The Effects of Task, Task Mapping, and Layout Space on User Performance in Information-Rich Virtual Environments

Nicholas F. Polys npolys@vt.edu
Lauren Shupp lshupp@vt.edu
James Volpe jvolpe@vt.edu
Vladimir Glina vglina@vt.edu
Chris North north@vt.edu
Virginia Polytechnic Institute and State University
Department of Computer Science
Blacksburg, VA 24061

ABSTRACT

How should abstract information be displayed in Information-Rich Virtual Environments (IRVEs)? There are a variety of techniques available, and it is important to determine which techniques help foster a user's understanding both within and between abstract and spatial information types. Our evaluation compared two such techniques: Object Space and Display Space. Users strongly prefer Display Space over Object Space, and those who use Display Space may perform better. Display Space was faster and more accurate than Object Space for tasks comparing abstract information. Object Space was more accurate for comparisons of spatial information. These results suggest that for abstract criteria, visibility is a more important requirement than perceptual coupling by depth and association cues. They also support the value of perceptual coupling for tasks with spatial criteria.

Keywords

Information-Rich Virtual Environments, Information Visualization, Desktop Virtual Environments, Object Space, Display Space, chemical informatics, molecular graphics.

INTRODUCTION

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.

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 they can use those relations to understand patterns or trends within and between the respective data types?

Our evaluation compares two display techniques: Object Space and Display Space. In the Object Space technique abstract information is always linked to the object it describes, even if the object is moved. In the Display Space technique a set number of pixels are allocated to displaying abstract information (Figures 1 and 2).

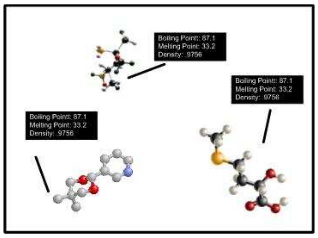

Figure 1: Object Space technique.

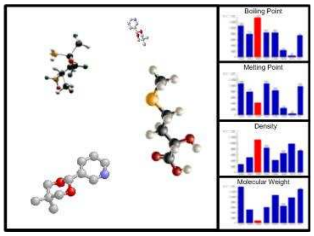

Figure 2: Display Space technique.

We sought to compare the two techniques further by asking two underlying questions. Is one display technique better for search or comparison tasks? Is one display technique better for tasks where the user is given spatial information and has to find abstract information or vice versa? In order to answer these questions, we made each a factor in the experiment, task type and task mapping respectively.

We chose to model an animal cell for the 3D environment. Molecules were placed within this environment and were the focus of user tasks. This domain is appropriate for this evaluation for many reasons:

- Molecule models are an excellent example of objects that combine both perceptual (spatial) and abstract information.
- 3D molecule models are widely available [6].
- Molecules are naturally represented in a biological context.
- A virtual cell environment requires navigation in all three dimensions.
- The location of objects and regions in an animal cell are flexible.
- Researchers have already observed that virtual environments in this domain have educational value [4, 13].

In our evaluation, we used the Cortona® VRML Client plug-in to view this 3D environment, and for the Display Space technique, we used the Snap-Together Visualization for viewing abstract information. This paper discusses advantages for both Display Space and Object Space techniques discovered by our evaluation.

RELATED WORK

This study was inspired by the Snap2Diverse project [10]. Snap2Diverse was a usability evaluation of the Display Space technique for IRVEs in the CAVE. The evaluation establishes that the technique is promising for conveying abstract information in VEs. Since the study used the Snap-Together Visualization system and chose molecules for its environment, our evaluation used a similar approach. Furthermore, it is clear from previous research that there is much interest in IRVEs, especially for scientific education.

Information Rich Virtual Environments

The need to visualize abstract information within VEs has lead to the development of IRVEs. Bowman, North et al. discuss how abstract information can be displayed on the screen in different locations (e.g. world-fixed, display-fixed, object-fixed, and user-fixed) [3]. Recent discussions have lead to a classification of techniques for designing a complete IRVE interface. The following are such display techniques: Object Identity, Viewport Space, World Space, User Space, Display Space, and Object Space. We want to extend this work by discovering the benefits and weaknesses of at least two of these techniques.

	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: the orthogonal Layout Space and Association dimensions in IRVE design

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. Depending on the IRVE display technique used, each of these spaces may present a range of depth cues that are consistent between the annotation and referent. In addition to the various depth cues that may be provided by a Layout Space, we can also consider the relation of object and annotation on the image plane in terms of Gestalt Association cues. Thus we have a 2 dimensional design space for Layout Attributes, shown in Table 1. High Association and High Occlusion reside in the top left corner. Low Association and Low Occlusion reside in the lower left corner.

Multiple Views in Visualizations

Multiple views are valuable for visualizing data [1, 5, 8], and interaction strategies that link related data between views can be beneficial. North defines tight coupling between views as constraints between related data marks in the views and offers a classification of tight coupling types defined through two main classes of user interface actions: selection and navigation. North also explains how various types of tight coupling are realized with the Snap-Together Visualization system [7]. We chose to use the Snap-Together Visualization system for the Display Space portion of our evaluation because the open-source tool not only makes organizing multiple visualizations simple and quick, but it has potential for advanced interaction strategies.

Polys, Moldenhauer, et. al. illustrate the importance of linking abstract information to objects in VEs as well as the need for visualization choices (e.g. text, graphs) [10]. This supported our decision to use the Snap-Together Visualization tool, since it allowed us to design our interfaces with these key concepts in mind. We chose to use both bar charts and a table for displaying abstract information in the Display Space technique.

Educational Value of Virtual Cell Environments

In recent years, VEs have become a popular and valuable educational tool. Bowman, Hodges et. al. found trends showing that students who use educational VEs perform better than those who only attend lectures [2]. McClean et. al. also report that augmenting science lectures with virtual environments for biology and possibly other disciplines, namely geology, can positively impact student learning [4]. White et al. stress the flexibility VEs offer for the experimental component of education and introduce the Virtual Cell – a VE which teaches fundamental concepts of cell biology [13]. Because there is a growing appreciation for IRVEs that model biological environments, we are confident that our work will benefit future research.

Chemical Informatics / Information Enhancement

Those who study chemical informatics and molecular visualization use computers to analyze chemical information, which often includes 3D molecular modeling. The Chemical Markup Language (CML) is a method of storing structural and molecular information together. CML provides the frame work for visualizing the spatial and abstract information needed to build the molecular compounds displayed in an IRVE. We used CML files provided by the NIST database to construct the virtual environments in our evaluation [6].

Polys explains that IRVEs can be constructed using Extensible Stylesheet Transformations (XLST) to transfer CML into X3D [11]. The Snap2Diverse project was implemented using XLST [10]. In our evaluation, we designed our VEs using the same method.

The VEs we developed provided a continuous viewpoint that allowed users to navigate around the molecules. Vazquez et. al. explain how different viewpoints from fixed locations have unique advantages [12]. One viewpoint may not be as advantageous for one task as it would be for another. Since our VE gave users the freedom to explore molecules from every angle, users were not restricted to a fixed viewpoint for all tasks.

While some work has been done to understand how 3D models should be presented, more work is needed to critically evaluate how abstract information should be incorporated.

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 occlusion-association 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 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 an empirical usability experiment, which is summarized in the following sections.

Object Space vs. Display Space

This experiment was a class project for Dr. Chris North's graduate class in Information Visualization, which was run by Shupp, Volpe, Glina, and Polys in 2004. 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 1). 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.

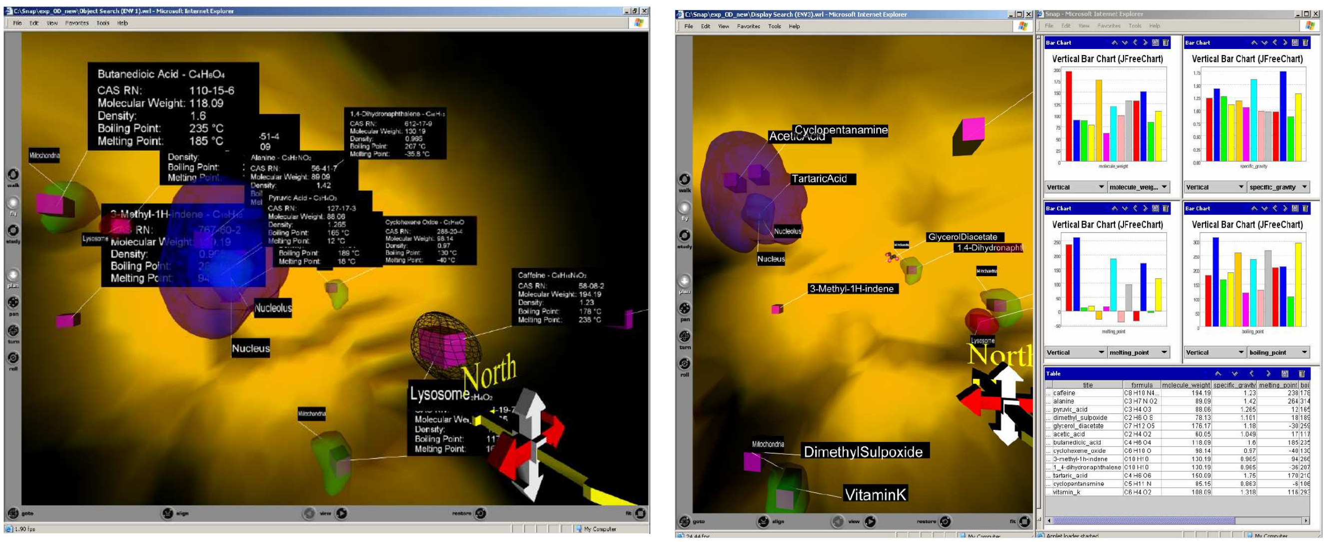

Figure 1 Object Space vs. Display Space

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 our conditions are:

- A single view visualization where all detail information is distributed in the virtual environment
- Provides 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 our conditions are:

- 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 - low occlusion, low association

USER STUDY

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.

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 2).

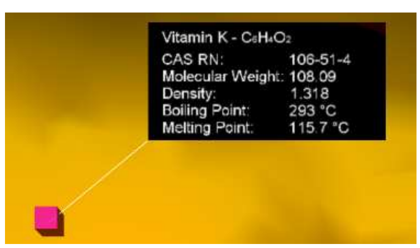

Figure 2: The Vitamin K molecule at a distance

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 wireframe switch was essential to let users select objects inside other objects in order to minimize their annotation (e.g. Figure 3). All of the aforementioned design decisions were explained in participant training.

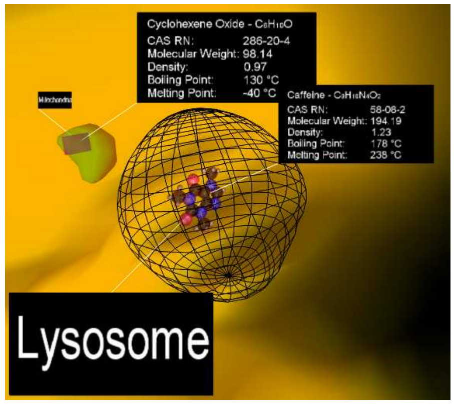

Figure 3: Close-up of Caffeine in a wire framed lysosome and an opaque mitochondria containing a landmark for Cyclohexene Oxide in the background

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 was 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 2. 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	None
Occlusion	O	O	O	
Motion Parallax	O	O	O	
Relative/Size / Perspective				
None				D

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

Tasks

In order to test how our IRVE layout techniques impact usability for search and comparison, we define 4 kinds of tasks (below, Table 3). 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?'
- 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?’
- 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?’

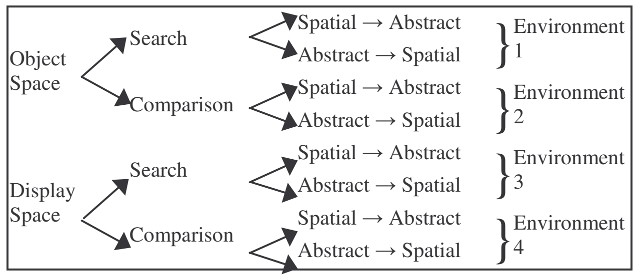

Table 3: Task Structure in the Object vs. Display Experiment

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 were counterbalanced and shuffled to eliminate the possibility of memory for previous locations or attributes (Table 3).

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 an aquarium 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.

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. But, users are on average more accurate in Display Space for spatial to abstract (S->A) search tasks (75%) than in Object Space (60.4%).

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 4 depicts this relation.

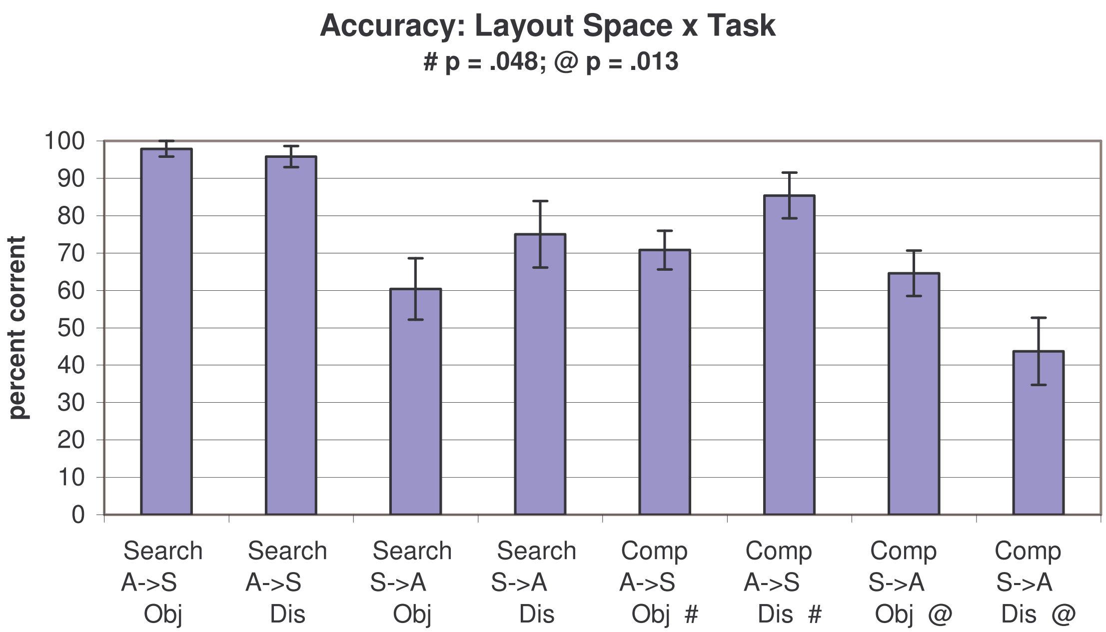

Figure 4: Average accuracy of the eight conditions.

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 5 shows this relationship.

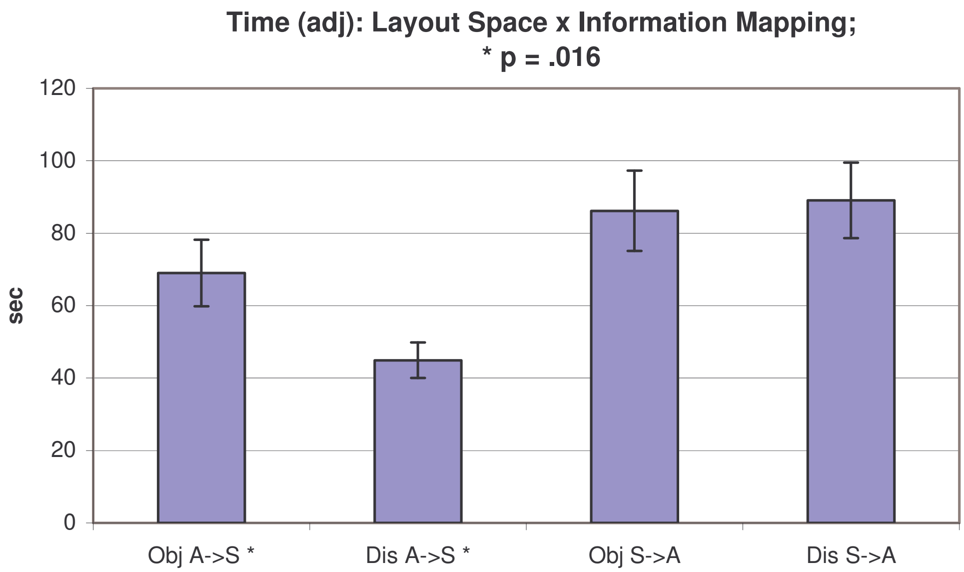

Figure 5: Average adjusted time for display technique and task mapping

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. 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.

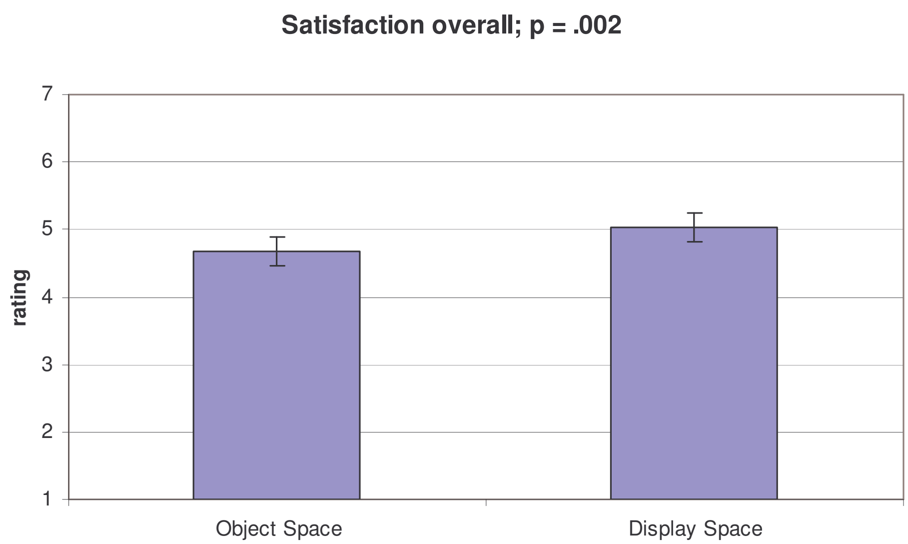

Figure 6: Average satisfaction rating for display techniques.

There was also a significant interaction between display technique and task mapping for participant satisfaction (F(1,15) = 5.971; p=0.027), shown in Figure 7. Pairwise t-tests reveal that the significant difference is on abstract to spatial tasks (A->S) where Display Space (5.5) was more satisfying than in

Object Space (5.7); t(15)= 3.525, p = 003. It does not appear that spatial to abstract (S->A) questions influence user satisfaction for either display technique.

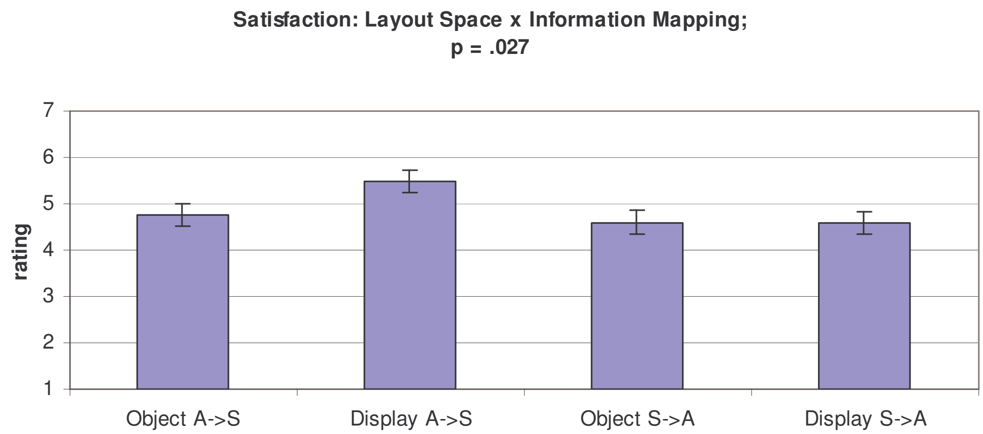

Figure 7: Average satisfaction rating for display technique and task mapping.

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 8.

Pairwise 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.

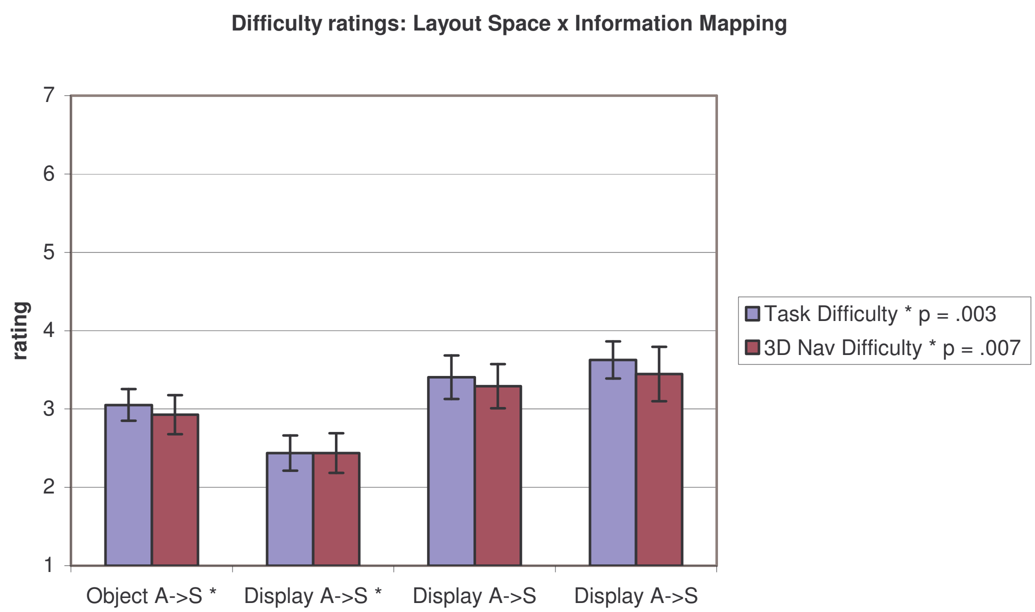

Figure 8: Average difficulty ratings for display technique and task information mapping

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 questions.

Although our experiment did not explicitly evaluate task-type and task information-mapping, 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.

CONCLUSIONS

In this experiment, we tested an Object space technique against a Display space technique and 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 2 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).

This evaluation and the prior Snap3Diverse Display space evaluation 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 visualizations- 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 need 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.

WORK IN PROGRESS

We are currently working on developing a Display Space interface that includes additional functionality. The technique is a great opportunity to integrate interaction strategies such as brushing and linking. This is yet another reason why we chose to use the Snap-Together Visualization.

To visualize virtual worlds developed with VRML technology and to enable communication between such worlds and other views, we need a browser capable of displaying VEs and working as a Snap component. Because Snap is implemented with Java, we had to find a means to communicate between VRML and Java. We chose the Xj3D toolkit because Xj3D is becoming a standard for rendering 3D graphics in Java. The toolkit package included a browser application which was to be converted into a Snap component applet.

We started by installing the Java3D API and copying Xj3D JAR files into the `/lib/ext` subdirectory of the main Java Runtime Environment directory. This is necessary to run the browser as an applet and eliminates the need to digitally sign the jars and write an installer for them.

Then we moved the browser functionality into a new class, `VrmlBrowser`. All the Snap components implement the `Snapable` interface (typically by inheriting from the `Snapplet` class which in turn inherits from `Snapable`). Our next step was to change the `VrmlBrowser` inheritance tree from the one shown in Figure 9 to the one in Figure 10, making it an applet.

The `start` method of this applet class uses the `vrml.eai.BrowserFactory.createVrmlComponent` method to instantiate a VRML browser. The last method instantiates the browser as an AWT component with a VRML display capability. This VRML browser runs as an applet which is the Snap component shown in Figure 11. Then the start method loads the virtual world corresponding to the URL specified in the browser address bar, adds a VRML event listener to the applet class, retrieves the names of the world scenegraph nodes, and saves these names in a hashset.

We provided two-way communication between our VRML browser and Snap so that user actions on objects in the virtual world would link to data items visualized by other Snap components and vice versa. There are VRML events and Snap events in the VRML browser. For each type, standard methods for adding and removing listeners and event firing are realized. The `load` method loads a new scenegraph and places a VRML event of the type `EventOut` in correspondence with every scenegraph node. In the `run` method, the message processing loop iterates through the nodes. For each VRML event to be fired, the `fireVrmlEvent` method calls the `eventOutChanged` method which produces and fires the corresponding Snap event. This allows for actions within the VE to be communicated to other views.

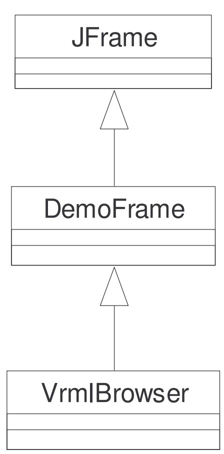

Figure 9: The old inheritance tree for VrmlBrowser.

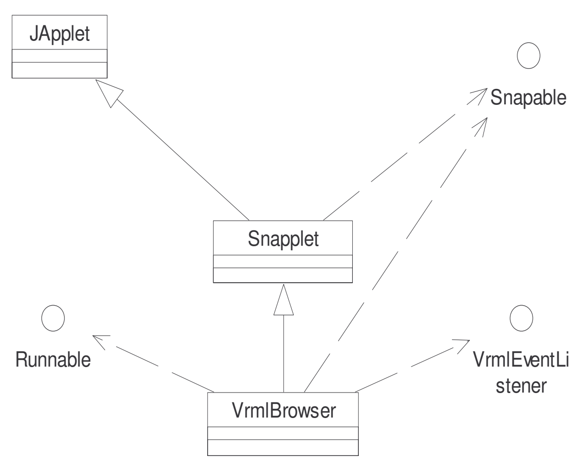

Figure 10: The new inheritance tree for VrmlBrowser

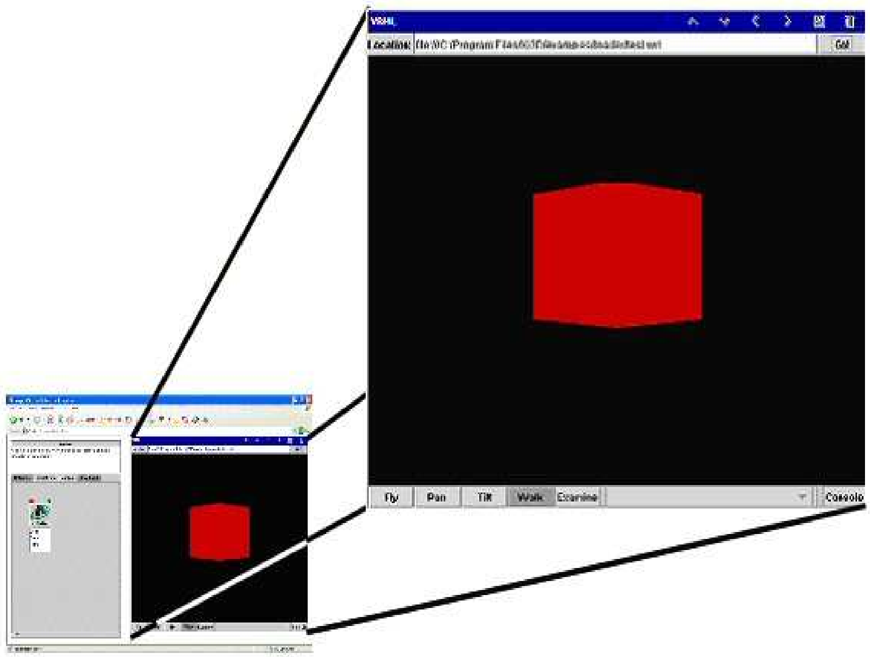

Figure 11: The Xj3D browser configured as a Snap component.

Snap also allows for actions in other views to be communicated to the VE. The VRML browser handles Snap events sent from other Snap components in the following way. All Snap components register with the Snap class called `CommandManager` by communicating what types of Snap events they want to receive. `CommandManager` acts as an event dispatcher. Within the `VrmlBrowser` class, the `performSnapEvent` method is responsible for processing the Snap events `VrmlBrowser` has registered for.

Currently, we have the Xj3D applet running as a Snap component and have the infrastructure ready for communication. We consider this work an important step towards future evaluations.

FUTURE WORK

The next step is to define communications between the Xj3D applet and other Snap views. Although the framework is in place to send and receive messages between views, it is not clear what messages should be exchanged. Once we understand what messages are to be exchanged, an interface can be designed to use interaction strategies (e.g. brushing and linking). For example, users could highlight a molecule in the VE and see the corresponding abstract information highlight simultaneously and vice versa.

We foresee two possible studies taking advantage of this tool. One evaluation could be similar to the one discussed in this paper by comparing an Object Space interface to a fully functional Display Space interface. The other could consider comparing display techniques using a head-mounted display or a high-resolution 3D environment like the CAVE.

ACKNOWLEDGMENTS

Special thanks to Bradley Vender from North Dakota State University and Alan Hudson from Yumetech for assistance with integrating the XJ3D browser as a Snap component. Thanks to the Visualization and Animation Group at Virginia Tech for use of their facilities and support in this project.

REFERENCES

1. Baldonado, M.Q., Woodruff, A.,and Kuchinsky, A. Guidelines for Using Multiple Views in Information Visualization. *Proceedings of the Working Conference on Advanced Visual Interfaces* (Palermo, Italy, May 24-26 2000) 110-119.

2. Bowman, D., Hodges, L., Allison, D., and Winemman, J. "The Educational Value of an Information-Rich Virtual Environment", *Presence: Teleoperators and Virtual Environments*, 8, 3 (June 1999), 317-331.

3. Bowman, D., North, C., Chen, J., Polys, N., Pyla, P., and Yilmaz, U. (2003). Information-Rich Virtual Environments: Theory, Tools, and Research Agenda. *Proceedings of ACM Virtual Reality Software and Technology* (Osaka, Japan, October 1-3, 2003) 81-90.

4. McClean, P., Saini-Eidukat, B., Schwert, D., Slator, B., and White, A. Virtual Worlds in Large Enrollment Science Classes Significantly Improve Authentic Learning. *Selected Papers from the 12th International Conference on College Teaching and Learning (ICCTL-01)* (Center for the Advancement of Teaching and Learning, Jacksonville, FL, April 17-21, 2001) 111-118.

5. Masui, T., Minckuchi, M., Borden, G., Kashiwagi, K. Multiple-View Approach for Smooth Information Retrieval. *Proceedings of the 8th annual ACM symposium on User interface and software technology* (Pittsburgh, PA, December 1995) 199 – 206.

6. NIST Chemistry WebBook. Available October 2004 at http://webbook.nist.gov/chemistry.

7. North, C., "Multiple Views and Tight Coupling in Visualization: A Language, Taxonomy, and System", *Proc. CSREA CISST 2001 Workshop of Fundamental Issues in Visualizations*, 2001.

8. North, C., Shneiderman, B. Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. *Proceedings of ACM Conference on Advanced Visual* Interfaces *(AVI '00)* (Palermo, Italy, May, 2000) 128-135.

9. Polys, N., Bowman, D., North, C. Information-Rich Virtual Environments: Challenges and Outlook. *NASA Workshop on the Knowledge Integrating Virtual Iron Bird* (Monterey, CA, April 2004).

10. Polys, N., Moldenhauer, M., Ray, A., Dandekar, C., North, C., Bowman, D. Snap2Diverse: Coordinating Information Visualizations and Virtual Environments. *Proceedings of SPIE Visualization and Data Analysis (EI10)* (San Jose, CA, Jan 18-22, 2004).

11. Polys, N. Stylesheet Transformations for Interactive Visualization: Towards Web3D Chemistry Curricula. *Web3D Symposium* (St. Malo, France, 2003).

12. Vazquez, P., Feixas, M., Sbert, M., Llobet, A. Viewpoint Entropy: A New Tool for Obtaining Good Views of Molecules. *Proceedings of the IEEE TCVG Symposium on Visualization* (Barselona, Spain, 2002).

13. White, A., McClean, P., and Slator, B. M. The Virtual Cell: an interactive, virtual environment for cell biology *Proceedings of the World Conference on Education Media and Hypermedia (ED-MEDIA 99)* (Seattle, WA, June 19-24, 1999) 1444-1445.