

**Design and Display of Enhancing Information in Desktop
Information-Rich Virtual Environments: Challenges and Techniques**

Nicholas F. Polys
npolys@vt.edu

Doug A. Bowman
bowman@vt.edu

*Department of Computer Science & Center for Human Computer Interaction
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24060; USA*

Design and Display of Enhancing Information in Desktop

Information-Rich Virtual Environments: Challenges and Techniques

Abstract:

Information-Rich Virtual Environments (IRVEs) have been described as environments in which perceptual information is enhanced with abstract (or symbolic) information such as text, numbers, images, audio, video, or hyperlinked resources. Desktop VE applications present the same information design and layout challenges as immersive VEs, but in addition, they may also be integrated with external windows or frames commonly used in desktop interfaces. This paper enumerates design approaches for the display of enhancing information both internal and external to the virtual world's render volume. Using standard web-based software frameworks, we explore a number of implicit and explicit spatial layout methods for the display and linking of abstract information, especially text. Within the virtual environment view, we demonstrate both Heads-Up-Displays and encapsulated scenegraph behaviors we call Semantic Objects. For desktop displays, which support information display venues external to the scene, we demonstrate the linking and integration of the scene with web browsers and the Snap-Together visualization a system. Finally, we describe the application of these techniques in the PathSim Visualizer, an IRVE interface for the biomedical domain. These design techniques are relevant for instructional and informative interfaces for a wide variety of desktop VE applications.

Keywords: information-rich virtual environments, visualization design, information psychophysics, multiple view architectures, desktop virtual environments

1. Introduction

At the intersection of the fields of virtual environments, visualization, and user interfaces we are aiming to supply users with user with relevant information at a minimum cognitive and execution load. We believe that in Information-Rich Virtual Environments (IRVEs), by enhancing perceptual information with abstract information, we can reduce Norman's 'Gulf of Evaluation' (1986) and promote a semantic directness that leads users to more accurate mental models of the phenomena they perceive. In order to "amplify cognition" as Card, Mackinlay, and Shneiderman (Card et al, 1999) suggest, we are motivated to efficiently employ human perception and cognition to design and communicate perceptual substrates for accurate interpretation and use.

However in most cases, perceptual substrates from the realm of our everyday experience are not enough. Consider the simple case where a user perceives a brick and feather dropped simultaneously from a high point. We have come to accept a scientific theory of gravity that could not be inferred from the stimuli; that is, it is not their difference in 'weight' that makes them fall at different rates. Supposedly, the difference is attributable to other variables such as 'mass' and 'air friction'. Without supplemental cues and information enhancement, such a perceptual phenomenon would be mischaracterized and misunderstood for decades or even centuries. Now (despite our everyday phenomenology) we believe it and base our action plans upon its veridicality.

Our work on Information-Rich Virtual Environments attempts to address this problem by developing design guidelines and software architectures for composing virtual environments that are enhanced with supplemental abstract information, which may be a variety of media resources and types. We have implemented a number of methods for

both in-scene annotations and desktop layouts and applied them in the domain of biomedical visualization.

In this paper we examine the challenges, tradeoffs, and techniques involved in the design and layout of supplemental annotation information for desktop virtual environments.

Section 2 motivates the enterprise by reviewing literature from information and interface design. In Section 3, we describe the design space of IRVEs and detail general challenges and those particular to desktop VEs. Section 4 contains the spatial layout techniques we are using in our work and section 5 details a real-world application of these methods.

2. Related Work

While a number of authors have written on various aspects of including supplemental, abstract information and resources in virtual environments, their discussions have suffered from a lack of precise definitions and holistic, systematic research approaches. Recently, Bowman et al. (2003) put forward a more generalized, formal theory and research agenda for Information-Rich Virtual Environments, and we will consider that paper as our launching point. We believe such a theory is helpful, if not essential, for improving the effectiveness and usability of VEs for complex, information-demanding tasks. This paper explores some of the information design issues raised in Bowman et al. (2003).

The goal of the IRVE 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 their effect. This line of inquiry has been termed the ‘inverse problem of design’ by Joseph Goguen (2000) and ‘Information Psychophysics’ by Colin Ware (2003). To this end, let us briefly review the relevant work from the domain of graphical information design and information visualization dealing with human perception, recognition, and concept formation.

The nature of visual perception is a crucial factor in the design of effective graphics and virtual environments. The challenge is to understand human perceptual dimensions and map abstract data to display in order that dependent variables can be instantly perceived - processed pre-consciously and in parallel (Friedhoff, 2000). Such properties of the visual system have been described (such as sensitivity to texture, color, motion, depth) and graphical presentation models have been formulated to exploit these properties such as pre-attentive processing (Pickett et al, 1996) and visual cues and perception (Ware, 2003; Keller, 1993). Watzman (2002) has examined usability guidelines and visual design principles as they relate to text typography and color usage. Watzman details the relation of principles such as harmony, balance, and simplicity to text legibility and readability.

In the context of IRVEs, as we shall see below, we are especially concerned with visibility, legibility, and the association of related abstract information to its referent object.

How users recognize and construct knowledge about what a graphic ‘means’ is also of crucial importance in visualization and IRVE applications. For users to understand and interpret complex images, higher-level cognitive processes are usually needed. A number of authors have enumerated design strategies and parameters for representing

signifiers in graphics (Bertin, 1981; Tufte, 1990) and there are effects from both the kind of data and the kind of task (Schneiderman, 1996). Thus we expect that we may have to identify IRVE design heuristics according to data type, display type, and task context.

Vanderdonckt & Gillo (1994) summarize visual layout techniques from an aesthetic and psychological point of view relating methods such as composition, association/dissociation, and ordering on a 2D grid-based structure. They also conclude that effective visual design should rely on task analysis. Sutcliffe & Faraday (1994) concentrate on user task and resource analysis to determine a task-knowledge structure, which they formalize as an entity-relationship model. This model enables the effective design of multimedia interfaces and presentation scripting- e.g. what media resources the user needs visual access to when. This is a useful approach to consider for IRVE design as it intends to formally identify items that need user attention and minimize perceptual overload and interference.

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 multisensory cues (Salzman et al, 1999). This is compelling evidence for the value of VEs as experiential learning tools (learning by doing) and for concept acquisition during the development of a (student) user's mental model. It is important to remember however, that subjects at different stages of cognitive development assign different meanings to the same perceptual event. By explicitly adding information about what the student is viewing within the virtual environment, IRVEs can enable more accurate interpretation.

It is our larger hypothesis that enhancing information in virtual environments can serve both an instructional purpose and an informative purpose. When designing for domain experts for example, having simultaneous access to supplemental, abstract information about a perceptual event can be valuable for insight generation. Insight and breakthroughs in science are often the result of mapping structures or principles from one domain to another; for example, reframing or redefining the problem in alternate terms (Perkins, 2000) or blending two or more cognitive spaces (Fauconnier, 1997). In addition, a flexible system for user-constructed multiple views (eg North and Schneiderman, 2000) can allow users to coordinate these views in ways unforeseen by the original designers.

3. Design Space and Layout Challenges

Supplemental 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 virtual environment. 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 or explicit user action such as selecting an object for details-on-demand. Users may have to browse, search, recognize, and compare this information in a unified visual context.

Annotations may be simple labels, detailed attributes such as field-value pairs, graphs, or related multimedia. With the exception of hyperlinked resources (which may be another 3D world), all types of annotations we have mentioned can be displayed on 2D surfaces

we call *panels*. Following Feiner et al's work (1993) in display techniques for Augmented Reality, we divide the possible display locations for annotation panels into *object-fixed*, *world-fixed*, *user-fixed*, and *display-fixed* categories. According to Feiner et al, augmenting information may be associated to a particular object in the world (object-fixed) or associated to a location in the world (world-fixed). If information travels with the user regardless of their navigational actions, it is classified as user-fixed. If the information is persistently located on the display screen, it is termed display-fixed. Annotations may be sourced from local or remote databases, from the 3D scene file, or both. How this informational content could actually be rendered is the focus of this section.

3.1 General Challenges

Inside the 3D scene or viewing frustum, IRVE designers must tackle a number of design challenges and perceptual issues. These include visibility, legibility, association, aggregation, and occlusion. We will examine each of these issues in turn.

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 nearby 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 as it dominates the visual field and becomes perceived as the referent itself rather than an attribute of the object.

Legibility

This brings us to our next consideration in the case of supplemental text or numeric information, which is *legibility*. If an annotation (such as text) is to be displayed and legible, it must be of sufficient size that users can read its letters and numbers. In the case of object or world fixed annotations, this 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 is 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 Billboards- not simply rotating around the y-axis as VRML or X3D Billboards do (Web3D, 1997, 2003). 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. As we shall see below, this background panel 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 reference relation with

minimal cognitive overhead. The laws of Gestalt perception (most recently summarized in Ware, 2003) including connectedness, proximity, common region, similarity, and common fate are most relevant here. In the case of an object-fixed annotation, the relation 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, being ‘near enough’ to an object that the relation is perceived (proximity, common region), or the annotation is rendered with the same color scheme as its referent object (similarity). Common fate refers to the principle that objects that move together in similar trajectories are related. The challenge to either implicit or explicit relations is that the relation can be understood from any perspective, even if the referent object is oddly shaped.

Aggregation

The content(s) of an annotation may be of a variety of data types, 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, 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 such as hyperlinked menus and display logic.

Occlusion

Finally, when considering the design of object and user fixed annotation panels, there is 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. Management and layout of panels in these situations can be accomplished either by a centralized manager class that knows the image-plane size and the span of 3D object's 2D projection or by a distributed ruleset that gives rise to emergent behaviors such as flocking.

3.2 Challenges Particular to Desktop VEs

In desktop virtual environments, designers have significantly more flexibility as to how and where annotation information is displayed. In immersive systems such as CAVEs or HMDs, user-fixed and display-fixed annotations are perceptually equivalent and there is no visible screen real estate outside the render volume. In desktop VEs however, a user-fixed display location such as a Heads-Up-Display (HUD) is categorically distinct from display-fixed locations such as HTML frames or popup windows, which also populate the screen space. These display-fixed locations are external from the user's viewport on the scene. 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 (Polys, 2003a).

In desktop contexts, where multiple, external frames and windows are viable display venues for complementary information and supplemental views, designers must 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 nearby an 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.

The details of association and interaction design are likely to be platform, application, and task specific. Still, steps can be taken in order to insure the perception of association by similarity (across windows or frames): the color of the description text or the color of the annotation background should match the referent object's color for example.

Violating such a simple guideline can lead to confusing and cognitively expensive user behavior (Stroop, 1935).

There is also the display of parallel information where the same data in the 3D scene is displayed on screen with another visual encoding in the display-fixed venue. Generally, these can be classified as the *multiple-views* approach and require the sharing of addressable data objects and coordinated event communication between the views. If a user selects a data object in a 2D view, the corresponding object(s) in the 3D environment should feedback and be selected and vice versa. In the case of standard web data formats and runtimes, such coordination is implemented through an API (application programmer interface). The API for VRML is the External Authoring Interface (EAI) and for X3D it is called the Scene Access Interface (SAI). While the EAI is capable in its own right, the new SAI provides a much more expansive and rigorous set of functionalities for integrating applications and windows with the virtual environment.

4. Design and Layout Techniques

On desktop platforms, designers have limited screen space to work with. Browser windows and embedded media objects in web pages (such as Web3D worlds) are usually sized in absolute pixels, while frames and tables can be sized by percentages or absolute pixels. Web pages are particularly adept at the integration and layout of media types including text, images, audio and video. In addition, using the HTML `<object>` tag, designers can embed VRML or X3D worlds in web pages that can be automatically loaded with a browser plugin.

For VRML and X3D worlds embedded in a web page at a fixed size, the user perspective on the virtual environment is specified by the `fieldOfView` field of the `ViewPoint` node. This is a value in radians with the default value being $\pi/4$; larger values create fish-eye effect while smaller values create tunneled, telescoping effects. Naturally, with a larger `fieldOfView`, more of the virtual environment content 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). Interestingly, the visibility, legibility, and association criteria described above are not violated when fish-eye views are used in conjunction with our techniques.

This section describes a set of solutions to the display challenges enumerated in the previous section. We motivate our designs from the work of Bederson et al (1996) who propose 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 the context. They have termed this 'interface physics' and demonstrated the Pad++ system for

‘semantic zooming’ where both the content of the representation and the manipulation affordances it provides are directly and naturally available to the user. We have encapsulated similar appearance and interaction behaviors in the definitions of 3D objects themselves and implemented a range of design options and layout techniques for the display of related abstract information across scales. We call these ‘Semantic Objects’.

4.1 General Techniques

The basic units in our system are a set of annotation panels that can be used in the 3D scene for object, world, or user fixed annotations. These panels address the legibility requirement noted above and expose as much functionality as possible for the world author including font color, 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. The string content of the Text node are exposed so that their content can be updated from events in the scenegraph. For textual and numeric information, we implemented 2 different panels for common situations: an ‘unstructured’ panel and a field-value pair panel with title (left and right of Figure 1 respectively). Similar panels may be constructed for Image and MovieTextures.

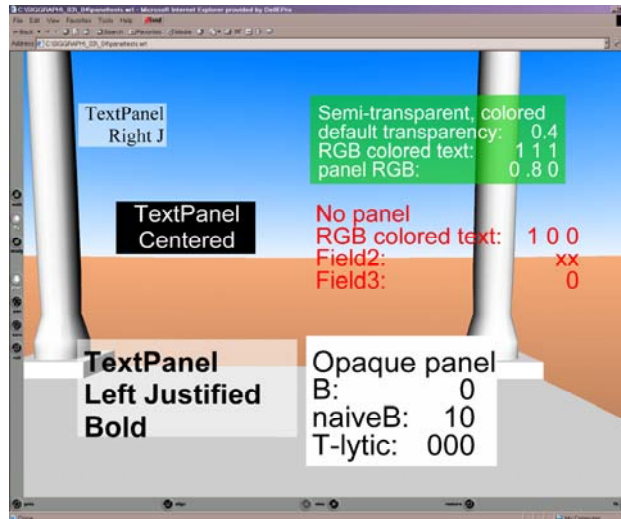


Figure 1: A variety of Text Annotation Panel configurations

Semantic Objects are especially designed for object and world fixed annotations. A number of important layout behaviors are encapsulated in the definition of Semantic Objects, which are parameterized for various solutions to the aggregation, visibility, and association issues mentioned above. Semantic Object nodes maintain two sets of ordered children: one for object geometries and one for annotation informatics (annotation panels). 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 time. Thus, authors can choose to aggregate abstract information when the user is far away and show progressively more detail as they approach the object.

The children of the infoLevel can be scaled a number of ways to maintain visibility. We implemented smooth scaling and periodic scaling as a function of user distance, as well as constant size. Preliminary evaluations on dynamic sizing of annotation panels have shown that the smooth scaling technique can confound the user's normal depth cues and thus periodic sizing may be preferable.

In addition, we implemented a number of spatial layout techniques to address the association problem in the case of object-fixed annotation panels. The first we call *Relative Orthogonal*, where the author simply specifies an infoPosition value (x, y, z) relative to the referent object. As the user navigates around the object, the annotation panel rotates to maintain this relative position orthogonal to the user's perspective. Figure 2 shows an example of this technique.

The second spatial layout technique for object-fixed association we call the *Bounding Prism* method. In this method, the annotation panel snaps to the corner of the object's bounding box nearest the user. One issue here is that when the panel is located on a 'left' corner of the bounds, the panel may have to be shifted left so as not to occlude the object. Optionally for oddly shaped objects, the author may specify a series of 8 coordinates that define a bounding prism containing the object. While the bounding prism is not a rendered object, Figure 3 shows an example of this technique with the bounding prism rendered for reference.

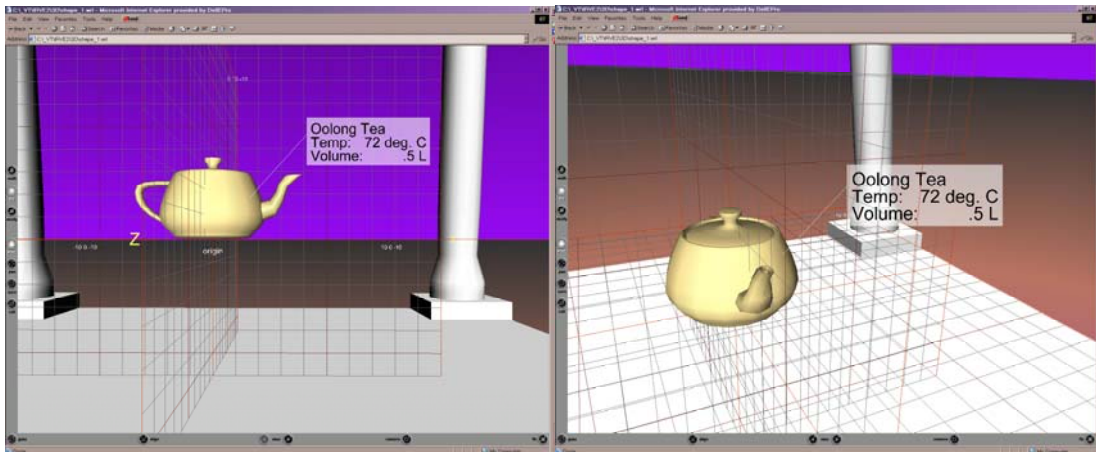
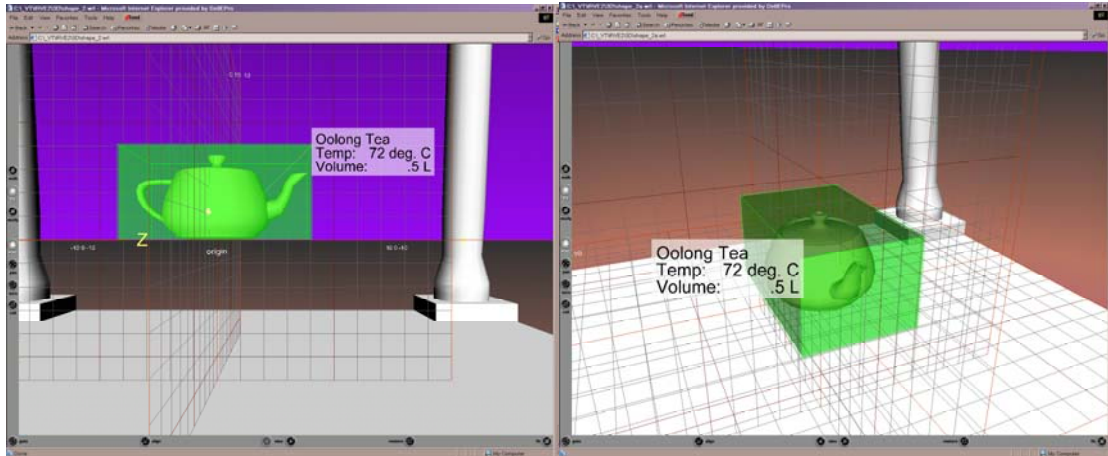


Figure 2: Object-fixed layout of a Semantic object's Annotation Panel (Relative Orthogonal)



**Figure 3: Object-fixed layout of a Semantic object's Annotation Panel
(Bounding Prism shown for reference)**

Lastly for object-fixed annotations, we examined a layout method to minimize the occlusion among panels and objects. Since we wanted to encapsulate layout behaviors in our Semantic Objects without resorting to an external manager, we implemented a *flocking* algorithm similar to Reynolds (1987) in which a simple set of attraction/avoidance rules give rise to complex, emergent behavior. In our current version, annotation panels are attracted to the nearest corner of the bounding prism and avoid other Semantic Objects. While this approach requires some tuning of attraction/repulsion depending on environment crowding, by implementing such an algorithm we can avoid the need for centralized control of annotation layout. Figure 4 shows the result of our approach.

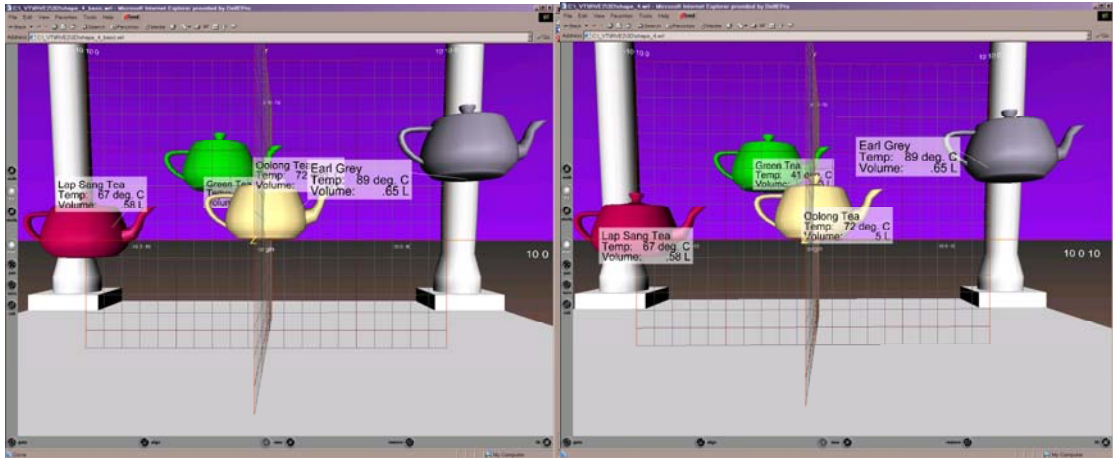


Figure 4: Object-fixed layouts of Semantic Objects without (left) and with (right) flocking behavior

User-fixed annotations are fixed to the user's view as they navigate the virtual environment. Typically, they appear as Heads Up Displays (HUDs) or Overlays on the image plane. While both are perceptually equivalent to the user, implementations can vary considerably. For example, Java3D has an API for the Canvas object (the image plane) onto which 2D graphics can be laid. The Canvas has x and y dimensions and objects rendered there are not defined as part of virtual environment (scenegraph) itself. In contrast, a common HUD as implemented in VRML or X3D is defined in the scenegraph, does not necessarily know the size of the image plane, and can have objects with x, y, and z coordinates. In this paradigm, the HUD must be offset from the user's viewpoint and scaled within the field of view (usually at the near (z) clipping plane in order that objects from the scene do not interfere with visibility at close ranges).

We implemented a VRML HUD prototype object that can take arbitrary sensor and geometry nodes such as text or image panels as children. Because these nodes are instantiated in the scenegraph, it is trivial to route events to objects in the HUD and vice versa. As we mentioned above, this is crucial to establishing correspondence relations

between scene objects and their annotation information through implicit or explicit user interaction. Figure 5 shows an example of our HUD object in use.

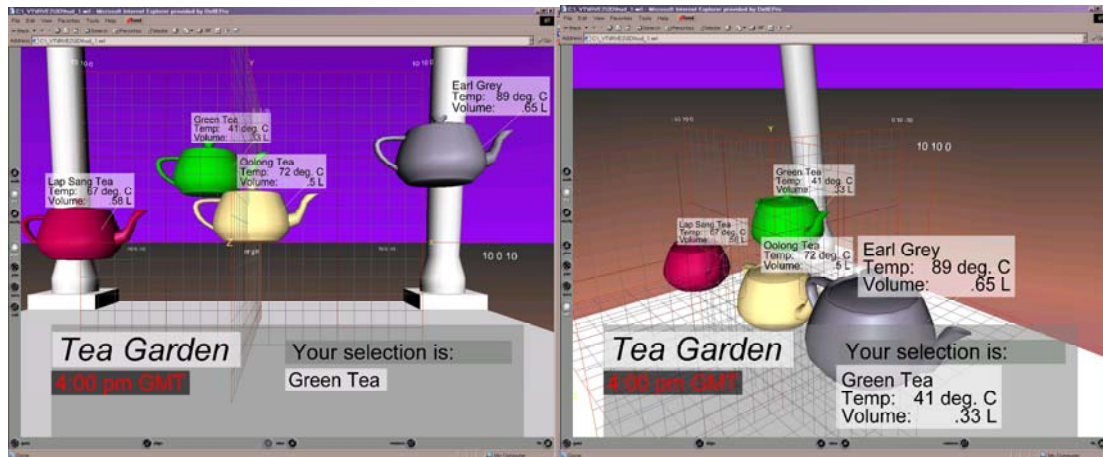


Figure 5: User-fixed layout of Annotation information on a HUD: Semantic Object annotations are displayed by mouse-over (left) and by selection (right)

4.2 Techniques Particular to Desktop VEs

While desktop virtual environments provide the experience of interactive 3D spaces to a larger audience (especially through Web3D standard formats), they also open up a large set of challenges and choices designers have to consider such as screen resolution and window management. In a web browser, supplemental abstract information may be provided by hyperlinks to load information in frames, popup windows, and applets.

If a user mouses-over an object in the 3D scene (an explicit action) and a window frame updates, a referential association is established. Similarly, if a user selects an object in the 3D scene and a popup window opens loading a display of detail attributes, that window becomes the annotation venue. This type of functionality is familiar to web users and standard formats include the Anchor node with a parameter where the target window or frame object may be specified. Other types of window management can be

accomplished through the use of scripts written in Javascript (ECMAScript) for example. This type of drill-down hyperlinking can be used for the display of further detail information, and may be classified as the traditional overview + detail technique from information visualization.

Semantic object's annotation panels themselves can carry Anchor node hyperlinks that launch a popup window for supplemental information or a detailed 3D view. Previously (Polys, 2003a), we have implemented a web-based portal interface using HTML, VRML, and Javascript that sniffs the user's screen resolution and launches supplemental information in a popup window when the user clicks on a world-sphere's 'Info' panel for more details (see Figure 6). Similar techniques can be used for the sizing and loading of annotations in frames. The VE itself is embedded in a frame where links in the HTML menu (left) load content in the 3D window.



Figure 6: Web Portal IRVE using frames to manage virtual world content and popup windows to manage display-fixed annotations

North (North 2001) has described a taxonomy and system for tightly-coupled multiple views which allows users to build their own coordinated visualizations. In the multiple-

views paradigm, unique object IDs are required to properly route events back and forth between windows. In the context of desktop IRVEs, visualizations across windows and frames can be coordinated by simple events such as: 1. Selecting 3D items ↔ selecting 2D items, 2. Navigating 3D views ↔ navigating 2D views, and 3. Selecting 2D items ↔ navigating 3D views for example. Respectively, a select event received by a Semantic Object could cause it to highlight; user proximity to Semantic objects could cause 2D visualizations to load in alternative windows or frames; Selecting an object's attributes from a table could cause the user's viewpoint to be updated to that object's location. Since the 'Visualization Schema' approach has shown a significant speed up on Overview+Detail tasks (North & Shneiderman 2001, 2000), we have successfully implemented a subset of these coordinations using the DIVERSE toolkit (Kelso et al, 2002) and data from Chemical Markup Language (Polys et al, 2004).

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. Recent empirical research supports these guidelines (Convertino et al, 2003) and methodologies for designing multiple windows (display-fixed annotations) should evaluate their design according to these criteria.

5. Application Example: Biomedical Visualization

We are using our Semantic objects and HUD in a biomedical visualization application designed for desktop users via a web interface. Anatomy and medical applications are a prime domain for IRVEs since there can be a wealth of abstract and temporal information related to spatial objects such as systems, organs, tissues, cells, etc. The PathSim Project (Duca et al, 2003) simulates pathogen/host agent interaction with a computer model built from clinical knowledge. Users can analyze the simulation results from the global level down to microscopic tissue with numerical, color coded, and histogram renderings of population numbers. Figure 7 shows a set of result data set from a simulated viral infection of the tonsils.

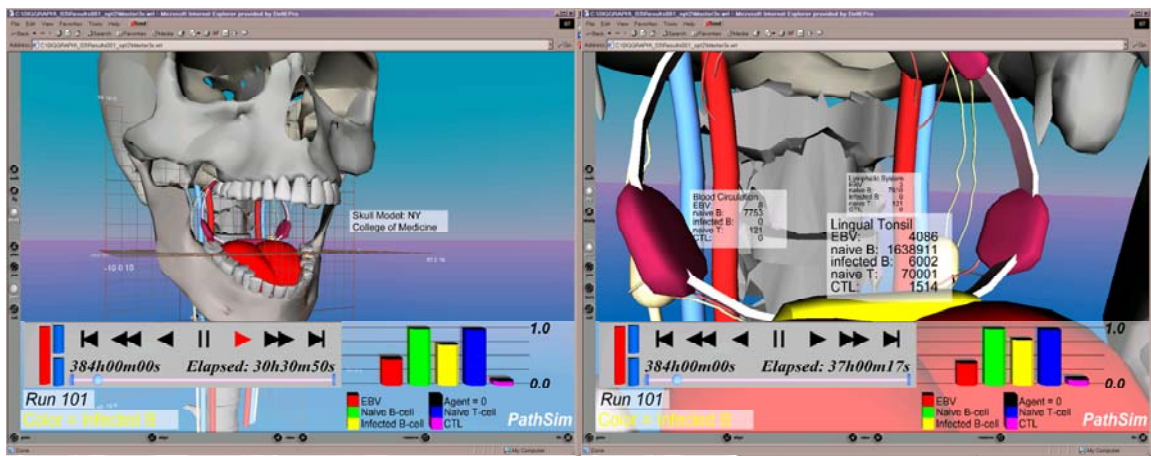


Figure 7: PathSim desktop IRVE application showing Semantic Objects and HUD display venues

For this application, systems biology investigators require both overview and detail abstract information as it relates to various systems and parts of the anatomy over time. For global view and control regardless of scale and time, the HUD contains DVD controller buttons and a slider interface for users to navigate through the time series. Additionally on the HUD are selectable buttons where users can pick the active population color attribute for the tonsils. The active color-coding and the simulation

run's index number are also explicitly displayed on the HUD using our text annotation panels.

In the PathSim desktop IRVE interface, we use Semantic Objects in a number of ways. For example, the 3D anatomical models are annotated with names, references, and scale factors. As users zoom into smaller scales, more detailed geometry and information is displayed. At these detail levels, field-value pairs are displayed showing the population numbers for each active tonsil and system (eg circulatory and lymphatic). At the microscopic levels, each unit tissue section is also annotated with population field-value pairs. The unit tissue sections are hyperlinked to popup detail window views of the finite difference mesh on which the simulation runs.

Given the anatomy and application requirements for this build of PathSim, we have determined that the relative orthogonal layout technique with periodically scaled annotation panels yields the most consistent display results with regards to visibility, legibility, association, and occlusion. User focus-groups confirm this result. Other anatomies (such as the brain or lungs) and other applications (such as educational) are likely to have different requirements for abstract annotation and the display techniques employed. Our hope is that a finite, well-designed set of annotation layout techniques such as these can answer any situation.

6. Conclusions and Future Work

We have enumerated the scope of design challenges, options, and implementations for the display of abstract information in desktop virtual environments. The techniques we describe solve a number of fundamental challenges for information design across display

locations. The next phase of our IRVE research involves the empirical evaluation and comparison of these IRVE design methods. We hope the results of this research will aid the development of design heuristics for improved instructional and informative interfaces that are applicable to a variety of domains including medicine, architecture, and CAD/CAM.

In addition, this work is contributing to the development of better standards such as X3D to support integrated visualization of abstract information in virtual environments. For example, our prototypes are elucidating perceptual and architectural issues for the design of new X3D components such as Annotation and Compositing. As data archives trend toward self-describing and annotated repositories, a systematic consideration of challenges and techniques for the display of enhancing abstract information is required for the development of more usable and better integrated IRVE interfaces.

Acknowledgements

The authors would like to thank members of the 3D Interaction Group at Virginia Tech's Center for Human Computer Interaction for their review of and input to this manuscript, especially Chad Wingrave and Jian Chen. In addition, the people involved with the PathSim project: Dr. Karen Duca, Dr. Reinhard Laubenbacher and their research team at The Virginia Bioinformatics Institute: Jignesh Shah, Rohan Luktuke, and John McGee. All screenshots are captured from a Windows desktop using the ParallelGraphics Cortona VRML plugin with Internet Explorer (www.Parallelgraphics.com). All trademarks are the property of their owners.

References

- Baldonado, M., Woodruff, A., Kuchinsky, A. (2000) "Guidelines for using Multiple Views in Information Visualization". *Proc. Advanced Visual Interfaces*.
- Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., David, B., & Furnas, G., (1996) "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics", *Journal of Visual Languages and Computing*, 7(1), 3-32.
- Bertin, Jaque (1981) *Graphics and Graphic Information Processing* (trans. William Berg & Paul Scott). de Gruyter, Berlin New York.
- Bowman, Doug, North, Chris, Chen, J., Polys, N., Pyla, P., Yilmaz, U., (2003) "Information-Rich Virtual Environments: Theory, Tools, and Research Agenda". *Proc. Virtual Reality Software and Technology (VRST)* .
- Card, S., Mackinlay, J., Shneiderman, B., (1999) *Information Visualization: Using Vision to Think*, Morgan Kaufmann.
- Convertino, Gregorio.; Chen, J; Yost, B.; Young-Sam Ryu; North, C. (2003) "Exploring context switching and cognition in dual-view coordinated visualizations". *Proc. Of the International Conference on Coordinated and Multiple Views in Exploratory Visualization*. Iss., 15 pp. 57- 66.
- Duca, Karen, Laubenbacher, Reinhard, Polys, Nicholas, Shah, Jignesh, Lukutke, Rohan, McGee, John (2003) *PathSim*, Virginia Bioinformatics Institute.
<http://www.vbi.vt.edu/~pathsim> .
- Fauconnier, Gilles, (1997) *Mappings in Thought and Language*, Cambridge University Press, New York.
- Feiner, S., Macintyre, B., Haupt, M., & Solomon, E. (1993) "Windows on the World: 2D Windows for 3D Augmented Reality", *Proc. of the ACM Symposium on User Interface Software and Technology*, 145-155.
- 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", *ACM Proc. CHI*, pp 16-23.
- Goguen, J. *Information Visualizations and Semiotic Morphisms*. (2000) UCSD
<http://www-ecse.ucsd.edu/users/goguen/papers/sm/vzln.html> .
- Keller, Peter R. (1993) *Visual Cues: Practical Data Visualization*, IEEE Computer Society Press, Piscataway, NJ.
- Kelso, J., Arsenault, L., Kriz, R., and Satterfield, S. (2002) "DIVERSE: A Framework for Building Extensible and Reconfigurable Device Independent Virtual Environments". *Proc. of IEEE Virtual Reality*. <http://diverse.sourceforge.net/> .
- Marilyn C. Salzman, Chris Dede, R. Bowen Loftin, & Jim Chen., (1999) "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning". *Presence: Teleoperators and Virtual Environments*, vol. 8, no. 3, pp. 293-316.

- Norman, D.A. (1986) Cognitive Engineering. In User Centered System Design, eds. DA Norman & SD Draper, Lawrence Erlbaum Associates, Hillsdale, NJ pp. 31-61.
- 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.
- North C. and Shneiderman B. (2001) "Component-Based, User-Constructed, Multiple-View Visualization". In *Proc. of ACM CHI*, pp. 201-202.
- 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.
- Perkins, David (2000) Archimedes' Bathtub. W.W. Norton & Co., New York.
- Pickett, Ronald M, Grinstein, G, Levkowitz, H, Smith, S (1995) Harnessing Preattentive Perceptual Processes in Visualization. In: Grinstein, G, and Levkoitz, H (eds.) Perceptual Issues in Visualization. Springer, Berlin Heidelberg New York.
- Polys, Nicholas F. (2003a) The VirtuPortal: Opening New Dimensions on the Web. <http://www.3DeZ.net>
- Polys, Nicholas F., Ray, Andrew, Moldenhaur, Max, Dandekar, Chetan, North, Chris (2004) "Snap2Diverse: Coordinating Information Visualizations and Virtual Environments". *Proc. SPIE Conference on Visualization and Data Analysis*.
- Reynolds, C.W., (1987) "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics (SIGGRAPH 1987)*, 21(4), 25-34.
- Shneiderman, B (1996) "The eyes have it: A task by data type taxonomy for information visualizations". *Proc. of IEEE Visual Languages*, pp. 336-343.
- J. Stroop, (1935) "Studies of interference in serial verbal reactions," *Journal of Experimental Psychology*, vol. 18, pp. 643—662.
- Sutcliffe, Alistair and Faraday, Peter (1994). "Designing Presentation in Multimedia Interfaces", *ACM Transactions on Computer-Human Interaction*, Boston.
- Tufte , Edward (1990) Envisioning Information. Graphics Press, Cheshire, CT.
- Vanderdonckt, Jean and Gillo, Xavier (1994). "Visual Techniques for Traditional and Multimedia Layouts", *ACM Transactions on Advanced Visual Interfaces*, Boston.
- Ware, Colin (2003) Design as Applied Perception. In: HCI Models, Theories, and Frameworks: Towards a Multidisciplinary Science. ed. J.M. Carroll, Morgan-Kaufmann, San Francisco.
- Watzman, Suzanne, (2002). Visual Design Principles for Usable Interfaces, In: Stanney, K (ed) Human Computer Interaction Handbook: design, implementations, and applications, Lawrence Erlbaum Associates Mahwah, NJ.
- Web3D Consortium Specifications: http://www.web3d.org/fs_specifications.htm
VRML [ISO/IEC 14772-1:1997, ISO/IEC 14772-2:2002]
X3D [ISO/IEC FCD 19775:200x, ISO/IEC FCD 19776:200x, ISO/IEC FCD 19777:200x]