Snap2Diverse: Coordinating Information Visualizations and Virtual Environments

Nicholas F. Polys, Andrew Ray, Maxim Moldenhauer, Chetan Dandekar, Dr. Chris North

Virginia Polytechnic Institute and State University
Department of Computer Science
Blacksburg, VA

ABSTRACT

The field of Information Visualization is concerned with improving with how users perceive, understand, and interact with visual representations of data sets. Immersive Virtual Environments (VEs) excel at providing researchers and designers a greater comprehension of the spatial features and relations of their data, models, and scenes. This project addresses the intersection of these two fields where information is visualized in a virtual environment. Specifically we are interested in visualizing abstract information in relation to spatial information in the context of a virtual environment. We describe a set of design issues for this type of integrated visualization and demonstrate a coordinated, multiple-views system supporting 2D and 3D visualization tasks such as overview, navigation, details-on-demand, and brushing-and-linking selection. Software architecture issues are discussed with details of our implementation applied to the domain of chemical information and visualization. Lastly, we subject our system to an informal usability evaluation and identify usability issues with interaction and navigation that may guide future work in these situations.

Keywords: Information-Rich Virtual Environments, Information Visualization, Immersive Systems, Molecular Graphics

1. INTRODUCTION & BACKGROUND

In many cases of data and domain, the nature of the data - especially its semantics and dimensionality - may determine the most 'appropriate' rendering and interaction scheme (2D or 3D). In modern design and research applications however, a significant proportion of data concerns geometrical and structural information as well as abstract or quantitative data associated with that geometry. For example, take a case in the domain of chemical engineering and biomedical design. Chemical data is multi-faceted and heterogeneous: structural, nominal, ordinal, and quantitative data (such as their physical characteristics) are all associated to molecules and their constituent atoms and bonds.

Established information visualization techniques for 2 dimensional displays include textual/tabular renderings, Scatter plots, Bar charts, Line graphs, Pie charts, etc. and each may be useful to show different qualities and relationships in the dataset [Card et al. 1999]. In chemistry and biomedical applications, there is a growing body of data on molecular compounds: their physical structure and composition, their physical properties, and the properties of their constituent parts. When considering information *about* a molecule or its parts, such established visualization techniques can be used. For example, names and numbers can be displayed as textual tables or Pie charts, and spectral data with Bar charts or Scatter plots.

In these contexts however, inherently spatial attributes of the data (such as the structural relationships of atoms and bonds in a molecule) are difficult to understand. Immersive 3D views of this spatial data may be desirable or required. In order to understand and integrate both spatial and abstract types of data, they must be simultaneously available to the user. Thus, we identify the need for the immersive visualization of interrelated spatial and abstract information. That is, a 3D virtual environment where objects and the environment are 'enhanced' with displays of their related abstract attributes.

This type of virtual environment has been generally described as 'Information-Rich" by Bowman et al [Bowman et al. 1998; Bowman et al. 1999]. An Information-Rich Virtual Environment (IRVE) is a virtual environment that is enhanced with multiple information types embedded within its spatial view construct. As users navigate within the spatial environment (the world and objects), they may need or want access to related information about the objects or world. This abstract or symbolic information could include text, links, numbers, graphical-plots, and audio/video annotations.

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 information-rich virtual environments as well. Such tasks have been improved through the use of tightly-coupled, multiple view visualization techniques [North & Shneiderman 2001, 2000]. We believe these qualities of parallelism of display and coordination of views in IRVEs can also give users easier, integrated access to information and help them to generate insight through stronger mental associations between physical and abstract information.

There are a number of possibilities as to how abstract information is related to the spatial information of an Information-Rich Virtual Environment. The first is where abstract information is embedded/associated with points/regions in spatial data; for example, a numerical description of an atom or molecule, or a text description of an organ of the human anatomy. These can take the form of visual items like pop-up labels or hyperlinks where the abstract data is related to localized objects in the space. Another possibility is that abstract information varies continuously across the space. This is like scientific visualization or census data where the measurements are a function of the space - the abstract data is structured by the spatial data.

A third possibility is when the structure of the spatial data and the structured abstract data are mutually interlinked; for example, a tree-structured directory links to locations in a building, and vice versa. Here, the abstract data has a structure of its own (e.g. a tree). Unfortunately, there is a lack of documented techniques to embed tightly coupled 2D visualizations in 3D immersive environments. This paper explores the latter case using the Coordinated Views approach to link structural visualizations in a virtual environment with visualizations of abstract data about those structures and their constituent parts.

Our system uses the DIVERSE toolkit [Kelso et al. 2002] and Snap-Together Visualization [North et al. 2002] for integrated visualization tasks such as overview, navigation, details-on-demand, and brushing-and-linking selection. It can be run on a desktop with 2D and 3D views in separate windows, but we are more interested in the use of coordinated 2D visualizations inside an immersive 3D world such as the CAVE. In a CAVE, input and output requirements are different. We demonstrate a system where users can visualize and interact with multiple information types by way of a 'Hanging Picture' window superimposed on the immersive virtual world (Figure 1).

Figure 1. CAVE user navigates to an atom and its bonds which are highlighted as a result of selection in the 2D Hanging Picture

2. RELATED WORK

While prior work has elucidated design criteria for 2D or 3D information visualizations, little of this work addresses the combination of *embedded*, *coordinated* 2D visualizations in an immersive environment. The challenge of this project is how software and communication architectures can work together to support coordinated displays and interactions across rendered dimensions. Our work extends previous research by having 2D and 3D views of a complex data set linked across web based Snap-Together Visualization system and the DIVERSE virtual reality software platform.

2.1 Information Visualization

A growing body of work is leveraging object-oriented software design to provide users with multiple views or renderings of the same data such as table view, a Histogram view, or a Scatter plot view. North [North 2001] has described a taxonomy and system for tightly-coupled views which allows users to build their own coordinated visualizations. These visualization 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 has shown a significant speed up on Overview+Detail tasks [North & Shneiderman 2001, 2000]. 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 and still communicate in a structured way.

Baldonado, Woodruff, and Kuchinsky [Baldonado et al. 2000] have proposed guidelines for building multiple view visualizations. They claim four criteria regarding how to choose multiple views: diversity, complementarity, parsimony, and decomposition. As well, they put forward four criteria for presentation and interaction design: space/time resource optimization, self-evidence, consistency, and attention management. While these guidelines are well-formulated for 2D media, none have been critically evaluated in the context of 3D worlds which we propose.

2.2 Immersion and 3D Visualizations

Our DIVERSE application supports both 3D visualization on the desktop, and 3D immersive display in the CAVE. Our motivation for immersive visualization of molecular structure is supported by previous research. According to Ware and Franck [Ware & Franck 2000], the graph that can be understood with a head-coupled stereo view is three times as large as the 2D graph for any given error rate. Also, Datey [Datey 2002] shows that visualization of inherently spatial data proves better in immersive VEs. Because our molecular data is both graphical and spatial in nature, immersive environments seem ideal.

Bowman et al [Bowman et al. 1998] have implemented and evaluated an information-rich 'Virtual Venue' with a number of features that are common to our project such as navigation (constrained travel) and 'embedded' information retrieval through menus and spatial hyperlinks, as well as audio cues and help. The use of menus in immersive systems has also been investigated showing a good translation of the 2D metaphor to 3D environs. Bowman and Wingrave have compared Pinch Glove, floating menu, and pen and tablet menu systems. Pen and tablet interaction was significantly faster, but users preferred the Pinch Glove system [Bowman and Wingrave 2001].

While design principles for interactive techniques have been described in terms of performance and naturalism [Bowman 2002], more work is needed to determine how these techniques can be applied to embedded, coordinated visualizations and if naturalism and performance are affected by issues of display form, interactive context, and their different combinations.

3 DESIGN

This paper describes our work to address the 2D-3D Coordinated View problem in the context of a molecular visualization application. We wanted to develop a system for integrating multiple views of heterogeneous physical and abstract data and identify usability issues with these displays in immersive environments. While this framework for coordinated viewing is applicable to a variety of data domains such as medicine, engineering, and architecture, we chose to use a set of Chemical Markup Language (CML) files since we had some experience with the language and the variety of data captured in the files exemplifies the combination of spatial and abstract information types.

Chemical Markup Language is a new approach to managing molecular information. It has a large scope as it covers disciplines from macromolecular sequences to inorganic molecules and quantum chemistry. CML is new in bringing the power of XML to the management of chemical information - its design supports interoperability with the XML family of tools and protocols. CML contains a number of kinds of information about a particular compound, from its physical atomic structure and elemental makeup to its water solubility, melting point, atomic weight, and various spectral descriptions. These are manifested by variety of data-types in tags describing structural, numerical, and meta data about the chemistry of the molecule. CML provides a base functionality for atomic, molecular, and crystallographic information and allows extensibility for other chemical applications.

We use the Snap-Together Visualization system to display abstract data about a molecule and its atoms through 2D visualizations such as Scatter plots and Data tables. 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 Data tables, Scatter plots, Bar charts, and others. 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 for these reasons of end-user flexibility and the ability for developers to define new visualization components.

We use the DIVERSE virtual reality toolkit to display inherently spatial data such as the 3D molecular structure. A considerable amount of work has been done in the field of visualizing chemical compounds with several types of visualizations implemented in 3D and immersive environments. Our project combines a web based visualization system with the 3D immersive view that the CAVE provides. 'A Functional Framework for Web-Based Information Visualization Systems.' [Bender et al. 2000] addresses 3D molecular visualization applications and gives a

good overview of the ways molecules can be visualized; our DIVERSE application, s_AtomView, uses the 'ball and stick' 3D display for atoms and bonds in molecules.

In order to give users simultaneous visual access to the spatial and abstract information, we demonstrate what we call 'The Hanging Picture Metaphor'. In this design, the Snap application window is 'hung' on a wall of the CAVE. The contents of the picture are the 2D visualization components setup through Snap's visualization schema. The Hanging Picture is different from common Heads-Up-Displays in that it is a separate, opaque application and does not reside within the scenegraph. Our DIVERSE application also supports a Heads-Up-Display where overview information can be displayed; however, this display venue is semi-transparent and is not pickable.

A principal advantage of Snap and the Hanging Picture is that the picture is customizable and interactive. Users can compose tightly coupled visualization components and coordinate selection events for brushing and linking interaction. When a CAVEAdapter is connected into the Snap visualization schema, select events can be routed to the DIVERSE system and the 3D objects corresponding to that selection are highlighted as shown in figures 1 and 2.

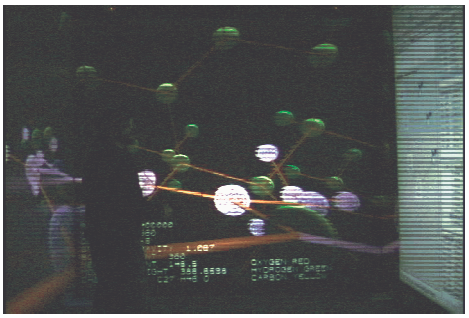

Figure 2. CAVE user explores the properties and relationships of Carbon atoms in a Histamine molecule

Coordinating 2D and 3D views in a virtual environment not only requires the sharing of addressable data objects and event communication across the applications but also special interaction facilities for immersive displays such as our CAVE. User input in CAVE systems is typically through the use of a 6 degree-of-freedom tracked 'wand' which has a number of buttons and a thumb-size joystick. We used the XWand software developed here at Virginia Tech to manage navigation interactions in the virtual environment and mouse emulation for selection and dragging in the Hanging Picture.

Tasks and scenarios we wanted to address concern how well users can locate, relate, and understand information across the 2D visualizations and the 3D world. Some example tasks a researcher or analyst may ask our system to support are shown in Table 1.

Exploring
Load molecules into the 3D View. 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)?
Finding
How many Carbons are there in Caffeine? How many of these have double bonds? Is the 8th Carbon of Caffeine in a ring or an arm?
Comparing
Which compound has more double bonds, Caffeine or Histamine? Which has more Nitrogen atoms? Which is heavier?
Pattern Recognition
What are the similarities and differences between Caffeine & Histamine molecules?

Table 1. Supported tasks and sample questions

4. ARCHITECTURE

The principle technologies used in our implementation (Figure 3) are described below.

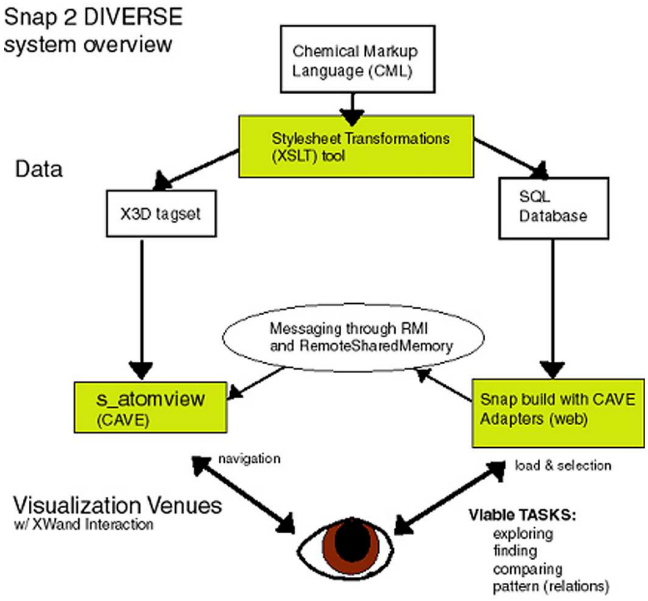

Figure 3. System Architecture

4.1 Data Sources

Murray-Rust et. al. have described the basic concepts of CML in their paper [Murray-Rust & Rzepa 1999]. In addition, they worked on a project culminating in the first fully operational system for managing complex chemical content entirely in interoperating XML-based markup languages [Murray-Rust et al. 1999]. This involved the extension of CML 1.0 and the development of mechanisms allowing the display of CML marked up molecules, spectra and reactions within a standard web browser.

The developers of CML have provided a DTD, Schema, and an API toolkit (CMLDOM). In addition, they provide Jumbo, a Java-based application to load and view a number of chemical formats in CML (CIF, MDL, MOL2, PDB, XYZ JMOL) and write them out to any of those formats. Given the capability and flexibility of CML to describe molecular information and the wide support for XML tools, we chose to source our data in this format. We began our information visualization project with a set of Chemical Markup Language (CML) files for common organic compounds such as Caffeine, Histamine, Cholesterol, and Digitoxigenin from the CML website.

Because CML is a dialect of XML, we are able to use Extensible Stylesheets (XSLT) to transform the single source data file to multiple representations [Polys 2003]. Snap requires its source data to be a relational database. Therefore we had to generate normalized, relational tables from the hierarchical CML data source. Additionally, because the DIVERSE AtomView application was written to load files into the 3D world and had little database capability, we needed to generate simple X3D files containing the molecules' geometry, atoms, and bonds and some basic text information. In order to accomplish this, we built a simple Java tool to easily transform different source files with different XSLT stylesheets.

We maintained consistency of the data items by using the CML id attribute to guarantee uniqueness of the elements across representations. Building the Snap relational database from a hierarchical data structure presented some challenges. First, we examined the semantics of the data structure in the CML file and determined that multiple tables were needed such as: molecule, atom, bond, atomic_constants, and spectral data. For each table in the database, we created an XSLT stylesheet to transform the appropriate CML tags to a comma-separated file (csv) which could be loaded into MS Access and then into SQLServer.

While building the database, we discovered some limitations in the data model for Snap that have to do with join operations. Consider atoms and bonds for example; a bond connects two atoms but from our available data, it does not have a directionality associated with it. For a Snap event to connect an atom and its bonds or a bond and its atoms,

we had to add an intermediary, redundant 'bond2way' table even though it makes no sense from the CML data model to claim that one is the start_atom and one is the end_atom.

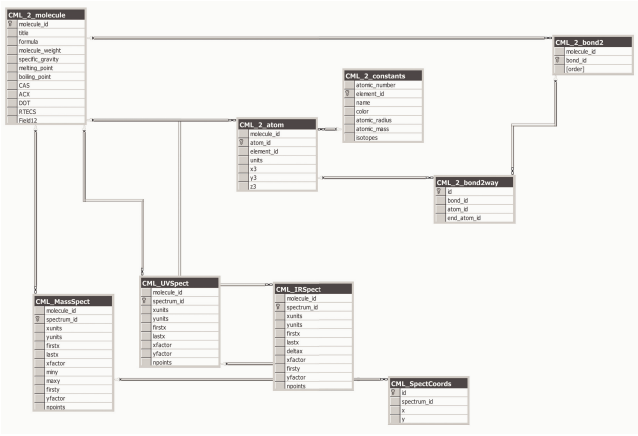

Figure 4. SQL Relationships in Molecules.dsn

Once the data was in the SQL server and the table relationships specified (see Figure 4), it could be loaded by a Snap application. For this project, we also had to build a set of Snap Adapters to process events to the virtual world. Because Snap currently has no event mechanism to declare from which visualization component an event originated, we built the adapter to actually manifest three adapters in the visualization schema: one for a molecule-visualization event, one for an atom visualization event, and one for a bond-visualization event.

4.12 Extensible Markup Language (XML) and Extensible Stylesheet Transformations (XSLT)

Numerous developer resources exist for the World Wide Web Consortium's XSLT specification [W3C] so we will focus on the design issues specific to CML and transformation to X3D and relational tables. 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 its various nodes.
2. The XSLT engine transforms the XML document using pattern matching and template rules in the style-sheet (.xsl);
3. Template elements and attribute values replace matched element/attribute patterns of the source document in the result document.

For each of our destination venues, we wrote a set of XSLT files specifying the proper attributes of the output method, encoding, media-type, and cdata-section-elements, and the DOCTYPE system for our result document.

4.13 Extensible 3D (X3D)

The Web3D Consortium's next-generation successor to VRML is X3D which, like XML, 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 bindings defined for multiple profiles and encodings and collected under a standard API [W3D]. The X3D specification describes the abstract performance of a directed, a-cyclic scenegraph which can be defined by an XML binding using DTDs and Schema. In addition, 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 application domains 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 X3D Task Group has provided a DTD, Schema, an interactive editor, and a set of XSLT and conversion tools for working with X3D and VRML97. The official file extension for X3D files is `.x3d` and the official MIME type for X3D files is defined as: `model/x3d`. While the structural geometry of the molecule could be described in X3D simply as primitives conforming to the Interchange Profile (Sphere, IndexedLineSet), our X3D data files for DIVERSE used high-level markup (Prototype tags) to describe the data in the scenegraph. This way, the displayed form of an atom or bond could be customized by the application (atoms as a Sphere, bonds as lines or cylinders).

4.2 Snap-Together Visualization

Snap provides an interface for developers to build their own Snap Adapters to custom visualization components. We take advantage of this capability by defining the CAVE Adapter, a set of classes that process events from Snap and communicate them to the remote DIVERSE visualization.

In order to embed our 2D Snap visualizations in the CAVE, we initialize our DIVERSE application (next section) and export the display of a remote machine Linux machine with Mozilla and Java to the CAVE wall. Our CAVE at Virginia Tech runs with 3 walls and a floor projection. We then run Mozilla on that remote machine and connect to our online Snap build and Molecules database. At this point, the browser window with the Snap applet are live and 'hung' on one wall of the CAVE. Since the Snap window is running from a remote Xserver and exported to a CAVE display, we can hang the picture anywhere on either the left or the right wall.

In our usability evaluation, we hung the Snap window aligned to the top-left of the right wall. Users can then connect their Snap visualization components and load, sort, and select data records of interest. As we have mentioned, Snap allows users to build their own coordinated multiple-view visualization schemas. This gives great flexibility for customizing the layout of data displays and interactions.

Figure 5 shows a typical Snap window setup connected to Molecules.dsn and coordinated with the CaveAdapters. In this screen shot, the hierarchy of molecular data tables is established through Select <-> Select event coordinations. The atom table is linked to a Scatter plot of atomic number versus atomic radius. For the case of ultra-violet (UV), infrared (IR), and Mass spectrum data, we setup Select <-> Load event coordinations to graphing components. In this case, the user has selected a Nitrogen atom from the molecule caffeine. The IR and UV spectrum response for the caffeine molecule is loaded into the bottom-left and bottom-right Scatter plots respectively.

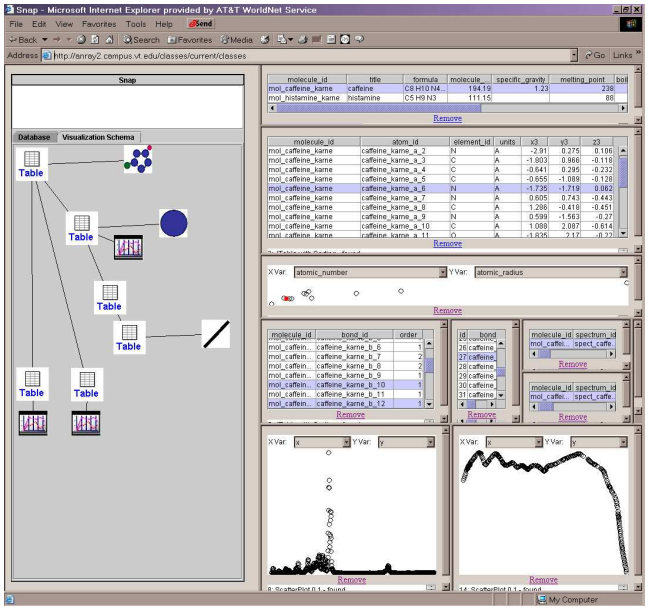

Figure 5. A sample Hanging Picture: a Snap visualization schema coordinated with the CaveAdapters. A Nitrogen atom of the Caffeine molecule is selected.

4.3 DIVERSE Virtual Environment

DIVERSE is a software toolkit for building portable virtual environment applications developed by the University Visualization and Animation Working Group at Virginia Tech. Our VE application is written for DIVERSE and extends AtomView, a molecular visualization tool for materials scientists also developed at Virginia Tech. DIVERSE provides some powerful advantages:

- A common user interface to interactive graphics and/or VE programs. Using DIVERSE the same program can be run on CAVE™, ImmersaDesk™, HMD (head mounted display), desktop and laptop without modification.
- A common API to VE oriented hardware such as trackers, wands, joysticks, and motion bases.
- A "remote shared memory" facility allows data from hardware or computation to be asynchronously shared between both local and remote processes

The DSO "plug-in" system that DIVERSE uses allows you to write one program which will run everything from a desktop workstation to a multi-walled CAVE. On each platform, it will use a display method and user interface appropriate to that platform.

S_AtomView, our DIVERSE application, is based off of work done during the summer of 2002 by Andrew Ray. His application was developed for materials science to visualize the results of large atom field simulations over time. The structure he used to represent the fields at different time steps was converted to represent different molecules. The previous implementation also did not include any bonds, so this had to be added for each molecule.

When the application starts, it reads in constants for the different elements, which are used to scale the atom sizes based on atomic radius. Next, geodes are created for each possible color, highlighted and un-highlighted, for the atoms and bonds. These geodes are linked to multiple atoms and bonds that are displayed in the application as spheres or cylinders. The program now waits for messages from the Snap side.

When a molecule message is received, the program reads the molecule name and appends the file suffix to read the atom, bond, and molecule description text data from a local X3D file. This molecule message also results in reading the molecule's corresponding color data from a separate file. This color data allows the program to link atoms and bonds to their corresponding geode. Based on the location of other molecules and the base location of the atom, atoms are added to the world. Bonds use the same information as atoms to find their correct position, but it also require vector calculations to line up between atoms.

Now that a molecule is added, the program waits for more messages. If another molecule message is received, the program removes the molecule if it already in the program and visible. Otherwise, it adds the molecule as before. If an atom or bond message is received, the program searches the corresponding molecule for the object and highlights or un-highlights based on the current state. Molecule description information can also be displayed in a Heads-Up-Display window and this information changes when you move from molecule to molecule (see Figures 1,2, and 6).

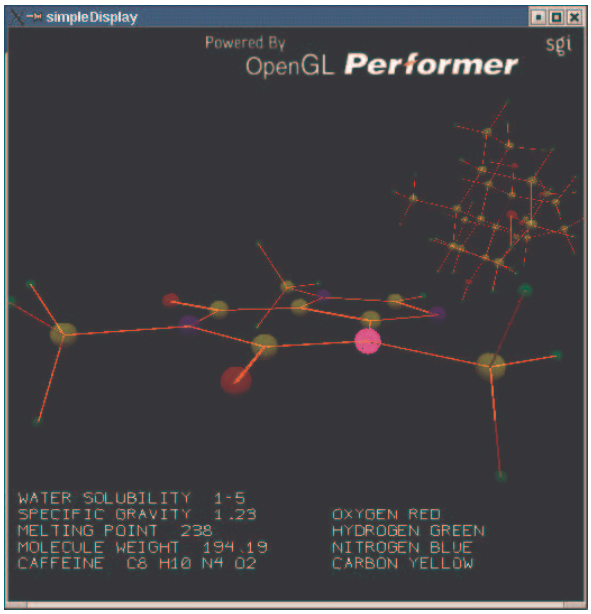

Figure 6. Two X3D molecules on a DIVERSE desktop platform. The Heads-Up-Display shows information relating to the proximal molecule. A Nitrogen atom of Caffeine is highlighted as a result of selection in Snap.

Future work on the DIVERSE side could include adding element letters inside each atom so they can easily be identified, sliders to change the transparency of the atom, and commands to allow easy re-centering and movement between molecules (see Polys 2003). Another focus should be on adding two-way communication back to the Snap side. This would require developing a mechanism for selection in 3D using the wand, which would be a challenge due to the limited number of buttons on the wand.

4.4 XWand

XWand is a program that allows you to use an input device such as a wand to start, stop, and manipulate any XWindows application. XWand works by using the DIVERSE Toolkit (DTK) to obtain input from devices in an immersive

environment such as an iDesk or CAVE. It works by using XTest extensions in conjunction with input from DTK. In addition to the XWand program, there is also an XWand DSO that, when loaded into a DIVERSE application, can stop and start the navigation for the currently running application, giving the XWand program control over the navigation events. We used this program to allow users to toggle their interactions between navigation in the CAVE environment and selection in the 2D applet view.

4.5 Event Communication and RMI messaging

The Diverse ToolKit (DTK) handles messages through its shared memory component. Snap handles interprocess communication within the JVM. Our project wanted to use both of these systems together but they do not mesh well together. If both systems were implemented in Java it would be easier, but DTK is a C++ toolkit. Our obvious first reaction was to use CORBA, but we decided this was overkill for the scope of the project that we were doing. We initially decided to try to allow two-way communication between Snap and DTK, but this also was beyond the scope of our timeframe. We decided to use the 3D world basically as a display-only environment that would allow the user to visualize information that they specifically chose from the 2D side of the world. We sent select messages based on molecules, atoms, and bonds from the Snap side of the world to the DTK part of the world.

DTK then uses the primary keys that Snap sent over and further processes those. The specific way we implemented communication in Snap was by using three specially-fixed adapters. These adapters received Snap events and then used Java RMI to send the information back to a server that we setup to receive these RMI commands. This computer was enabled with a special DTK program written that Java could via an RMI invocation. By using the networking features of DTK, we were able to then connect the CAVE to this computer and the CAVE would receive the messages from the Snap server. In the future it may be more beneficial to come up with a consolidated way of having one intelligent adapter that can send and receive Snap events from DTK. This would probably be implemented with some form of CORBA or some similar technology.

5 USABILITY EVALUATION

5.1 Procedure

The team members discussed possible usability issues and the usability tasks to test different aspects of the system. Subjects were chosen from a cross-section of people: an undergraduate chemistry student, a graduate computer science student, a materials scientist, and virtual reality experts from within our lab.
The specific aspects we wanted to evaluate are:

- Viability of a visualization involving 2D and 3D displays simultaneously
- User preference based on nature of data, i.e. whether users choose most appropriate visualizations for different types of data.
- Effectiveness of 3D to visualize inherently spatial data as against 2D.
- Use of a novel interaction system (XWand) for interaction with the 2D display in a 3D immersive environment.

The format followed for the usability study was ‘task and response’. The subjects were given a set of tasks to be performed one by one and their feedback was noted. Six tasks were formulated which can be classified into three categories: finding tasks, spatial tasks, and comparison tasks. The finding tasks involved getting the number of atoms or bonds in a molecule or finding a specific attribute of an atom, bond or molecule. The spatial tasks were to relate attributes and structural features of a complex organic molecule. These included questions such as: “How many Hydrogens are there in caffeine? Are they mostly in a ring or an arm?”, etc.. The comparison tasks asked users to detect and describe similarities and differences between two molecules such as their size, molecular weight, and shape. We used the molecules caffeine and histamine for the evaluations.
The step-by-step procedure was as follows:

- Introduction: The subjects were introduced to the system and the scope and limitations of the system were explained. The XWand interaction techniques were demonstrated, and the procedure was described. The subjects were asked to think aloud during the tasks.
- Tasks: The tasks were read out for the subject and their 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.
- Closing questionnaire: At the end of the usability study each user was given a closing questionnaire to reflect on their opinion about this system.

5.2 Results

The results of the usability evaluation were collected by summarizing the information we obtained from user observation and feedback. They consist of technical deficiencies, usability measures, and suggestions by subjects. One practical limitation of the system is the long setup time. Starting the program, launching a web-browser, loading Snap applets, connecting to a database, creating the relations between the coordinated Snap views took 10 to 15 minutes on average. Some technical shortcomings of the implementation became apparent during the study: a decrease in the 3D navigation speed as users repeatedly toggled the wand (for interaction with the Hanging Picture), and no way to automatically reset the Viewpoint to a 3D molecular structure in the space if reference gets lost (if the molecules were too far away or the user became disoriented).

Usability measures we evaluated included: the time to understand the basic concept of coordinated 2D and 3D visualizations, the learning time for system interaction, and the degree to which a new user is overwhelmed by the stimulus of CAVE, etc. For initial questions, the most inexperienced subject took a long time to answer and required suggestions from the project team. The use of wand was initially confusing for some subjects but soon they could fluently use the wand for toggling between navigation and selection mode, to navigate by flying, and to select a Snap information record (in the later tasks).

In most cases, users chose suitable visualizations to recover different information for different tasks. This suggests that users were capable of integrating and using both the abstract and spatial information via our design. However, more formal study is needed to verify this result.

Learning time was surprisingly low, but some subjects were observed to be overwhelmed by the system. This may be the result of the overall design of the environment since the displays provided much more information and affordances than needed to complete the specific tasks. That is, for the given tasks, the information the system actually provided exceeded the information required to complete that task, thus overloading the user. If users were chemistry experts or more familiar with the system, they could coordinate their own visualization schemas and thus have a better understanding of the contents of the Hanging Picture.

Some important features suggested by the subjects were to: allow the two-way selection of atoms (i.e. 3D to 2D as well as 2D to 3D; currently only the latter is possible), reduce the size of the 2D browser window or put it on some hand-held device like a tablet in order to avoid blocking of the virtual world, and maybe use a stylus and tablet interface for faster interaction than the wand. The closing questionnaire showed that the subjects considered this system a better learning device than a textbook and the wand was fairly intuitive as an interaction device. However, the opinion was divided over whether the wand was a hindrance in using the 2D Snap window.

After summarizing the results and improving the initial build, we produced an informational video about the project details and filmed it in the CAVE. We consider these results as important consideration for future work in the design of Information-Rich Virtual Environments.

6 CONCLUSIONS AND FUTURE WORK

Technologically, we have demonstrated a heterogeneous system for embedding interactive, user-defined 2D visualizations inside 3D virtual environments and coordinating them across a network. In terms of our chemical visualization and analysis application, Chemical Markup Language and XSLT gives us great flexibility in the loading of chemical data into the pipeline from different formats. We also believe our approach could be applied embed and coordinate any XML data in such Information-Rich Virtual Environments.

Our current implementation makes use of the wand and the Hanging Picture to interact with the 2D data of the Snap view. However, usability results for these interaction methods suggested improvements could be made. Selecting from the ‘Hanging Picture’ wall was slow: the response of the pointer when moving over the Hanging Picture was tedious. In addition, the Hanging Picture blocked parts of the 3D world (the molecules), and could not be minimized and maximized seamlessly. Some of these problems could be addressed by upgrading and tuning the Xwindow and Xwand settings. Another interesting alternative to explore in the CAVE are the tradeoffs involved in putting the Snap display and interaction window on a hand-held tablet.

In the evaluations, the location, size, contents, and layout of the visualization components were fixed for the users. New users experienced a high cognitive load just familiarizing themselves with the meanings and relations in the contents of the Hanging Picture. While this was affected by user expertise and familiarity with the database, we believe that the design and delivery of 2D information visualizations in a 3D spatial context is still an area ripe for future work.

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. Nonetheless, it has proven a useful system to integrate with 3D views and the CAVE.

For our system to be more successful, more virtual environment interaction techniques should be implemented and evaluated. For example, 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. Also, since we do not have two-way selection implemented (selections in the CAVE being coordinated back to selections in the Snap view), future 3D selection options should be explored including picking touch, laser pointer, and spotlight techniques [Forsberg et al. 1996][Lindeman et al. 1999].

We believe this work contributes to the field by demonstrating an Information-Rich Virtual Environment using multiple, linked views of spatial and abstract data. The architecture we describe is flexible for linking Snap with DIVERSE and could be applied to any number of data domains. Finally, our initial evaluation of this system identifies critical usability concerns when coupling 2D information visualizations with 3D immersive virtual environments.

ACKNOWLEDGMENTS

We thank Nathan Conklin and Varun Saini of Virginia Tech for their assistance with the Snap system. Additional thanks are due Dr. Doug Bowman, Dr. Ron Kriz, John Kelso, and the DIVERSE team at Virginia Tech for their support in this project.

REFERENCES

1. BENDER, M., KLEIN, R., DISCH, A., EBERT, A. 2000. "A functional framework for web-based information visualization systems." *Visualization and Computer Graphics, IEEE Transactions on,* Vol.6, Iss.1, pp. 8- 23.
2. BOLTER, JAY, HODGES, LARRY F., MEYER, THOMAS, NICHOLS, ALISON 1995. "Integrating Perceptual and Symbolic Information in VR". *IEEE* July.
3. BOWMAN, DOUG, HODGES, L., and BOLTER, J. 1998 "The Virtual Venue: User-Computer Interaction in Information-Rich Virtual Environments". *Presence: Teleoperators and Virtual Environments*, **7**(5): p. 478-493.
4. BOWMAN, D. WINEMAN, J., HODGES, L., and ALLISON, D. 1999. The Educational Value of an Information-Rich Virtual Environment. *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 3, June, pp. 317-331.
5. BOWMAN, D.A., WINGRAVE, C. A. 2001. "Design and evaluation of menu systems for immersive virtual environments" Virtual Reality, 2001. Proceedings. *IEEE* .
6. BOWMAN, DOUG 2002. "Principles for the Design of Performance-oriented Interaction Techniques". In *Handbook of Virtual Environments*. edited by Stanney, Kay M. Lawrence Erlbaum Associates. Mahwah, New Jersey.
7. BALDONADO, M., WOODRUFF, A., KUCHINSKY, A. 2000. "Guidelines for using Multiple Views in Information Visualization". *Proc. Advanced Visual Interfaces.*
8. CARD, S., MACKINLAY, J., SHNEIDERMAN, B., 1999. *Information Visualization: Using Vision to Think*, Morgan Kaufmann.
9. DATEY, AMEYA 2002 Experiments in the Use of Immersion for Information Visualization . Master's Thesis, Virginia Tech. http://scholar.lib.vt.edu/theses/available/etd-05092002-151043/
10. FORSBERG A., HERNDON K., ZELEZNIK R., 1996. "Aperture Based Selection for Immersive Virtual Environments". *Brown University site of the NSF Science and Technology Center for Computer Graphics and Scientific Visualization.*
11. KELSO, J., ARSENAULT, L., KRIZ, R., and SATTERFIELD, S. 2002. DIVERSE: A Framework for Building Extensible and Reconfigurable Device Independent Virtual Environments. *Proceedings of IEEE Virtual Reality*. http://diverse.sourceforge.net/
12. LINDEMAN, R.W., SIBERT, J.L., HAHN, J.K. 1999. "Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments" *Proceedings of IEEE Virtual Reality* pgs: 205- 212.
13. MURRAY-RUST P. and RZEPA, H. S. 1999. "Chemical markup Language and XML Part I. Basic principles", *J. Chem. Inf. Comp. Sci.*, 39, 928.
14. MURRAY-RUST, PETER, RZEPA HENRY S., and WRIGHT, MICHAEL 2001. "Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content" *New J. Chem.*, 618-634.
15. NORTH, C., CONKLIN, N., and SAINI, V., 2002. "Visualization Schemas for Flexible Information Visualization", *Proc. IEEE InfoVis 2002 Symposium.*
16. NORTH, C. 2001. "Multiple Views and Tight Coupling in Visualization: A Language, Taxonomy, and System". *Proc. CSREA CISST 2001 Workshop of Fundamental Issues in Visualization*, pg. 626-632.
17. NORTH C. and SHNEIDERMAN B. 2001. "Component-Based, User-Constructed, Multiple-View Visualization". In *Proc. of ACM CHI*, pp. 201-202.

18. NORTH C. and SHNEIDERMAN B. 2000. “Snap-Together Visualization: Can Users Construct and Operate Coordinated Views?”, *Intl. Journal of Human-Computer Studies*, Academic Press, November 53(5), pg. 715-739.
19. POLYS, NICHOLAS F. 2003. “Stylesheet Transformations for Interactive Visualization: Towards a Web3D Chemistry Curricula”, *Proc. of the Web3D 2003 Symposium*, ACM SIGGRAPH.
20. WARE, C. and FRANCK G. 1994. “Viewing a graph in a virtual reality display is three times as good as a 2D diagram”. In A. L. Ambler and T. D. Kimura, editors, *Proc. IEEE Symp. Visual Languages* (VL'94), pages 182-183.
21. WISS, U., CARR, D., JONSSON, H. 1998. “Evaluating Three-Dimensional Information Visualization Designs: a Case Study of Three Designs.” Lulea University of Technology Dept. of Computer Science and Electrical Engineering, http://citeseer.nj.nec.com/wiss98evaluating.html.
22. [W3C] The World Wide Web Consortium Specifications:
23. Extensible Markup Language (XML): http://www.w3.org/XML
24. Extensible Stylesheet Transformations (XSLT): http://www.w3.org/TR/xslt11
25. [W3D] The Web3D Consortium Specifications:
26. Extensible 3D (X3D), Virtual Reality Modeling Language (VRML- ISO/IEC 14772:1997) http://www.web3d.org/fs_specifications.htm
27. X3D TaskGroup: http://www.web3d.org/x3d.html
28. Software Development Kit: http://sdk.web3d.org