1	1	1	0	0	0	1	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	0
S02	1	1	1	0	0	0	1	0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	0	0
S03	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
S04	1	1	1	0	1	1	1	0	1	1	0	0	1	0	1	1	1	1	1	1	1	0	0	0
S05	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0
S06	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
S07	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	0	0	1	1	1	0	0	0
S08	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0
S09	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
S10	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
S11	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	0
S12	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	0	1	1	0	1	0	0
S13	1	1	1	1	1	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	0	1	0	0
S14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
S15	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S16	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	0	0

	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
Ideal Time	2	3	2.5	2	4	5.4	4.22	5.08	2.72	3.88	4.38	3.96	3	7	7	4	4	6	2.11	4.08	5.79	5.11	4.62	5.72

Time Sub Ideal

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	47.31	46.89	103.16	300	66.11	265	40.99	52.4	46.83	78.82	176.82	113.21	20.72	31.48	13.95	44.93	52.83	193.95	44.17	27.51	134.72	168.03	121.79	300
S02	21.35	80.69	17.78	60.36	56.8	300	8.25	73.95	48.09	212.39	58.9	97.31	11.28	25.63	22.85	28.98	46.1	48.17	31.3	31.64	79.91	46.66	87.23	208.18
S03	23.28	53.25	27.75	53.14	55.86	75.16	43.93	74.37	78.3	29.12	73.89	69.88	24.93	32.66	21.51	46.66	49.25	40.29	56.95	32.9	72.37	73.7	83.15	60.07
S04	16.84	74.4	30.05	41.6	17.28	15.98	36.77	79.91	172.68	76.75	41.16	89.93	121.45	114.15	30.1	77.91	86.77	163.43	61	47.32	110.66	90.02	162.64	289.69
S05	9.9	54.195	137.525	36.595	56.605	81.445	24.68	199.78	175.443	300	300	300	26.94	71.875	26.82	67.145	175.53	38.18	198.795	32.89	73.08	69.185	226.68	69.075
S06	18.57	52.715	24.88	266.68	11.9	142.24	65.14	208.04	101.86	77.01	29.41	164.37	9.595	20.49	22.45	42.46	20.04	247	34.97	32.83	46.315	78.255	31.26	137.755
S07	88.12	75.53	161.92	83.3	35.515	155.045	65.03	201.195	296.16	63.615	58.1	32.08	24.455	19.98	28.185	65.05	62.61	300	77.01	51.14	137.78	264.465	300	143.19
S08	29.95	60.82	23.065	131.05	20.255	70.795	60.27	69.435	86.405	61.39	97.065	265.25	77.425	41.445	65.95	51.025	162.31	160.945	86.66	84.27	163	133.82	89.725	109.105
S09	17.015	38.055	21.265	48.065	16.895	52.78	64.8	111.08	38.11	20.645	23.9	122.555	14.415	18.365	16.37	19.43	22.525	20.91	30.55	19.7	44.895	39.21	149.345	116.93
S10	27.25	44.49	14.51	41.49	20.82	57.9	25.81	97.89	17.89	23.77	57.51	264.64	17.32	16.51	34.44	19.87	30.13	24.11	89.45	16.46	56.17	77.46	34.65	48.91
S11	26.625	97.555	76.18	37.855	22.2	7.495	21.675	32.23	41.545	46.355	42.175	120.335	25.415	38.775	47.375	43.375	47.17	44.195	112.455	82.24	109.835	72.54	86.165	129.065
S12	16.48	58.53	35.885	53.615	41.45	49.86	29.435	106.14	108.19	37.76	22.86	117.57	18.455	23.96	33.99	20.445	38.14	26.485	34.305	18.325	93.805	77.84	167.155	136.95
S13	33.46	86.745	22.82	45.21	30.095	224.575	118.575	107.415	165.24	31.575	164.485	187.495	25.255	33.595	34.73	28.66	23.955	300	30.13	41.57	89.565	95.865	32.8	58.41
S14	57.095	23.595	25.715	35.965	16.77	16.675	36.16	118.725	17.5	40.655	32.585	129.285	10.205	23.5	24.525	15.255	29.615	25.58	43.44	27.085	49.115	41.215	20.5	101.14
S15	40.605	34.75	49.365	9.285	21.92	13.215	38.75	195.875	29.23	63.09	38.125	204.74	16.755	74.61	28.845	23.51	22.2	33.135	33.015	38.155	83.995	115.825	126.755	90.845
S16	108.09	28.885	211.795	59.825	36.965	137.435	147.93	300	183.38	91.87	105.61	300	30.27	25.085	31.21	35.8	49.15	146.845	72.995	75.205	162.03	86.225	186.11	227.14

Experiment 2

Satisfaction

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	5	6	3	1	5	1	7	2	5	6	2	2	6	6	7	5	6	1	5	5	3	2	2	1
S02	5	5	3	6	6	4	7	6	6	4	6	5	7	6	3	7	6	3	6	5	4	4	4	7
S03	6	4	3	3	5	6	5	3	5	5	5	4	7	5	6	6	5	5	4	6	5	4	5	5
S04	5	6	3	6	6	5	5	6	3	3	6	3	5	5	6	6	6	4	6	5	6	5	4	1
S05	7	7	3	7	7	7	7	5	4	3	4	3	7	7	7	7	3	7	5	7	7	7	4	7
S06	5	4	3	3	6	4	5	5	4	6	6	5	6	6	6	5	6	5	5	6	6	6	5	5
S07	6	3	3	5	6	3	6	1	4	7	3	6	7	7	7	6	6	2	5	5	4	3	1	3
S08	6	6	3	6	6	6	6	6	5	6	6	5	6	6	6	6	6	5	6	6	6	6	6	5
S09	6	3	3	3	5	4	2	2	5	5	5	2	6	6	6	5	5	3	4	6	5	5	3	2
S10	6	3	3	4	4	3	5	3	4	3	3	1	4	4	4	4	4	3	4	4	4	3	3	4
S11	4	1	3	4	4	4	4	1	4	4	4	2	4	4	4	4	4	4	4	4	4	4	4	4
S12	1	5	3	4	5	6	5	5	6	5	7	5	6	7	6	7	6	3	6	7	7	6	4	2
S13	6	5	3	2	6	2	5	4	5	5	4	3	6	6	6	6	6	1	3	6	4	5	6	4
S14	4	4	3	4	4	3	3	3	4	3	4	4	4	4	4	4	4	4	3	4	3	3	4	3
S15	7	7	3	7	7	7	6	1	7	6	6	6	7	7	7	7	7	6	7	3	7	6	6	6
S16	7	7	3	6	7	1	7	5	6	7	7	3	7	7	7	7	7	1	6	7	7	7	5	2

Difficulty

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	3	4	5	5	3	5	2	2	4	2	4	6	4	3	2	1	2	6	1	1	7	6	6	7
S02	2	4	1	3	3	7	3	3	1	5	5	5	1	1	6	2	2	7	2	1	2	3	2	1
S03	2	5	3	6	5	5	3	6	4	3	5	5	1	3	3	3	5	4	5	3	5	5	6	5
S04	2	2	2	3	2	1	1	5	3	2	2	4	2	3	2	2	4	5	1	3	3	3	5	7
S05	1	2	4	2	1	1	1	3	3	6	4	7	1	2	1	3	1	2	3	1	1	2	7	1
S06	2	3	3	7	2	6	3	5	5	3	3	5	1	2	2	2	2	7	3	3	3	3	4	5
S07	1	3	3	2	1	4	1	7	5	1	4	1	1	1	1	1	1	7	2	1	3	6	7	4
S08	1	1	1	1	1	1	1	2	2	1	1	1	1	2	2	2	1	2	1	2	2	2	2	1
S09	2	3	2	5	2	4	4	5	4	2	2	6	1	2	2	3	3	3	4	2	3	3	6	6
S10	2	5	4	2	4	4	2	6	2	3	3	7	2	3	2	3	3	3	5	3	3	5	4	3
S11	4	6	7	4	4	2	4	5	3	5	5	6	2	3	4	3	2	6	7	6	7	7	7	7
S12	6	3	5	2	5	1	1	5	1	2	1	5	3	1	2	1	1	2	1	1	6	5	7	7
S13	2	2	1	5	2	6	3	5	4	1	6	6	1	1	2	1	2	7	5	2	5	3	1	4
S14	4	3	3	4	3	4	2	4	2	3	3	6	3	3	3	3	3	4	3	2	4	3	3	5
S15	1	2	2	1	1	1	1	6	1	3	1	5	1	2	1	1	1	1	1	1	5	3	5	4
S16	2	1	4	2	1	7	2	5	5	1	1	6	1	1	1	1	1	6	1	1	2	1	5	6

Experiment 2

Navigation Difficulty

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	5	4	5	7	4	7	2	2	2	2	2	2	3	4	2	2	2	6	2	2	6	5	6	7
S02	1	2	1	2	2	7	3	3	2	5	2	4	1	1	6	2	1	5	2	1	2	3	3	1
S03	3	5	3	6	4	4	3	6	5	3	4	3	4	3	3	4	3	5	3	3	4	6	6	4
S04	2	3	2	4	3	1	1	5	5	3	3	5	3	3	2	3	6	6	2	3	3	5	6	7
S05	1	7	1	1	1	1	1	3	3	6	6	7	1	1	1	1	1	1	2	1	1	1	1	1
S06	4	3	3	7	2	6	4	4	4	4	3	7	2	2	2	3	2	7	2	2	3	5	3	5
S07	1	4	5	2	1	4	1	3	6	1	4	1	1	1	1	2	7	5	4	4	1	6	1	5
S08	1	2	1	2	1	2	2	2	2	1	2	2	1	2	2	2	2	3	2	2	3	6	1	2
S09	2	2	2	2	2	3	2	4	2	2	2	5	2	2	2	2	2	3	2	2	2	2	4	5
S10	3	4	4	4	4	6	2	6	2	3	4	7	5	3	5	4	4	4	4	4	3	5	4	3
S11	5	7	6	4	5	3	3	5	4	5	5	6	4	3	1	4	3	5	7	7	6	7	7	7
S12	5	2	6	3	4	7	1	4	2	1	1	2	2	2	1	1	1	5	2	1	6	2	1	1
S13	3	2	1	2	2	6	3	5	3	2	5	5	1	2	3	1	2	7	2	2	3	5	2	5
S14	3	3	3	4	3	4	3	4	3	3	4	5	2	3	3	3	3	4	4	3	3	4	4	5
S15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
S16	1	1	4	2	1	4	2	2	1	1	1	3	1	1	1	1	1	3	1	1	2	2	4	4

2D Useful?

User	2D_t1	2D_t2	2D_t3	2D_t4	2D_t5	2D_t6	2D_t7	2D_t8	2D_t9	2D_t10	2D_t11	2D_t12	2D_t13	2D_t14	2D_t15	2D_t16	2D_t17	2D_t18	2D_t19	2D_t20	2D_t21	2D_t22	2D_t23	2D_t24
S01	5	6	6	3	2	1	7	6	6	6	5	5	6	6	7	7	6	1	1	6	6	6	6	2
S02	6	6	7	5	5	4	7	7	6	5	7	5	3	7	4	7	4	2	6	5	7	7	4	7
S03	2	6	6	3	4	4	2	5	5	5	5	4	1	6	6	4	5	1	1	6	6	5	1	1
S04	4	6	6	2	6	5	1	7	2	1	6	1	2	6	5	7	6	2	2	6	6	6	6	1
S05	7	7	6	7	5	7	1	6	5	4	6	4	4	7	7	7	7	1	4	7	7	1	1	1
S06	3	6	6	2	7	3	5	7	6	3	6	4	7	7	7	6	6	2	4	7	7	5	6	3
S07	7	7	7	5	5	1	1	1	1	2	1	2	1	7	7	5	5	1	1	3	5	1	1	1
S08	6	6	7	6	6	5	6	6	7	7	7	6	7	7	7	6	7	1	1	7	6	7	7	5
S09	2	6	6	2	4	3	6	7	4	3	5	2	3	6	6	5	5	1	2	7	6	4	3	2
S10	6	5	6	6	6	5	1	4	5	1	4	1	2	2	4	5	3	2	3	5	6	3	3	2
S11	1	4	7	1	2	2	1	4	3	1	1	1	1	5	5	6	5	4	4	6	6	2	2	4
S12	5	4	6	5	7	7	7	1	7	7	7	7	6	7	7	7	6	1	1	7	7	7	4	1
S13	7	6	7	5	7	6	6	6	6	6	7	2	1	7	7	7	7	2	1	7	6	6	7	5
S14	5	4	4	2	4	3	4	5	5	4	2	3	1	5	4	4	2	1	1	5	3	2	1	1
S15	5	6	6	5	6	5	1	6	4	3	4	4	1	6	6	6	6	1	1	7	5	1	6	1
S16	7	7	7	7	7	1	5	2	1	7	6	3	7	7	7	7	7	1	6	7	7	7	1	1

Experiment 2

3D Useful?

User	3D_t1	3D_t2	3D_t3	3D_t4	3D_t5	3D_t6	3D_t7	3D_t8	3D_t9	3D_t10	3D_t11	3D_t12	3D_t13	3D_t14	3D_t15	3D_t16	3D_t17	3D_t18	3D_t19	3D_t20	3D_t21	3D_t22	3D_t23	3D_t24
S01	3	2	6	1	5	1	5	5	6	6	5	6	4	5	6	2	6	5	6	6	6	7	4	6
S02	6	4	6	5	7	6	7	1	7	5	6	5	7	7	7	7	7	5	7	6	5	7	6	7
S03	6	6	5	4	6	6	6	4	6	6	6	5	4	6	5	4	5	6	6	3	4	6	6	6
S04	6	4	5	5	6	5	7	2	4	5	6	2	6	3	5	5	6	4	6	5	4	5	5	1
S05	7	7	6	7	7	7	1	5	5	6	4	3	7	7	7	7	2	7	6	7	7	7	5	7
S06	6	6	3	6	6	4	3	4	6	6	5	6	3	6	6	5	5	6	3	6	6	4	5	6
S07	7	3	2	5	6	3	6	2	2	7	2	6	7	7	7	6	6	1	5	1	7	2	1	3
S08	6	6	7	6	7	6	6	6	6	6	6	5	6	6	6	6	6	6	6	6	6	5	7	6
S09	3	2	4	3	6	6	2	2	6	6	5	6	2	4	4	5	6	6	5	2	5	5	5	5
S10	6	2	4	6	3	5	5	3	4	6	5	2	5	5	4	5	5	4	4	3	4	4	3	3
S11	5	4	6	4	5	3	5	1	5	6	6	6	6	5	1	5	2	7	6	6	6	6	6	4
S12	2	6	3	3	6	5	4	2	7	5	7	3	6	6	5	6	6	3	7	7	2	4	2	6
S13	6	5	7	5	7	6	6	3	4	6	6	6	7	6	5	3	6	2	2	5	5	5	6	3
S14	4	2	3	4	3	4	5	2	5	5	5	5	3	3	2	4	3	4	4	2	2	4	4	4
S15	6	6	4	7	7	7	6	4	6	6	6	7	7	6	6	7	6	7	7	4	5	7	6	7
S16	7	7	2	3	7	1	7	3	4	7	6	5	6	7	7	2	7	1	6	4	7	7	6	3

C.3 Descriptive Statistics

Experiment 2 Object vs. Display Space

ACCURACY

Descriptive Statistics per task

	Mean	Std. Deviation	N
OB_S_AS	.9792	8.333E-02	16
OB_S_SA	.6042	.3270	16
OB_C_AS	.7083	.2064	16
OB_C_SA	.6458	.3096	16
DIS_S_AS	.9583	.1139	16
DIS_S_SA	.7500	.3549	16
DIS_C_AS	.8542	.2425	16
DIS_C_SA	.4375	.3594	16

TIME

Descriptive Statistics per task

	Mean	Std. Deviation	N
OBJ_S_AS	51.5979	27.0870	16
OBJ_S_SA	72.8557	53.3939	16
OBJ_C_AS	92.9892	51.8351	16
OBJ_C_SA	107.4179	62.1906	16
DIS_S_AS	32.7145	18.8788	16
DIS_S_SA	70.0428	42.3179	16
DIS_C_AS	66.7433	26.3982	16
DIS_C_SA	117.9734	52.9016	16

SATISFACTION

Descriptive Statistics

	Me	Std. Deviation	N
OBJ_S_AS	4.833333	1.074968	16
OBJ_S_SA	4.708333	1.275844	16
OBJ_C_AS	4.583333	1.007380	16
OBJ_C_SA	4.479167	1.121961	16
DIS_S_AS	5.833333	1.054093	16
DIS_S_SA	4.916667	.811948	16
DIS_C_AS	5.145833	1.003466	16
DIS_C_SA	4.229167	1.280733	16

TASK DIFFICULTY

Descriptive Statistics per task

	Mean	Std. Deviation	N
OBJ_S_AS	2.833333	1.211060	16
OBJ_S_SA	3.187500	1.360658	16
OBJ_C_AS	3.270833	.912617	16
OBJ_C_SA	3.625000	1.252405	16
DIS_S_AS	1.979167	.704154	16
DIS_S_SA	2.875000	.909823	16
DIS_C_AS	2.895833	1.331770	16
DIS_C_SA	4.375000	1.548596	16

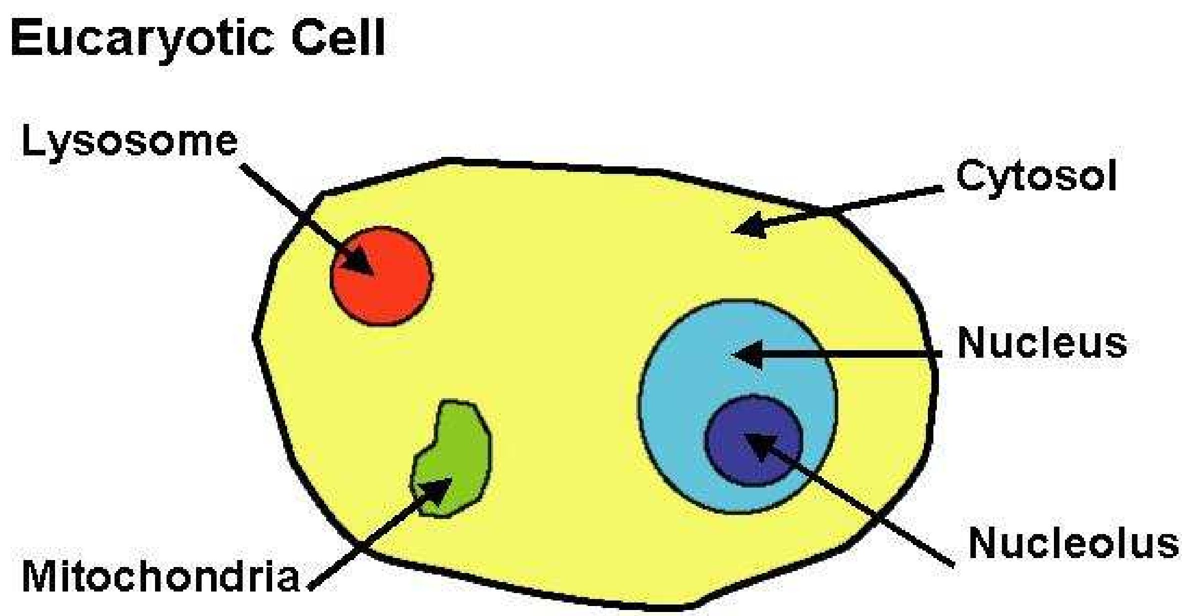
NAVIGATION DIFFICULTY

Descriptive Statistics per task

	Mean	Std. Deviation	N
OBJ_S_AS	2.937500	1.356568	16
OBJ_S_SA	3.312500	1.483084	16
OBJ_C_AS	2.916667	.992565	16
OBJ_C_SA	3.270833	1.555010	16
DIS_S_AS	2.166667	.958394	16
DIS_S_SA	3.062500	1.218492	16
DIS_C_AS	2.708333	1.332639	16
DIS_C_SA	3.833333	1.721326	16

D. Experiments 3 & 4

User Guide



1	2	3	4	5	6
Least Difficult					**Most Difficult**

1	2	3	4	5	6
Least Satisfying					**Most Satisfying**

Experiment 3

Experiment Description

Thank you for participating in this experiment. Your input in this experiment has the potential to impact how virtual environments are designed. You will be asked to perform a set of tasks using a large screen virtual environment system. These tasks consist of navigating through a 3D cell environment finding and comparing molecule objects.

The session will last about forty-five minutes. You are welcome to take rest breaks as needed. One scheduled rest break will be given about half-way through the experiment. You may also terminate your participation at any time, for any reason.

We are not evaluating you; rather, you are helping us to evaluate the system. All information that you help us attain will remain anonymous. The time you take to do each task and other aspects of your interaction with the system will be measured. You may be asked questions during and after the evaluation.

You will have the opportunity to explore the program before beginning the experiment. Before each task, you will be given full instructions. It is important that you understand the instructions before beginning each task. If anything is unclear, be sure to ask before beginning the task.

The results of this study will be kept strictly confidential. The information you provide will have your name removed and only a subject number will identify your evaluation.

Experiment 3 & 4

User Questionnaire

Please help us to categorize our user population by completing the following items.

Gender (circle one): Male Female

Age: ______________ Do you wear glasses or contact lenses (circle one)?

No Glasses Contact Lenses

Occupation (if student, indicate graduate or undergraduate):

____

Major / Area of specialization (if student): ___________________________________

Rate your familiarity with computers: (circle one)

•--------------------------•--------------------------•--------------------------•--------------------------•

not at all familiar not very familiar somewhat familiar fairly familiar very familiar

How often do you use computers... (circle the best answer)

...for work? . *...for fun?*

...for work?	...for fun?
a. not at all	a. not at all
b. once a month	b. once a month
c. once a week	c. once a week
d. several times a week	d. several times a week
e. daily	e. daily

What are the applications you use most (please check all that apply)

___ Desktop publishing
___ Word processing
___ Data processing
___ Architecture / CAD / 3D modeling
___ Games
___ Statistical Analysis
___ Information Visualization

Have you ever used a virtual reality (VR) system? If so, please describe it (what type of display was used, what kind of application (e.g. game, architectural walk-through) was running, how did you interact with the system, etc.).

E. Experiment 3

E.1 Materials

Evaluation of Object Space Layouts in Information-Rich Virtual Environments

Justification of Project

This project involves testing human performance on various tasks using Object Space layouts in Information-Rich Virtual Environments (IRVEs). IRVEs combine virtual environments (VEs) and information visualizations. The purpose of this investigation is to understand the usability tradeoffs between IRVE layout techniques and their appropriateness for supporting search and comparison tasks. Previous work has shown Object Space layouts to be advantageous on large displays. This experiment seeks to further optimize Object Space layouts and annotation scaling factors for such systems where depth cues must be managed for association and occlusion. This research will directly contribute to development of new and efficaceous IRVE interfaces. Human subjects must be used so that we can quantify how real users interact with the environment on the tasks involved.

Procedures

For this experiment, approximately 24 student subjects will be recruited. The only selection criteria will be near-perfect vision (corrected or uncorrected) and the ability to operate a desktop computer. Subjects of any age (over 18) or gender will be accepted into the study.

Subjects will attend one session of approximately 1 hour in length. The session will include:

1. Read and fill out user questionnairre
2. Virtual environment training (mouse controls & metaphors)
3. Read a set of instructions for the experiment (enclosed)
4. Explore an example IRVE cell with 2 example comparison tasks
5. Perform various tasks using IRVE interface (enclosed)

The specific technology used in the experiment will be a large projection display system without head-tracking or stereo-rendering. Subjects will be sitting and all tasks can be completed with a mouse. For example, tasks will involve navigating through a virtual cell, selecting and inspecting molecules.

Risks and Benefits

Subjects will be allowed to take breaks during the experiment and to quit the experiment if they feel dizzy or nauseous. See enclosed Informed Consent form for more information.

Confidentiality/Anonymity

Only the investigators will have access to the data. Each subject will be identified by number only on data forms and reports. When reporting data, only aggregate information and statistical analysis of the whole group of subjects will be published. See enclosed Experiment Description for more information.

Informed Consent

Not Required.

Virginia Tech
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Institutional Review Board

Carmen Green
IRB Administrator
Research Compliance Office
1880 Pratt Drive, Suite 2006(0497), Blacksburg, VA 24061
Office: 540/231-4358; FAX: 540/231-0959
email: ctgreen@vt.edu

DATE: October 4, 2005

MEMORANDUM

TO: Doug A. Bowman Computer Science 0106
Nicholas Polys

FROM: Carmen Green

SUBJECT: **IRB Exempt Approval:** "Object Space Layouts for Large Screen IRVEs" IRB # 05-459

I have reviewed your request to the IRB for exemption for the above referenced project. I concur that the research falls within the exempt status. Approval is granted effective as of October 4, 2005.

Virginia Tech has an approved Federal Wide Assurance (FWA00000572, exp. 7/20/07) on file with OHRP, and its IRB Registration Number is IRB00000667.

cc: File

Experiment 3

Task key

Experiment 3					
ID	**Question**		**Answer**	**Technique**	**Scaling**
	S: A ->S				
_t1	Where in the cell is the molecule with a molecular weight of 85.15?	a	Lysosome (cyclop)	SBounds	Normal
_t2	Where in the cell is the molecule with a molecular weight of 89.09?	b	Mito (alanine)	SBounds	Periodic
_t3	Where in the cell is the molecule with a molecular weight of 118.9?	c	Nucleolus - butan	SBounds	Continuous
_t4	Where in the cell is the molecule with a molecular weight of 88.06?	d	Lysosome (pyru)	ForceD	Normal
_t5	Where in the cell is the molecule with a molecular weight of 194.19?	e	Mito (caff)	ForceD	Periodic
_t6	Where in the cell is the molecule with a molecular weight of 176.17?	f	Nucleolus - gycerol	ForceD	Continuous
	S: S -> A				
_t7	What molecule is right next to the nucleus?	g	3-methyl-idene	SBounds	Normal
_t8	What molecule is right next to the nucleus?	h	cyclohexine oxide	SBounds	Periodic
_t9	What molecule is right next to the nucleus?	I	1,4dihidroneapthalene	SBounds	Continuous
_t10	What molecule is right next to the nucleus?	j	vitamink (cl)	ForceD	Normal
_t11	What molecule is right next to the nucleus?	k	pyruvic acid	ForceD	Periodic
_t12	What molecule is right next to the nucleus?	l	glycerol_diacetate	ForceD	Continuous
	C: S->A				
_t13	What molecule is the furthest North?	m	dimethyl sulpoxide	SBounds	Normal
_t14	What molecule is the furthest North?	n	Butanenioitic Acid	SBounds	Periodic
_t15	What molecule is the furthest North?	o	caffeine	SBounds	Continuous
_t16	What molecule is the furthest North?	p	tartaric acid	ForceD	Normal
_t17	What molecule is the furthest North?	q	vitamin k	ForceD	Periodic
_t18	What molecule is the furthest North?	r	alanine	ForceD	Continuous
	C: A->S				
_t19	Where in the cell is the molecule with the highest boiling point?	s	Lysosome (Vitamin K)	SBounds	Normal
_t20	Where in the cell is the molecule with the lowest boiling point?	t	Mitochondria (3-methyl…)	SBounds	Periodic
_t21	Where in the cell is the molecule with the highest boiling point?	u	Nucleolus (Acetic Acid)	SBounds	Continuous
_t22	Where in the cell is the molecule with the lowest boiling point?	v	Cytosol (Butanoic Acid)	ForceD	Normal
_t23	Where in the cell is the molecule with the highest boiling point?	w	Mitochondria (Alanine)	ForceD	Periodic
_t24	Where in the cell is the molecule with the lowest boiling point?	x	Lysosome (Caffeine)	ForceD	Continuous

E.2 Results

Experiment 3										
User	**Gender**	**Age**	**Vision Aids**	**Occupation**	**Major**	**Computer Familiarity**	**Comp. Usage (work)**	**Comp. Usage (fun)**	**VR Experience**	**Applications**
S01	F	19	No	Ugrad	HNFE	fairly	daily	several times a week	No	WP
S02	F	22	Glasses	Ugrad	Psych	somewhat	daily	once a week	No	WP
S03	F	21	No	Ugrad	Psych	somewhat	-	daily	No	WP, GA
S04	F	20	C Lenses	Ugrad	Psych	not very	daily	daily	No	WP
S05	F	18	C Lenses	Ugrad	Spanish	fairly	daily	daily	HMD once	WP
S06	F	19	C Lenses	Ugrad	Comm.	somewhat	several times a week	once a month	No	WP
S07	F	18	No	Ugrad	History	somewhat	several times a week	daily	No	WP
S08	F	19	No	Ugrad	HNFE	fairly	daily	daily	No	WP
S09	F	20	C Lenses	Ugrad	Psych	fairly	daily	daily	No	WP,DP,SA
S10	F	17	No	Ugrad	U Studies	very	daily	daily	No	WP,A,GA
S11	M	20	C Lenses	Ugrad	Accnt	somewhat	-	daily	No	WP
S12	F	18	C Lenses	Grad	Marktng	fairly	daily	daily	No	WP,DP,GA
S13	F	18	No	Ugrad	-	somewhat	several times a week	daily	No	WP

Experiment 3		
Cognitive Battery Scoring: (Right-Wrong)		
	Hidden Patterns	**Number Comparison**
S01	3	28
S02	82	29
S03	117	27.5
S04	47	20.5
S05	100	30.5
S06	98	29.5
S07	57	23
S08	64	20.5
S09	60	25
S10	43	12
S11	86	12.5
S12	94	24
S13	77	27

Experiment 3

Accuracy

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1	1
S02	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	1	1	1	0	1
S03	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
S04	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S05	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
S06	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
S07	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	1	0	1	1
S08	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S09	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1
S10	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1
S11	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1	1
S12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
S13	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0

Time

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	131.693	51.065	53.7	52.405	55.103	41.335	88.439	86.6	33.414	147.176	63.726	90.934	51.611	97.655	54.916	125.736	135.637	150.404	145.539	129.541	141.391	166.276	110.502	68.294
S02	183.06	73.58	78.269	56.055	84.144	54.495	60.433	137.945	76.633	101.615	107.742	89.889	84.104	56.144	110.686	110.498	65.828	245.331	109.634	160.926	136.808	89.715	112.547	174.168
S03	67.747	57.19	114.88	76.738	48.307	141.226	161.621	52.474	117.449	190.552	76.362	62.641	82.737	64.003	62.687	69.736	118.701	68.374	93.765	176.878	110.807	170.237	234.396	40.053
S04	112.483	65.664	115.255	27.193	46.428	95.457	38.079	69.877	**53.767**	61.951	44.767	41.196	38.157	35.917	47.133	58.192	31.719	34.32	110.118	127.129	120.19	158.425	147.335	67.246
S05	54.025	84.429	85.698	49.357	22.869	56.97	294.326	71.788	68.311	80.403	49.263	39.191	34.022	70.034	47.399	62.985	75.25	89.692	137.467	91.024	128.492	145.456	146.317	101.879
S06	95.717	98.796	214.312	76.313	54.763	47.03	75.609	41.875	63.192	58.007	45.965	40.639	52.596	59.375	23.093	121.141	61.319	69.64	84.95	134.442	115.343	118.547	79.594	68.03
S07	243.656	69.75	102.953	48.578	29.453	74.641	28.875	45.969	35.703	135.141	103.015	58.203	133.625	50.187	30.859	61.907	81.843	41.125	217.156	96.484	75.688	169.593	140.156	51.031
S08	131.7	70.875	119.454	79.266	51.938	92.844	39.422	101.297	37.969	92.11	62.672	31.906	57.25	85.687	107.968	36.734	67.234	60.516	61.906	94.297	88.531	104	146.218	61.375
S09	41.723	39.931	225.69	76.033	35.285	51.961	25.425	44.295	67.409	58.703	35.051	68.509	20.354	18.89	49.608	31.578	40.297	29.648	67.986	120.399	61.252	139.84	77.914	84.632
S10	202.384	152.787	144.607	137.291	67.309	96.163	80.107	70.662	111.14	228.96	79.878	79.848	76.216	108.234	46.748	122.985	162.318	235.375	236.421	134.719	145.359	224.291	236.532	287.62
S11	159.203	41.828	144.172	24.718	25.016	36.437	49.359	49.281	36.172	43.625	34.687	51.922	27.5	23.719	17.125	16.937	25.485	27.843	73.297	54.453	92.938	51.328	107.36	44.265
S12	116.218	73.984	52.327	39.285	179.969	87.631	112.2	197.468	76.395	156.36	108.641	86.232	100.063	71.485	*151.203*	116.595	*136.769*	112	139.297	118.188	149.703	216.17	134.797	82.516
S13	183.547	*170.297*	151.328	29.656	*23.203*	41.156	25.656	33.844	28.906	33.906	34.297	18.86	50.906	23.672	21.453	88.078	26.656	23.907	81.875	65.14	81.985	105.703	72.109	49.265

Distance Navigated

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	905.4255	499.7663	558.0364	885.7286	538.7115	642.8646	684.1602	792.7962	574.3241	13077.27	666.5454	655.9429	766.277	865.7605	367.5391	1127.717	1710.572	2581.969	1590.676	386.3472	1583.388	2288.182	1260.788	846.2065
S02	425.4934	497.8842	575.0069	762.1171	526.0112	464.3245	452.1321	534.7743	646.1351	476.056	419.5905	1000.069	563.342	516.6353	739.4417	505.5057	454.6707	410.4567	833.1123	912.0003	760.6993	1524.896	720.0517	1225.37
S03	512.4969	468.3385	902.1341	793.7447	514.974	530.8342	601.0318	608.7051	676.0369	2231.474	790.5992	1040.686	607.4213	727.3476	925.8222	1608.011	4054.133	3184.064	632.8916	405.7568	695.4616	2396.89	1977.299	1621.665
S04	763.6242	576.3169	1292.386	517.059	446.839	509.8534	715.9464	943.3135	**539.191**	917.9241	523.6794	818.3509	848.7695	553.852	454.9718	555.7373	610.945	612.0282	1351.125	1203.638	1208.08	968.0113	747.239	420.6755
S05	1164.407	748.1287	653.0998	504.4541	1626.692	879.5335	1100.477	1200.799	807.6457	1955.39	2792.404	781.1193	1069.009	936.7753	5304.393	1351.88	2086.162	3428.841	1836.039	1737.631	1076.992	3278.578	1482.1	919.8992
S06	5210.243	2727.429	883.0857	450.3088	0	321.3924	491.19	501.2108	402.8865	700.1485	319.4027	356.4656	577.8142	475.3302	0	614.5676	646.9508	415.0938	1041.925	415.353	655.828	1442.32	562.4185	546.19
S07	871.9263	471.3189	646.8872	551.0088	716.5982	496.5215	553.4663	2172.651	831.277	896.033	10392.76	1394.276	798.6111	826.2108	915.2035	1018.037	887.2972	5025.246	951.1382	1336.751	1252.641	1718.791	3134.31	5098.768
S08	765.9999	572.3509	1229.016	683.2538	531.0812	1404.333	1473.199	581.583	1021.291	1370.902	1754.448	730.4793	878.0215	759.2373	350.9983	1303.861	681.8316	915.458	1613.386	1618.453	912.5681	3349.508	4243.14	455.3353
S09	1212.684	571.4775	1160.608	523.4992	306.9102	471.4411	492.8521	710.6478	646.1351	1062.034	510.4752	435.2143	588.5593	476.9291	458.6528	592.8084	478.8037	510.5212	1107.301	1558.682	968.6644	5000.75	1822.177	807.2466
S10	561.2121	687.5231	1101.048	708.381	373.4987	501.665	597.8878	985.7402	529.0062	1247.546	617.1194	424.7508	805.1895	1066.921	891.1865	1247.672	1551.043	1427.689	842.4315	920.7422	786.9817	2265.602	996.1176	776.6247
S11	1158.389	737.1088	999.4509	503.7605	900.5833	571.1627	791.1193	839.8915	679.9707	1431.162	694.0479	641.9694	849.4064	954.2455	806.4714	3312.367	812.2116	673.3145	1127.796	1054.177	648.5239	1473.962	622.1819	1151.701
S12	1503.686	517.8896	1154.281	587.1443	474.5416	478.9401	448.6701	514.8631	420.7424	723.7383	1166.709	596.5738	1188.245	771.3219	*401.875*	842.5407	*664.313*	434.0344	1444.914	754.8997	712.6258	3111.179	1845.953	405.3233
S13	1549.794	*813.059*	1110.936	937.051	*728.397*	518.0091	607.639	843.126	547.871	1050.579	1293.798	562.6423	1039.732	904.5615	1128.628	1423.825	1116.28	945.0741	753.4128	750.2366	1249.63	2105.952	1955.574	896.5032

Experiment 3

Satisfaction

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	5	3	3	3	3	4	4	3	3	3	3	3	3	3	3	2	5	2	4	4	4	4	3	4
S02	5	2	3	3	4	4	4	2	3	4	3	4	4	3	2	4	3	3	3	5	4	4	4	4
S03	4	5	3	4	5	4	4	4	3	2	2	4	4	4	4	4	5	3	4	3	4	4	2	5
S04	2	5	2	5	4	3	5	5	4	4	3	4	5	6	5	4	6	3	3	4	5	2	3	3
S05	4	5	4	5	6	4	5	4	4	3	5	4	5	4	5	4	4	3	5	3	3	3	4	3
S06	3	5	4	2	4	6	4	3	4	3	4	5	3	4	4	2	4	5	3	4	4	2	5	5
S07	3	4	4	5	6	5	6	5	6	3	5	5	3	5	6	3	1	5	1	4	5	1	4	5
S08	3	4	5	4	5	3	5	3	3	3	3	5	4	3	3	4	3	4	4	3	4	4	3	4
S09	6	6	2	5	6	5	5	6	3	5	5	5	6	6	6	5	4	5	6	5	6	2	5	6
S10	5	5	4	3	4	3	5	4	4	4	3	5	4	3	4	3	3	4	4	5	5	4	4	4
S11	2	4	3	5	6	5	5	4	4	2	4	4	3	3	5	5	3	5	3	2	2	2	2	5
S12	5	5	5	4	5	4	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	4	5	5
S13	3	3	4	6	6	4	6	5	6	6	6	6	5	6	6	5	5	6	5	4	4		4	5

Difficulty

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	4	2	1	2	2	2	3	4	2	3	2	3	3	3	2	4	6	6	4	4	3	5	3	2
S02	6	2	3	2	4	3	4	5	4	5	4	4	5	4	6	4	4	6	3	5	5	5	4	4
S03	3	1	5	2	2	4	3	4	4	5	4	3	2	3	4	4	4	3	2	4	3	3	4	2
S04	5	2	4	1	2	3	2	3	2	3	3	2	2	2	1	2	1	3	4	4	5	5	4	5
S05	3	3	5	3	2	3	3	4	2	6	3	2	3	5	3	4	5	5	4	4	5	5	4	5
S06	4	4	5	3	3	3	3	3	3	3	2	2	3	3	2	3	4	3	3	4	4	4	4	3
S07	6	3	3	2	1	3	1	2	1	5	4	2	5	3	1	4	3	2	6	3	4	5	6	2
S08	5	3	4	3	2	3	2	5	4	3	4	2	3	4	4	3	5	4	4	5	3	3	4	4
S09	1	1	6	3	1	2	2	1	3	5	2	3	1	1	1	3	4	2	2	3	1	5	1	1
S10	3	4	2	4	2	3	3	3	4	3	5	2	3	4	4	5	5	4	5	3	4	4	4	5
S11	3	2	6	2	1	2	2	3	4	4	4	4	4	4	5	5	4	3	3	4	4	3	4	2
S12	4	3	4	3	5	2	4	6	2	5	5	4	5	5	5	5	6	5	5	4	4	5	5	4
S13	5	5	5	2	1	3	1	3	1	2	2	1	3	1	1	4	3	2	3	4	6		4	3

E.3 Descriptive Statistics

Layout Technique: B = ScreenBounds, F = ForceDirected
Scaling: N = None, P = Periodic, C = Continuous

ACCURACY
Descriptive Statistics - Overall

	Mean	Std. Deviation	N
BN	.9423	.1096	13
BP	.8077	.1498	13
BC	.8846	.1941	13
FN	.8654	.1651	13
FP	.8077	.1498	13
FC	.8462	.1626	13

TIME
Descriptive Statistics - Overall

	Mean	Std. Deviation	N
BN	99.44728846	33.11118784	13
BP	83.11871154	21.87214095	13
BC	89.00903846	16.54150823	13
FN	97.00153846	36.40427620	13
FP	83.55165385	32.79393796	13
FC	77.80509615	39.80387708	13

NAVIGATION DISTANCE
Descriptive Statistics - Overall

	Mean	Std. Deviation	N
BN	986.95574075	334.67044122	13
BP	846.31709905	202.53624429	13
BC	881.64917959	348.74227613	13
FN	1576.47926248	917.95302959	13
FP	1283.67269870	913.21536184	13
FC	1018.55218396	699.85192466	13

SATISFACTION
Descriptive Statistics - Overall

	Mean	Std. Deviation	N
BN	4.1731	.7527	13
BP	4.1346	.8204	13
BC	4.0577	.6934	13
FN	3.6859	.8001	13
FP	4.1154	.6663	13
FC	4.3077	.7648	13

DIFFICULTY

Descriptive Statistics - Overall

	Mean	Std. Deviation	N
BN	3.3654	.8392	13
BP	3.3462	.7809	13
BC	3.4423	.7981	13
FN	3.6282	.6025	13
FP	3.4038	.8135	13
FC	3.0769	.6645	13

F. Experiment 4

F.1 Materials

Evaluation of Viewport Space Layouts in Desktop Information-Rich Virtual Environments

Justification of Project

This project involves testing human performance on various tasks using Object Space layouts in Information-Rich Virtual Environments (IRVEs). IRVEs combine virtual environments (VEs) and information visualizations. The purpose of this investigation is to understand the usability tradeoffs between IRVE layout techniques and their appropriateness for supporting search and comparison tasks. Previous work has shown Viewport Space layouts to be advantageous on conventional screen IRVEs. This experiment seeks to further optimize Viewport Space layouts for such systems where visual configuration must be managed for association and occlusion. This research will directly contribute to development of new and efficaceous IRVE interfaces. Human subjects must be used so that we can quantify how real users interact with the environment on the tasks involved.

Procedures

For this experiment, approximately 24 student subjects will be recruited. The only selection criteria will be near-perfect vision (corrected or uncorrected) and the ability to operate a desktop computer. Subjects of any age (over 18) or gender will be accepted into the study.

Subjects will attend one session of approximately 1 hour in length. The session will include:

1. Read and fill out user questionnaire
2. Virtual environment training (mouse controls & metaphors)
3. Read a set of instructions for the experiment (enclosed)
4. Explore an example IRVE cell with 2 example comparison tasks
5. Perform various tasks using IRVE interface (enclosed)

The specific technology used in the experiment will be a single monitor desktop system without head-tracking or stereo-rendering. Subjects will be sitting and all tasks can be completed with a mouse. For example, tasks will involve navigating through a virtual cell, selecting and inspecting molecules.

Risks and Benefits

Subjects will be allowed to take breaks during the experiment and to quit the experiment if they feel dizzy or nauseous. See enclosed Informed Consent form for more information.

Confidentiality/Anonymity

Only the investigators will have access to the data. Each subject will be identified by number only on data forms and reports. When reporting data, only aggregate information and statistical analysis of the whole group of subjects will be published. See enclosed Experiment Description for more information.

Informed Consent

Not Required.

VirginiaTech

Office of Research Compliance
1880 Pratt Drive (0497)
Blacksburg, Virginia 24061
540/231-4358 Fax: 540/231-0959
E-mail: ctgreen@vt.edu
www.irb.vt.edu
FWA00000572(expires 7/20/07)
IRB # is IRB00000667.

DATE: March 14, 2006

MEMORANDUM

TO: Doug A. Bowman
Nicholas Polys

FROM: Carmen Green

SUBJECT: **IRB Exempt Approval:** "Viewport Space Layouts in Desktop Irves" , IRB # 06-161

I have reviewed your request to the IRB for exemption for the above referenced project. I concur that the research falls within the exempt status. Approval is granted effective as of March 14, 2006.

As an investigator of human subjects, your responsibilities include the following:

1. Report promptly proposed changes in previously approved human subject research activities to the IRB, including changes to your study forms, procedures and investigators, regardless of how minor. The proposed changes must not be initiated without IRB review and approval, except where necessary to eliminate apparent immediate hazards to the subjects.

2. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

cc: File
Department Reviewer:Doug A. Bowman

Invent the Future

VIRGINIA POLYTECHNIC INSTITUTE UNIVERSITY AND STATE UNIVERSITY

An equal opportunity, affirmative action institution

Task key

Experiment 3					
ID	Question		Answer	Technique	Scaling
	S: A ->S				
_t1	Where in the cell is the molecule with a molecular weight of 85.15?	a	Lysosome (cyclop)	Proximal	Line
_t2	Where in the cell is the molecule with a molecular weight of 89.09?	b	Mito (alanine)	Proximal	Polygon
_t3	Where in the cell is the molecule with a molecular weight of 118.9?	c	Nucleolus - butan	Proximal	Semi-transp poly
_t4	Where in the cell is the molecule with a molecular weight of 88.06?	d	Lysosome (pyru)	Semantic	Line
_t5	Where in the cell is the molecule with a molecular weight of 194.19?	e	Mito (caff)	Semantic	Polygon
_t6	Where in the cell is the molecule with a molecular weight of 176.17?	f	Nucleolus - gycerol	Semantic	Semi-transp poly
	S: S -> A				
_t7	What molecule is right next to the nucleus?	g	3-methyl-idene	Proximal	Line
_t8	What molecule is right next to the nucleus?	h	cyclohexine oxide	Proximal	Polygon
_t9	What molecule is right next to the nucleus?	I	1,4dihidroneapthalene	Proximal	Semi-transp poly
_t10	What molecule is right next to the nucleus?	j	vitamink (cl)	Semantic	Line
_t11	What molecule is right next to the nucleus?	k	pyruvic acid	Semantic	Polygon
_t12	What molecule is right next to the nucleus?	l	glycerol_diacetate	Semantic	Semi-transp poly
	C: S->A				
_t13	What molecule is the furthest North?	m	dimethyl sulpoxide	Proximal	Line
_t14	What molecule is the furthest North?	n	Butanenioitic Acid	Proximal	Polygon
_t15	What molecule is the furthest North?	o	caffeine	Proximal	Semi-transp poly
_t16	What molecule is the furthest North?	p	tartaric acid	Semantic	Line
_t17	What molecule is the furthest North?	q	vitamin k	Semantic	Polygon
_t18	What molecule is the furthest North?	r	alanine	Semantic	Semi-transp poly
	C: A->S				
_t19	Where in the cell is the molecule with the highest boiling point?	s	Lysosome (Vitamin K)	Proximal	Line
_t20	Where in the cell is the molecule with the lowest boiling point?	t	Mitochondria (3-methyl…)	Proximal	Polygon
_t21	Where in the cell is the molecule with the highest boiling point?	u	Nucleolus (Acetic Acid)	Proximal	Semi-transp poly
_t22	Where in the cell is the molecule with the lowest boiling point?	v	Cytosol (Butanoic Acid)	Semantic	Line
_t23	Where in the cell is the molecule with the highest boiling point?	w	Mitochondria (Alanine)	Semantic	Polygon
_t24	Where in the cell is the molecule with the lowest boiling point?	x	Lysosome (Caffeine)	Semantic	Semi-transp poly

F.2 Results

Experiment 4

User	Gender	Age	Vision Aids	Occupation	Major	Computer Familiarity	Comp. Usage (work)	Comp. Usage (fun)	VR Experience	Applications
S01	F	21	C lenses	Ugrad	Psych	very	daily	daily	No	D, WP, DP, GA
S02	M	22	C lenses	Ugrad	Psych	very	several times a week	daily	No	WP, GA
S03	M	19	No	Ugrad	-	fairly	daily	daily	No	WP, GA
S04	M	20	No	Ugrad	Mech Eng	very	several times a week	daily	No	WP, DP, GA
S05	M	18	C lenses	Ugrad	BioChem	fairly	daily	daily	No	WP, DP, GA
S06	M	19	Glasses	Ugrad	BioChem	somewhat	daily	several times a week	No	WP. GA,SA
S07	F	18	C lenses	Ugrad	Comm.	somewhat	daily	several times a week	No	WP
S08	F	18	No	Ugrad	HNFE	very	daily	daily	No	WP, DP, GA
S09	F	21	C lenses	Ugrad	Psych	very	daily	daily	No	D, WP, DP,GA
S10	F	21	C lenses	Ugrad	Psych	somewhat	daily	daily	No	WP
S11	M	18	No	Ugrad	U. Stud,	fairly	daily	several times a week	No	WP, GA
S12	M	21	No	Ugrad	Psych	fairly	daily	daily	No	WP, DP,SA
S13	F	19	C lenses	Ugrad	Biol	fairly	daily	daily	No	WP
S14	F	19	No	Ugrad	MathEd	very	daily	daily	No	WP, GA
S15	F	20	No	Ugrad	Psych	fairly	daily	daily	No	WP, GA
S16	F	18	No	Ugrad	U. Stud,	very	daily	daily	No	WP, DP, SA
S17	F	19	Glasses	Ugrad	Psych, Bio	somewhat	daily	daily	No	WP
S18	M	19	Glasses	Ugrad	History	fairly	several times a week	daily	No	WP, GA
S19	M	19	No	Ugrad	U. Stud,	fairly	daily	several times a week	No	WP, GA

Experiment 4

Cognitive Battery Scoring: `(Right-Wrong)`

	Hidden Patterns	Number Comparisons
S01	122	18
S02	120	21
S03	37	22
S04	150	22
S05	112	16.5
S06	153	38.5
S07	118	30
S08	164	33
S09	93	27.5
S10	132	23
S11	116	21.5
S12	115	27.5
S13	101	30
S14	96	28.5
S15	46	21
S16	110	29.5
S17	88	28.5
S18	122	16
S19	60	18

Experiment 4

Accuracy

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	0	1	1	1	0	0	1	1
S02	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1	1
S03	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	0	1	1	1
S04	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1
S05	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1
S06	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1
S07	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
S08	1	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1
S09	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0
S10	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
S11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1
S12	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
S13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
S14	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	1
S15	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1
S16	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	0	1	1
S17	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
S18	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1
S19	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1

Time

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	41.97	33.7	40.03	84.36	33.31	41.33	16.38	19.41	18.7	18.41	16	28.48	36.83	52.22	101.58	39.31	45.61	41.99	25.31	71.92	38.66	30.3	33.23	28.41
S02	78.66	43.63	127.67	66.75	37.88	27.5	39.95	79.33	87.13	195.7	43.5	37.89	74.03	110.34	91.56	49.13	76.06	86.91	47.73	109.06	55.73	36.63	60.89	40.95
S03	38.56	41.61	42.33	30.94	46.61	32.7	15.36	18.23	18.34	19.45	24.33	11.77	22.38	35.58	19.97	19.52	36.02	24.97	35.13	32.58	45.25	31.92	33.41	79.98
S04	54.08	58.63	66.31	28.16	18.39	37.45	94.53	44.99	**60.58**	134.36	39.23	31.36	65.66	48.69	140.64	32.84	86.58	35.2	54.89	72.42	24.13	33.67	22.16	43.03
S05	69.13	54.55	15.47	17.39	14.14	13.63	17.52	12.67	15	8.03	8.56	9.03	16.56	23.58	23.41	27.61	29.33	18.44	19.56	60.08	44.25	16.44	21.7	23.88
S06	45.94	47.89	62.52	25.45	31.09	24.2	19.42	23.16	15.83	20.02	18.38	13.84	26.8	26.02	24.16	20.27	37.27	19.2	35.02	59.31	40.06	28.8	26.77	38.06
S07	41.95	33.92	8.34	20	24.2	24.08	16.69	20.36	29.95	50.33	9.16	22.47	17.5	17.45	12.89	11.77	21.33	18.28	17.3	93.63	58.2	19.64	58.75	27.72
S08	33.47	13.78	12.59	20.14	26.17	16.22	53	54.45	83.67	39.81	20.17	27.24	21.47	25.78	18.56	12.89	58.94	19.75	18.11	25.88	67.03	76.36	19.34	24.98
S09	65.55	32.5	79.03	30.2	54.25	32.3	61.31	42.38	59.13	31.63	32.77	22.34	27.05	36.14	19.36	18.94	24.16	29.56	35.66	154.88	36.03	29.25	38.75	70.2
S10	68.64	44.26	11.03	17.2	17.72	23.28	8.11	28.91	11.47	68.78	7.67	10.23	24.19	16.81	12.84	16.28	14.27	18.77	15.7	51.22	17.31	25.88	25.08	25.36
S11	86.03	43.63	18.69	28.97	42.47	22.7	47.89	39	100.64	76.81	14.24	34.39	29.64	31.94	37.05	31.38	55.06	14.67	31.53	28.7	72.92	55.45	23.91	26.92
S12	48.14	50.92	86.28	31.45	35.42	33.53	41.67	36.64	46.13	45.39	21.86	30.69	51.89	37.36	*78.52*	55.8	*71.88*	36.98	38.38	40.63	26.58	54	93.81	49.55
S13	25.41	*70.91*	299.89	24.11	*17.28*	37.06	23.78	60.02	81.67	65.36	15.06	48.86	34.34	21.8	49.28	17.45	25.53	35.95	39.02	32.64	60.2	58.61	26.27	90.03
S14	96.09	*50.5*	75.39	41.28	*28.38*	43.27	210.11	89.84	49.73	48.78	73	49.36	85.39	47.34	66.56	61.86	67.48	59.69	36.05	42.39	46.94	48.17	32.94	48.72
S15	34.88	51.05	51.47	40.35	34.27	43.77	46.87	49.4	42.65	33.55	45.63	25.03	42.2	67.28	47.4	44.77	161.43	79.89	32.86	40.18	42.28	55.8	33.13	61.89
S16	74.06	19.77	37.22	21.24	51.1	64.68	32.66	116.16	60.17	141.87	49.08	132.63	212.31	161.45	132.61	72.16	46.3	238.33	22.51	142.21	100.38	38.2	42.58	48.7
S17	42.41	29.94	27.77	25.16	18.72	28.31	15.63	20.47	20.41	11.38	14.08	37.31	23.39	37.33	31.89	16.98	30.28	33.91	20.59	29.48	33.34	27.66	24.2	31.7
S18	54.06	26	40.17	20.7	26.75	25.97	33.92	18.13	43.13	16.02	20.03	32.25	24.89	21.66	14.55	9.25	50.64	40.95	35.78	96.25	39.14	26.97	38.13	41.42
S19	71.75	53.91	133.13	37.03	68.45	68.77	29.13	29.2	34.3	52.66	32.08	19.25	44.63	47.31	30.06	24.63	49.42	34.56	40.31	33.28	24.52	24.42	28.66	38.06

Distance Navigated

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	268.3559	403.8965	520.4955	709.5374	412.162	502.3614	0	375.807	406.576	489.5748	0	349.2253	742.1511	525.1382	2201.296	576.1181	709.3588	557.8726	0	560.3884	434.8208	335.043	520.1912	336.4455
S02	859.1707	0	1076.403	861.0768	506.1295	594.181	879.29	1021.729	1445.524	1723.713	735.6204	658.9544	2080.157	1849.964	1680.244	1087.356	1108.531	2173.484	798.6227	1090.347	398.3737	527.1777	1024.217	557.214
S03	294.8982	644.8563	457.5338	487.3863	507.2169	487.4613	315.4578	420.1897	316.4835	548.4392	262.137	0	511.5544	522.8415	382.7712	560.5847	595.8671	524.191	401.7873	360.6822	318.6055	548.6029	506.9748	709.4755
S04	448.1437	699.6414	788.7262	580.1103	411.1735	558.3188	1642.902	1030.605	1399.931	7631.058	705.9706	645.9992	1880.769	1160.73	3437.936	1040.355	2743.38	1052.332	564.6051	772.8358	369.293	342.5677	359.2786	532.7949
S05	136.056	626.4321	0	0	0	0	284.7818	0	0	0	0	0	369.2555	397.6133	373.5164	283.4092	488.0709	425.9334	0	440.1418	290.3466	0	0	261.9655
S06	367.8854	622.8312	527.1254	414.7159	537.641	462.9132	412.0318	412.8263	379.2825	464.2185	451.3804	427.8567	541.6969	503.4191	469.7042	549.3821	492.5309	442.2893	460.3619	657.5109	491.6231	521.209	325.6805	508.9642
S07	157.5005	479.111	0	506.6044	260.6112	476.7255	250.2593	271.469	404.7672	339.8076	0	250.05	355.102	412.1435	250.05	271.615	444.4106	448.7265	0	617.5951	329.5648	0	754.3621	337.6431
S08	203.2911	250.1243	0	343.8133	305.5907	336.1499	799.3085	1054.705	1233.87	940.9176	442.6549	511.7793	473.1217	513.6761	477.1694	546.6134	1056.141	557.3888	0	290.9943	727.5221	503.7668	0	250.05
S09	423.2949	577.7421	672.2033	517.5286	527.9857	482.4538	614.7023	515.2262	459.105	555.3052	402.9562	428.0166	502.9218	484.3643	386.3016	508.9261	489.1725	574.9685	481.0518	1447.884	361.4547	0	512.4144	516.2367
S10	379.023	0	0	0	0	387.6513	0	376.2711	0	697.9916	0	0	455.4114	341.6205	319.3108	282.8846	350.566	333.327	0	490.9659	0	0	0	0
S11	115.28	391.4261	0	0	349.2787	0	0	349.5392	791.0235	499.8747	0	401.8524	356.2698	0	487.0613	586.1004	627.2955	0	0	0	366.26	0	0	303.659
S12	319.1625	586.8014	611.6422	567.5023	521.4244	475.2648	562.1319	493.6266	549.0944	838.7338	250.05	442.0989	1021.493	865.2025	1258.25	1068.401	889.295	869.2174	353.091	466.1022	340.3861	658.8933	921.6368	517.7719
S13	0	451.9183	2143.724	0	0	0	427.6482	595.6252	958.6298	911.1828	309.8815	548.529	500.8577	486.7772	1003.294	509.0686	597.8428	591.188	0	0	390.0799	571.3304	0	1154.649
S14	489.7153	604.5823	595.5595	554.6237	324.2979	559.3009	2760.123	1438.144	801.9016	806.5303	1452.392	803.8096	1626.467	1022.34	1468.514	1356.752	1868.178	1476.133	291.3892	670.0114	439.9853	608.0137	511.3192	744.9601
S15	171.1272	318.6488	721.1412	385.2761	338.2601	467.1536	455.575	749.8021	700.6609	714.9654	632.2129	399.853	492.1261	1249.052	1235.526	1556.301	2754.024	1607.755	334.428	318.1244	270.1896	294.5692	339.734	438.1712
S16	333.9798	456.5983	297.0709	301.2054	565.6309	496.4291	287.3842	783.3332	592.0191	779.8262	351.421	577.9444	1327.371	887.1875	868.6456	1195.117	1054.646	2112.606	0	939.3366	525.8822	574.9009	483.0721	258.1677
S17	170.8665	332.2587	0	0	0	0	289.9312	0	0	0	0	0	364.1963	788.4598	593.2225	466.4274	575.7351	699.1122	0	0	432.2055	0	0	309.1574
S18	391.045	554.7696	754.7205	524.8938	513.2393	479.3736	603.869	430.0687	713.2391	493.6599	511.3727	827.9871	285.8435	401.3999	250.05	0	1527.429	1089.138	527.2497	901.4718	458.8503	0	434.5042	562.8828
S19	298.9483	506.2782	798.9115	493.9003	482.4888	476.2905	370.3685	295.3986	325.842	878.7585	380.9793	423.0034	353.3426	620.2999	364.7655	510.2113	733.048	0	358.7075	0	264.2007	0	274.3675	281.5725

Experiment 4

Satisfaction

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	3	5	4	3	5	5	6	5	6	5	6	5	6	5	3	6	5	4	6	4	5	6	6	5
S02	4	6	2	6	6	6	5	6	5	6	6	6	4	5	4	6	4	6	4	6	6	6	2	6
S03	5	4	3	6	6	3	6	3	3	5	6	2	5	6	4	6	4	6	6	5	5	6	6	4
S04	3	5	3	5	6	5	4	3	4	5	4	5	4	4	3	5	4	4	5	5	6	5	4	5
S05	5	4	5	4	4	4	5	5	5	5	5	4	5	5	5	5	4	5	4	4	4	5	4	4
S06	3	5	4	5	4	3	3	3	4	3	4	2	2	3	4	4	5	2	3	5	3	5	3	5
S07	5	6	5	6	6	5	6	5	6	6	6	5	5	6	6	6	6	4	6	6	6	6	6	6
S08	4	4	4	5	5	5	3	4	2	3	3	3	3	3	3	3	3	3	5	4	5	5	5	4
S09	3	5	3	5	5	4	5	6	5	4	4	5	2	2	5	2	2	4	5	6	6	6	6	6
S10	3	4	3	3	4	2	3	3	3	2	5	4	4	4	4	4	2	4	2	2	3	5	4	5
S11	4	4	5	5	4	5	6	5	5	5	5	6	5	4	5	5	5	5	5	5	5	6	5	5
S12	5	3	2	4	6	5	4	2	2	4	3	5	2	4	3	4	3	4	4	3	5	4	6	5
S13	1	1	2	4	4	2	2	2	2	3	4	3	2	4	3	1	4	3	4	2	2	2	4	4
S14	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S15	5	5	6	6	6	6	6	6	5	5	6	6	6	6	6	6	6	6	6	2	4	6	4	6
S16	6	6	6	6	6	6	5	6	5	6	6	6	6	6	6	6	4	6	6	6	6	4	6	6
S17	3	4	3	4	4	4	4	4	3	3	5	4	3	4	4	4	3	3	4	3	5	4	3	4
S18	5	5	5	5	5	5	3	4	4	3	6	6	4	3	4	4	4	4	4	3	5	5	5	4
S19	4	4	5	4	3	4	3	4	6	6	6	3	3	4	4	4	5	3	5	4	4	4	5	4

Difficulty

User	_t1	_t2	_t3	_t4	_t5	_t6	_t7	_t8	_t9	_t10	_t11	_t12	_t13	_t14	_t15	_t16	_t17	_t18	_t19	_t20	_t21	_t22	_t23	_t24
S01	2	2	2	4	1	3	1	2	1	2	1	1	2	2	4	1	2	2	1	3	2	1	2	1
S02	1	1	3	1	1	1	1	1	2	1	1	1	1	1	2	1	2	1	1	1	1	1	3	1
S03	2	3	4	1	1	3	2	2	4	3	1	3	2	2	2	1	2	2	1	1	3	2	1	3
S04	3	2	3	2	1	3	4	5	2	3	2	2	6	3	3	4	4	4	1	3	2	1	2	3
S05	3	4	5	2	3	4	3	5	4	4	4	3	3	5	4	4	6	4	4	3	2	3	3	2
S06	2	1	1	1	1	2	2	3	2	3	2	3	4	4	3	2	1	5	1	2	1	1	2	1
S07	4	1	3	2	1	2	1	2	2	1	1	2	2	3	2	2	2	3	2	2	3	1	1	1
S08	2	2	3	1	2	2	3	3	4	2	3	3	4	3	3	3	4	4	2	3	2	1	2	2
S09	4	2	3	1	3	2	2	2	2	3	2	1	4	5	2	4	4	3	1	1	1	1	1	1
S10	4	2	3	2	2	1	2	3	4	5	1	3	2	3	2	2	2	3	1	3	2	1	3	2
S11	4	4	3	2	4	3	1	2	2	2	2	2	2	4	2	3	4	3	3	3	3	2	2	4
S12	2	3	4	2	1	2	4	4	4	4	4	2	2	3	3	2	4	3	2	3	2	2	1	2
S13	5	6	5	1	1	5	4	2	1	4	1	1	1	1	1	5	4	2	1	5	4	1	1	1
S14	4	3	4	3	3	2	2	2	2	3	2	1	3	3	3	3	2	2	3	3	4	2	2	3
S15	2	2	1	1	1	1	1	2	2	3	1	1	1	1	1	1	2	1	1	3	2	1	3	1
S16	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	3	1	1	1	3	3	1	1
S17	2	2	4	2	3	2	3	3	3	3	2	2	3	3	3	3	3	3	2	5	1	2	1	2
S18	5	2	2	2	2	3	2	4	3	6	1	1	3	4	2	3	2	4	2	4	2	3	2	3
S19	5	4	2	3	3	3	2	2	6	6	1	3	1	3	4	2	4	1	3	2	5	4	2	3

F.3 Descriptive Statistics

Layout Technique: S = Semantic, P = Proximity
Connectedness: L = Line, P = Polygon, S = Semitransparent Polygon

ACCURACY

Descriptive Statistics - Overall

	Mean	Std. Deviation	N
PL	.8553	.1268	19
PP	.8684	.1742	19
PS	.8158	.1834	19
AL	.8684	.1529	19
AP	.7763	.1147	19
AS	.8421	.1493	19

TIME

Descriptive Statistics – Overall

	Mean	Std. Deviation	N
PL	44.4375	21.9615	19
PP	48.5082	20.2622	19
PS	51.8392	26.4041	19
AL	39.3446	18.0511	19
AP	36.8775	13.9836	19
AS	39.3780	22.7128	19

NAVIGATION DISTANCE

Descriptive Statistics - Overall

	Mean	Std. Deviation	N
PL	468.3540911	346.3424278	19
PP	554.9855718	231.9598729	19
PS	607.3747199	365.2885310	19
AL	592.1635221	506.3736734	19
AP	520.0714794	305.8057755	19
AS	513.9004176	251.6761923	19

SATISFACTION

Descriptive Statistics - Overall

	Mean	Std. Deviation	N
PL	4.2763	1.0405	19
PP	4.3289	.9540	19
PS	4.2895	.9252	19
AL	4.6842	.9311	19
AP	4.6842	.6762	19
AS	4.5132	.9409	19

DIFFICULTY

Descriptive Statistics - Overall

	Mean	Std. Deviation	N
PL	2.3553	.7087	19
PP	2.6711	.8039	19
PS	2.6316	.7138	19
AL	2.2895	.7181	19
AP	2.1053	.6837	19
AS	2.2237	.6867	19

G. Post-hoc Analysis

of Experiments 3 and 4

ACCURACY

Descriptive Statistics - Overall

	Display Context	Mean	Std. Deviation	N
HIGH	**OL**	.84615385	.14571882	13
	VD	.84210526	.16052272	19
	Total	.84375000	.15226780	32
LOW	**OL**	.78846154	9.3883452E-02	13
	VD	.76973684	9.5589889E-02	19
	Total	.77734375	9.3833968E-02	32

TIME

Descriptive Statistics - Overall

	Display Context	Mean	Std. Deviation	N
HIGH	OL	86.06442308	17.24699816	13
	VD	50.17414474	20.63265684	19
	Total	64.75457031	26.13540422	32
LOW	OL	80.67903846	33.74518742	13
	VD	38.12802632	15.91102342	19
	Total	55.41437500	32.22763046	32

NAVIGATION DISTANCE

Descriptive Statistics - Overall

	Display Context	Mean	Std. Deviation	N
HIGH	**OL**	863.98311731	239.35984085	13
	VD	581.18014868	272.29751620	19
	Total	696.06885469	291.79483226	32
LOW	**OL**	1151.11242596	792.79899308	13
	VD	516.98595921	267.33592777	19
	Total	774.59983633	620.42211609	32

H. Digital Resources

File	Description
irve.dtd	A description of XML tags and attributes for the IRVE display components developed for this work
irve.xsd	A description of XML elements and types for the IRVE display components developed for this work
PathSimv2.avi	A video demonstrating the PathSim Visualizer IRVE interface to results of a large-scale agent-based simulation
O.avi	A video demonstrating the Object Space technique used in Experiment 1
V.avi	A video demonstrating the Viewpoirt Space technique used in Experiment 1 on the single-screen condition
O_S_n.avi	A video demonstrating the ScreenBounds technique with no annotation scaling (Experiment 3)
O_S_p.avi	A video demonstrating the ScreenBounds technique with periodic annotation scaling (Experiment 3)
O_S_c.avi	A video demonstrating the ScreenBounds technique with continuous annotation scaling (Experiment 3)
O_F_n_avi	A video demonstrating the ForceDirected technique with no annotation scaling (Experiment 3)
O_F_p_avi	A video demonstrating the ForceDirected technique with periodic annotation scaling (Experiment 3)
O_F_c_avi	A video demonstrating the ForceDirected technique with continuous annotation scaling (Experiment 3)
V_S_l.avi	A video demonstrating the HUD BorderLayout technique with line connector (Experiment 4)

V_S_s.avi	A video demonstrating the HUD BorderLayout technique with semi-transparent polygon connector (Experiment 4)
V_S_p.avi	A video demonstrating the HUD BorderLayout technique with opaque polygon connector (Experiment 4)
V_P_l.avi	A video demonstrating the HUD Proximity BorderLayout technique with line connector (Experiment 4)
V_P_s.avi	A video demonstrating the HUD Proximity BorderLayout technique with semi-transparent polygon connector (Experiment 4)
V_P_p.avi	A video demonstrating the HUD Proximity BorderLayout technique with opaque polygon connector (Experiment 4)