PathSim Visualizer: an Information-Rich Virtual Environment Framework for Systems Biology

Nicholas F. Polys
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061
(01) 540.231.6931
npolys@vt.edu

Doug A. Bowman
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061
(01) 540.231.6931
bowman@vt.edu

Chris North
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061
(01) 540.231.6931
north@vt.edu

ABSTRACT

Increasingly, biology researchers and medical practitioners are using computational tools to model and analyze dynamic systems across scales from the macro to the cellular to the biochemical level. We are using Information-Rich Virtual Environments (IRVEs) to display the results of biological simulations, and to allow users to interact with those simulations. While simulation architectures, algorithms, and processing power have enjoyed continuous optimization to date, the user interfaces to these applications have not. The problems of designing such IRVE interfaces arise from the requirement that a variety of spatial and abstract information must be integrated into one coherent experience for the user. This paper explores the design and development issues encountered in our implementation of a bioinformatics application, PathSim (**Path**ogen **Sim**ulation). Specifically, we describe the information and interaction issues in building a front-end tool to visually analyze the results of an agent-based immunology simulation. Finally, we present custom scenegraph objects and consider candidate functionality for future standards components.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Language Constructs and Features – *abstract data types, polymorphism, control structures.*

General Terms

Design, Human Factors, Standardization, Languages.

Keywords

Virtual Environments, Bioinformatics, Information Visualization

1. INTRODUCTION

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

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

We are applying the usability engineering process [Rosson & Carroll, 2003] to develop a visualization tool for in silico immunology simulations. In silico experiments are useful when little clinical data exists or when experiments are too dangerous or unethical to perform in vivo. The PathSim Project [Duca et al., 2003] simulates pathogen and host interaction with an agent-based computer model built from current biomedical knowledge. In PathSim, systems biology investigators are concerned with different infection behaviors as they are related to various systems and parts of the anatomy over time. PathSim simulations may run on large servers or clusters, but the results must accessible to researchers on desktop machines across the network.

Our work has been iterative, gathering user requirements, designing and implementing the interface framework, and refining it through user evaluations. This paper enumerates the problems and tradeoffs we encountered in building the prototype system for PathSim Visualizer and provides the rationale and details behind our design solutions. These solutions involve encapsulating physical scales and information behaviors into custom scenegraph objects that manage scale, timeseries, and information visualizations for in silico research and analysis. We believe the work presented in this paper highlights current deficiencies and opportunities for standards-based 3D information environments.

2. RELATED WORK

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

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

While PathSim has obvious medical applications, it goes much further than that. It may also serve as a basic research tool for life scientists working on a range of questions and a teaching tool that could find application from K-12 all the way to professional medical training. The value and need for such tools have long been recognized [Farrell & Zappulla, 1989; Kling-Petersen et al, 1999]. It has been shown that conceptual learning can be aided by features of VEs such as: their spatial, 3-dimensional aspect, their support for users to change their frames of reference, and the inclusion of multi-sensory cues [Salzman et al., 1999]. This is compelling evidence for the value of VEs as experiential learning tools and for concept acquisition during the development of a user's mental model. The NYU School of Medicine (Bogart et al., 2001) has published a number of anatomy courseware modules in VRML that provide an IRVE interface to detailed models of the human head. The Open Virtual Reality Testbed Group at the National Institute of Standards and Technology has produced AnthroGloss [Ressler, 2003], which is an IRVE Anthropometric Landmark Glossary in VRML. We combine referenced elements and adaptations of these models to provide users with context as they explore PathSim simulation results.

Systems biology researchers have begun to use modern computing power to simulate the immune system using generalized cellular automata [i.e. Celada & Seiden, 1992; Bernaschi et al., 2000; Grilo et al., 2001; Puzone et al., 2002]. These simulations use probabilistic or deterministic rules to govern the interaction of automata on some lattice or in some grid space. There is a broad range of implementation details concerning the simulation that cannot be covered here. These are principally concerned with the nature and evaluation of the rules governing agent interaction. However, the PathSim system is unique in that the agents (Virions, B-cells, T-cells, etc.) may number in the millions (10^8) and they travel and interact on a micro-scale 3D mesh that approximates average human anatomy.

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

3. SYSTEM DESIGN

To provide access to a broad range of users, the PathSim simulation engine is run on a server or High-Performance Computing (HPC) system, but provides setup and visualization facilities through a web-based front end (Figure 1).

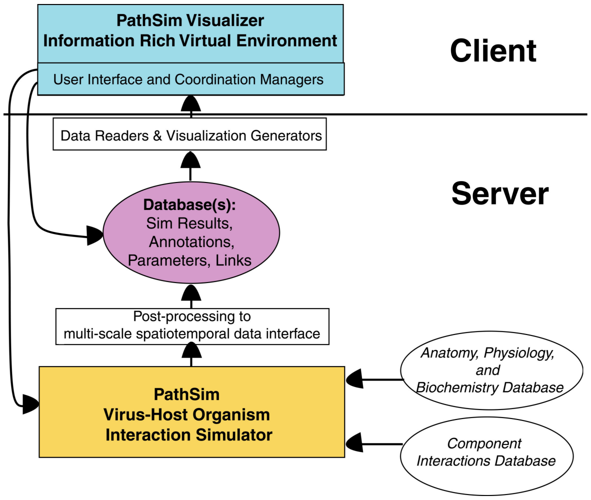

PathSim Visualizer
Information Rich Virtual Environment
User Interface and Coordination Managers
Client
Data Readers & Visualization Generators
Server
Database(s):
Sim Results,
Annotations,
Parameters, Links
Post-processing to
multi-scale spatiotemporal data interface
Anatomy, Physiology, and Biochemistry Database
PathSim
Virus-Host Organism
Interaction Simulator
Component Interactions Database

Figure 1: PathSim Architecture

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

We applied Hierarchical Task Analysis [Diaper, 1989] to the generated list of end-user activities, and enumerated the subtasks and artifacts required for each to be accomplished. To understand the details of user interactions across the scales of information and space, we developed a 'Task-Action Grammar' [Payne & Greene, 1986] for each task that allowed us to consider the command lexicon and keep the operational rules simple- combinations of navigation, selection, and manipulation interactions had to operate consistently across scales and data views.

From an information design perspective, we then employed a 'Task-Knowledge Structure' analysis [Sutcliffe & Faraday, 1994], which concentrates on user task and resource analysis to formalize an entity-relationship model. This model enables the effective design of multimedia interfaces and information presentation – i.e. what media resources the user needs visual access to when. This is an important technique for IRVE design as it intends to formally identify items that need user attention and minimize perceptual overload and interference per task.

As we have mentioned, the variety and volume of data to be displayed by PathSim Visualizer required a rational design approach. Multiple meshes, and the simulation results on those meshes had to be accounted for. In IRVE terms, we had to address: where the enhancing information was located, how it was associated to an object or location, and how dense and aggregated it was depending on the viewed scale.

A variety of user controls are required within the PathSim IRVE for navigation, selection, and manipulation. Navigation requires not only both spatial and temporal agency in the context of the virtual environment, but also the steerage of views of abstract information. Each anatomical structure has scalar properties and descriptions, and users may require overview and detail information of different objects or levels simultaneously; scene logics and selection interfaces aid users in fulfilling this requirement. Manipulation user interfaces are essential for time and space indexing.

3.1 Multi-scale Spatial Information

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

The anatomical description is hierarchical XML and distributed across a number of referenced files. The fundamental unit is a hexagonal section of tonsiliar tissue modeled to include mesh points representing: the tonsil surface, reticulated epithelium, mantle zone, and germinal center. Figure 2 shows a visualization of the unit anatomical mesh with spheres representing the location of mesh points and white lines representing the possible travel paths for agents. Blood from the circulation system enters the tissue through the High-Epithelial Venule (HEV) and lymph is drained into the lymphatic system from the mantle zone. Figure 3 shows a labeled example of how unit tissues are arranged to approximate the lymphatic tissue of the tonsils. Users can generate interconnected lattices of the unit and tonsil mesh by supplying the tonsil's surface area dimensions. After the main tonsils of the Waldeyer's ring are defined, another type of tissue (diffuse lymphatic) is instantiated to connect them. The relation of all tonsil and connective tissues is described in a macro-level tissue file that defines the simulation environment. Any subsequent processing and visualization is based on references to this hierarchical simulation mesh.

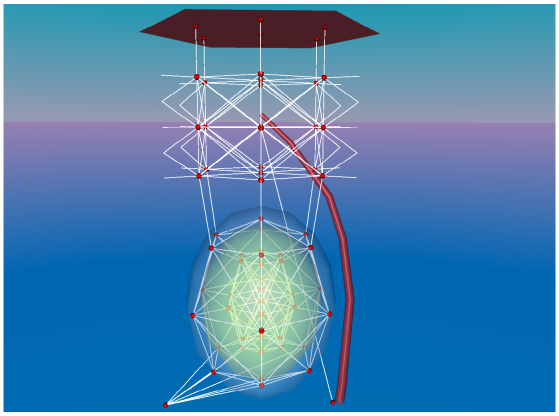


Figure 2: a VRML view of the unit section tissue mesh translated from XML

PathSim Visualizer manages an integrated information environment across multiple scales. Users have a number of first-person spatial navigation options including free-navigational modes such as: fly, pan, turn, and examine. This empowers users to explore the system, zooming in and out of anatomical structures as desired. Expert users can employ control keys for quick mode changes. To aid wayfinding, certain structures persist across scales (serving as landmarks). In addition, the result space is navigable by predefined viewpoints, which can be visited sequentially or randomly through menu activation. This guarantees that all content can be accessible, and users can recover from any disorientation.

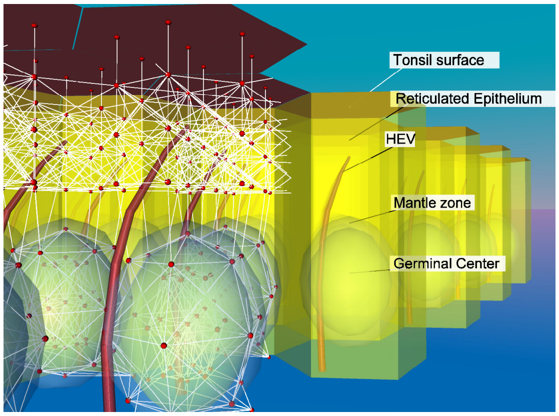


Figure 3: A labeled view of a tonsil tissue mesh

PathSim Visualizer manages an integrated information environment across multiple scales using standards technology. Users have a number of first-person spatial navigation options including free-navigational modes such as: fly, pan, turn, and examine. This empowers users to explore the system, zooming in and out of anatomical structures as desired. Expert users can employ control keys for quick mode changes. To aid wayfinding, certain structures persist across scales (serving as landmarks). In addition, the result space is navigable by predefined viewpoints, which can be visited sequentially or randomly through menu activation. This guarantees that all content can be accessible, and users can recover from any disorientation.

PathSim Visualizer manages macro and micro scale result visualizations using proximity-based filtering and scene logic `Scripts {}`. As users approach a given anatomical structure, the micro-scale meshes and results are loaded and synchronized to the time on the users' Heads-Up-Display (HUD). Figure 4 shows a macro-scale view of the Waldeyer's ring simulation environment in context.

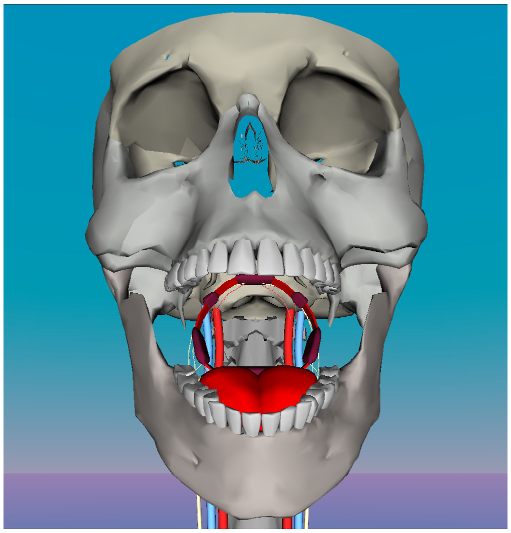


Figure 4: The generated Waldeyer's Ring
(skull model [Bogart et al., 2001] shown for reference)

3.2 Abstract Information

There is a variety of abstract information that may be relevant to a researcher investigating a digital biology simulation through PathSim. This information may be represented graphically or numerically within the virtual environment:

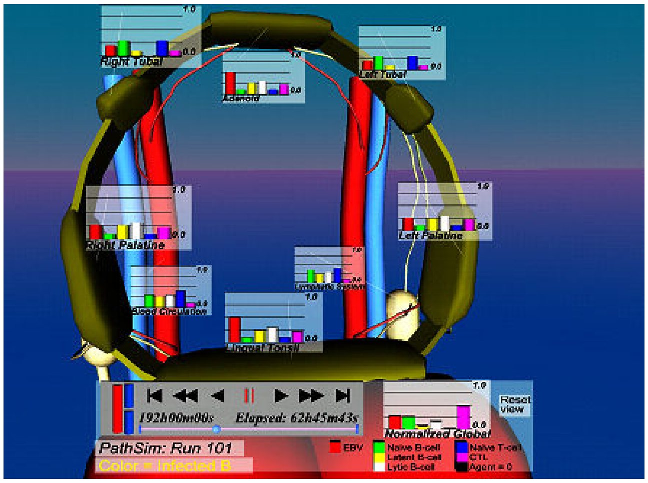
- **•Host/Pathogen populations for the system**
- **•Host/Pathogen populations per local region or unit**
- **•Annotations, hyperlinks, and references about the structure or process being evaluated**

The PathSim Visualizer implements custom software objects to manage, layout, and display this abstract information in the context of the virtual environment. These are described in detail in Section 5.

From a design standpoint, there are at least four possibilities for locating information annotations based on what coordinate system they are relative to. These are: world-fixed, display-fixed, user-fixed, and view-fixed locations [Feiner et. al., 1993; Bowman et al., 2003]. Information may be associated to a particular object in the world, which is termed as 'object-fixed'. Alternatively, the text may be associated to a location in the world, which is termed 'world-fixed'. If the annotation travels with the user regardless of their navigational actions, this is classified as 'user-fixed'; if persistently located relative to the display, its is termed 'display-fixed'.

PathSim Visualizer displays abstract information related to the simulation in user, world, and object fixed locations. PathSim Visualizer gives the user a Heads-Up-Display where system variables and global state are displayed. This HUD functions as a read-out and control panel, travelling with the user throughout the environment. Information displays in the environment aggregate data from smaller scales into suitable, object-fixed visual representations at larger scales (figures 5 - 8). This overview-plus-detail helps investigators explore and understand the dynamics of the system:

- •**HUD**- animated bar-graph for system's global population
- •**Agent Views –** Color coding of anatomy by agent type
- •**Toggle Population Display** – animated bar-graph or numerical encoding for a given structure; the bar-graph is normalized to show the region's population in proportion to the global population.
- •**Links** – hyperlinked websites, resources, and references may be rendered in additional windows (display-fixed locations)

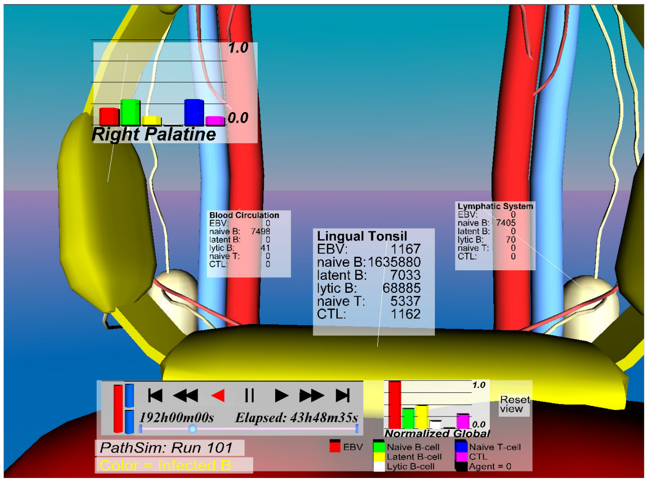


192h00m00s Elapsed: 62h45m43s

PathSim: Run 101

Color = Infected B

Figure 5: A macro-scale view of PathSim results and information displays



Blood Circulation
EBV: 0
naive B: 7498
latent B: 0
lytic B: 41
naive T: 0
CTL: 0

Lingual Tonsil
EBV: 1167
naive B: 1635880
latent B: 7033
lytic B: 68885
naive T: 5337
CTL: 1162

Lymphatic System
EBV: 0
naive B: 7405
latent B: 0
lytic B: 70
naive T: 0
CTL: 0

192h00m00s Elapsed: 43h48m35s

PathSim: Run 101

Color = Infected B

Figure 6: A macro-scale view of PathSim results with the Lingual tonsil, circulatory, and lymphatic systems toggled to numeric view (details-on-demand)

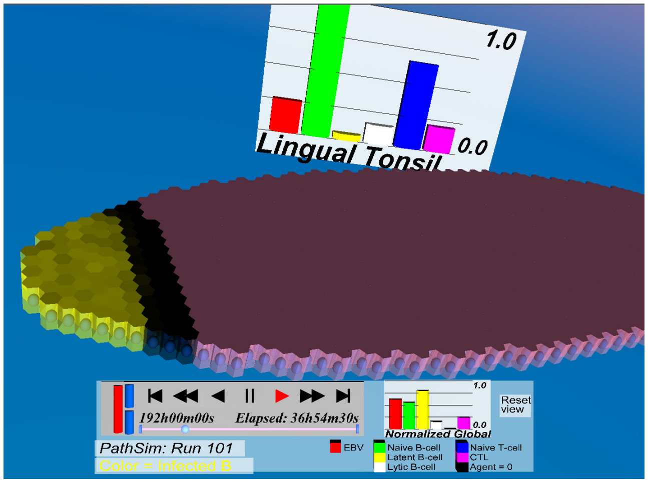


Figure 7: A micro-scale view of an infection in the Lingual tonsil

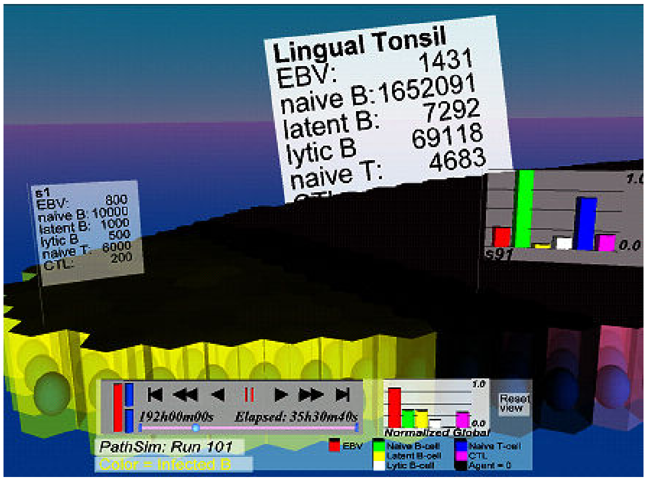


Figure 8: Zooming into the micro-scale view of the infection in the Lingual tonsil (details-on-demand)

3.3 Temporal Information

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

Investigators into dynamical systems such as the immune system need capable controls to manage and index the temporal dimension: coarse enough to find a maximum population value in a month of simulated infection time and fine-grained enough to examine behavior at 15 intervals. PathSim Visualizer synchronizes data across scales through a familiar DVD interface that gives both absolute and relative time control (adapted from NPS SAVAGE archive [2003]). Figure 6 shows a micro-scale view of the infection, synchronized to the time controller on the HUD.

4. SYSTEM IMPLEMENTATION

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

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

Currently, the PathSim Visualizer output runs fully on desktop workstations using standards-based formats such as XML, X3D, and VRML. While we have successfully demonstrated our information display aspects in Head Mounted Displays (HMDs), elumens Domes, and the CAVE, our future work is to extend the system's interactive capabilities to immersive display and input devices. For future versions, we will continue to apply the user-centered design process to identify human-computer interaction issues and visualization features for biomedical IRVE toolkits like PathSim.

5. SCENEGRAPH OBJECTS

5.1 Nested Scales

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

The HUD interface is loaded in the top-level file, which also contains ProximitySensors and Scripts to manage scene and state information. In the top-level file, a WorldGroup `Group {}` is defined that contains macro-scale models such as the body, skull. The visualization processing scripts wrap each scale model of anatomy and result animations in a `PROTO` declaration. There is one `set_fraction` eventIn on the `PROTO` interface that is processed by a TimeManager `Script {}`. Within the

Prototype declaration of each scale, the TimeManager script is connected to all sequencers and interpolaters that animate at that scale. This keeps event management encapsulated across scales and allows models to be loaded and connected to the environment easily.

As users zoom into the head and neck area and the Waldeyer's ring becomes visible, the simulation results are loaded into the WorldGroup using a `Browser.createVRMLFromString` method. The string is an `EXTERNPROTO` definition and an instance. Routes between the DVD controller and the new scene are added in order to link the scene to user global time. Similarly, as users zoom into specific anatomical structures (i.e. the tonsils), the appropriate detail geometry and simulation results are loaded into the WorldGroup as an `EXTERNPROTO` instance and Routes are added.

5.2 Semantic Objects

Recently in our IRVE research, we have implemented a set of IRVE behaviors encapsulated as Semantic Objects for VR scenegraphs [Bowman et al. 2003; Polys & Bowman 2003]. Semantic Objects are a conceptual and programmatic abstraction of spatial objects in the visual space of the IRVE that carry associated information along with their geometric and appearance information. Thus, information layout locations for Semantic Objects can all be described as object-fixed. The advantages of defining annotation information and display behaviors along with the objects are: display behaviors are in the scenegraph and operate independently of the display's size and resolution, and no central 'layout manager' is needed. This has made it possible to deploy Semantic Objects and their related information display objects across desktops, HMDs, Domes, and theoretically a CAVE as well.

5.2.1 Information Displays

Our currently defined objects for the display of associated abstract information are: unstructured text, text as field-value pairs, bar-graphs, images, movies, or audio clips. For example, the exposed functionality of the text annotation panels allows authors to specify values along each of the VRML attributes (i.e. font family, style, color etc.). In order to aid text legibility across a wide variety of scenes, text panels may be instantiated with or without a label background whose color and transparency may be specified. The text's background is automatically sized to the number of lines and characters in the MFString. Unstructured text panels are organized simply by the semantics of the `Text {String []}` field.

However, one common use for text annotations is to display an object's name and its attributes such as field-value pairs. We implemented a structured text panel that allows authors to specify a `title`, a set of `field_names`, and a set of `field_values`. Since each text string is an exposedField, textual content can be dynamically updated. Each of these parameters on text display objects gives IRVE designers flexibility to define the visual characteristics of text labels or field/value pairs across a range of environments.

In many applications, text or numeric detail has to be managed to reduce crowding and distraction in the interface. To reduce cognitive load in PathSim Visualizer, we also implemented a color-coded population bar-graph where users can get a quick, qualitative understanding of the agent populations in a certain location. To support details-on-demand, users can click the bar-graph to toggle the information display to a field-value pair text panel for the agent populations.

5.2.2 Layout Behaviors

There are a number of principal parameters on Semantic Objects. First, separate 'level of detail' groups can be defined for the object geometry and the annotation information; this insures the capability for designers to aggregate referring information independent of the object's levels of detail. Second, a Semantic object's abstract information display can be associated to the geometry by way of the Gestalt connectedness principle - such as a drawn line (i.e. Ware [2000]). Thirdly, the scaling of the annotation group is a function of user visibility and proximity with options for constant size, periodically sized, or smoothly sized. Finally, our abstract display objects act as true 3D Billboards insuring legibility from any viewing angle. This combined functionality can aid authors in mitigating the aggregation, association, density and tradeoffs in IRVE design.

We also implemented a set of Semantic Objects that vary the spatial location of the annotation group relative to the object. The display location of the annotation is a function of the user's position and viewing angle to the object. We defined 3 layout techniques in order to aid association and prevent occlusion in the annotated scene: the relative orthogonal, the bounding prism, and the bounding prism plus flocking [Bowman et al. 2003; Polys & Bowman, 2003]. While Semantic Objects' information display is always a function of user's proximity and angle, we also built a version that renders the information group only when the user toggle-selects the object. This final feature is used to pop-up (or hide) information panels for secondary anatomical structures.

5.3 MFSequencers

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

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

5.4 Heads Up Display

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

6. SUMMARY and FUTURE WORK

Through the PathSim project, we have implemented a number of custom information and interaction objects to meet the requirements of Systems Biologists to explore multi-scale, heterogeneous information. These scenegraph objects attempt to resolve tradeoffs on the dimensions of the IRVE design space [Bowman et al., 2003]. The feature summary is shown in Table 1. In the process of implementing these objects, we have discovered deficiencies and opportunities in current Web3D standards languages.

Table 1: PathSim Design Features and IRVE Design Dimensions (S=Spatial, V= Visual; E= Explicit, I=Implicit)

Information Feature	Location	Association	Aggregation
Agent Population View: Information panel per object/unit	Object-fixed	SE, VE	Low
Agent Population View: Information panel per region	World-fixed	SE, VI	High
Agent Population View: Information panel for system (global)	User-fixed	SI, VI	Highest
Time Read-out and Controller	User-fixed	SI, VI	Low
Links & References	Display-fixed	SI, VI	High

In our formative evaluations, each of the scenegraph objects described in section 5 (Nested Scales, Semantic Objects, MFSequencers, and Heads-Up-Display) has been identified as distinct and usable in our application across a range of platforms including desktops, HMDs, and Domes. Some of the functionality as implemented in VRML/X3D has known limitations (such as the HUD clipping problem). While some browsers can support 'Overlays' or rendering 'Layers' (Xj3D and BitManagement respectively), this problem can only be solved through improvement of the standard.

Some functionality, like Nested Scales is addressed by new X3D capabilities such as IMPORT/EXPORT where Inlined worlds can communicate events with their parent world. The multi-fielded Sequencer nodes fit easily into X3D paradigm and could be candidate nodes for future standardization. Semantic Objects however, provide high-level user interface behaviors that may not be best implemented in the current X3D paradigm. In conjunction with the X3D Specification Working Group, we are developing future components as a foundation to address these interface requirements. These include the Annotation Component and the Compositing Component (in progress).

Future work includes further exploring and optimizing these object/information behaviors through formal usability evaluations and proposing them as future standardization components for X3D. A proposed Annotation Component for example, would provide better support for the functionality we encapsulate in Semantic Objects. In the proposed component, associated information lives as geometry in a coordinate space parallel to the display surface; there is a reference point, an offset, and a connection point that can be connected by a lead line.

A Compositing Component would allow more sophisticated author control over rendering (i.e. Z-order, clipping, etc). This would improve support for Heads-Up-Displays, which are common in applications, but awkward between browsers. One additional functionality to pursue is an interface to define 'Application Surfaces' where the windows of other analytic tools can be mapped to a pickable 3D surface. Such functionality is extremely desirable; previously, we have implemented display-fixed prototypes using the DIVERSE tookit and XWindows for a molecular IRVE application in the CAVE [Polys et al., forthcoming 2004]. Further work in this vein must address resolve operating system, software, and hardware architectures.

For PathSim itself, we intend further integration of information resources such as published biochemical and cellular models, new multi-scale data and visualization architectures, and interface improvements such as analytic tools and indexing through the abstract data. Future bioinformatics research will involve using PathSim with other anatomical models, mesh generation techniques, and pathogen agents.

7. ACKNOWLEDGMENTS

The authors would like to thank the people involved with the PathSim project: Dr. Karen Duca, Dr. Reinhard Laubenbacher and their research team at The Virginia Bioinformatics Institute: Rohan Luktuke, John McGee, Jignesh Shah, and Purvi Saraiya.

8. REFERENCES

[1] BitManagement Contact Browser http://www.bitmanagement.de , 2003.

[2] Bogart, Bruce, Nachbar, Martin, Kirov, Miro, McNeil, Dean NYU School of Medicine "VR Anatomy Courseware" http://endeavor.med.nyu.edu/courses/anatomy/courseware/vranat/ , 2001.

[3] Bowman, D.A., North, C., Chen, J., Polys, N., Pyla, P.S., and Yilmaz, U. "Information-Rich Virtual Environments: Theory, Tools, and Research Agenda". Proc. Virtual Reality Software and Technology (VRST), 2003.

[4] Celada, Franco & Seiden, Philip "A model of cellular interactions in the immune system". Immunology Today Vol. 13 No. 2 pp. 56-62, 1992.

[5] Diaper, D. "Task analysis for knowledge-based descriptions (TAKD); The method and an example". In Task Analysis for Human-Computer Interaction, er. D. Diaper, 108-59. Chichester, England: Ellis-Horwood. 1989.

[6] DIVERSE Toolkit http://diverse.sourceforge.net , 2003.

[7] Duca, Karen, Laubenbacher, Reinhard , Polys, Nicholas F., Luktuke, Rohan, McGee, John, Shah, Jignesh "The PathSim Project", http://www.vbi.vt.edu/~pathsim, 2003.

[8] Farrell, E.J. and R.A. Zappulla, "Three-Dimensional Data Visualization and Biomedical Applications". Critical Reviews Biomedical Engineering, 16(4): p. 323-326, 1989.

[9] Feiner, S., Macintyre, B., Haupt, M., and Solomon, E. "Windows on the World: 2D Windows for 3D Augmented Reality". Proc. ACM Symposium on User Interface Software and Technology, 145-155, 1993.

[10] Goguen, J. "Information Visualizations and Semiotic Morphisms". UCSD http://www-ecse.ucsd.edu/users/goguen/papers/sm/vzln.html , 2000

[11] Grilo, António, Caetano, Artur, Rosa, Agostinho "Agent Based Artificial Immune System Genetic and Evolutionary Computation". Conference Late Breaking Papers . San Francisco citeseer.nj.nec.com/442254.html , 2001.

[12] Kling-Petersen, T., R. Pascher, and M. Rydmark, "Virtual Reality on the Web: The Potential of Different Methodologies and Visualization Techniques for Scientific Research and Medical Education". Stud Health Technol Inform, 62: p. 181-186, 1999.

[13] Naval Postgraduate School SAVAGE content archive http://web.nps.navy.mil/~brutzman/Savage/contents.html , 2003.

[14] Payne, S. & Green, T. "Task-action grammars: A model of the mental representation of task languages". Human-Computer Interaction*, 2,* 93-133, 1986.

[15] Physiome Project, The: CellML, AnatML, FieldML http://www.physiome.org , 2003.

[16] Polys, Nicholas.F. & Bowman, Doug.A., "Design and Display of Enhancing Information in Desktop Information-Rich Virtual Environments: Challenges and Techniques". Computer Science Technical Report, Virginia Tech, http://eprints.cs.vt.edu:8000/archive/00000699/ , 2003.

[17] Polys, Nicholas F., Ray, Andrew, Moldenhaur, Max, Dandekar, Chetan, North, Chris "Snap2Diverse: Coordinating Information Visualizations and Virtual Environments". Proc. SPIE Conference on Visualization and Data Analysis, 2004.

[18] Puzone, R., Kohler, B., Seiden, P., Celada, F. "IMMSIM, a flexible model for in machina experiments on immune system responses". Future Generation Computer Systems 18 Elsevier Science B.V. 961-972, 2002.

[19] Ressler, Sandy NIST: Open Virtual Reality Testbed Anthropometric Landmarks: http://ovrt.nist.gov/projects/vrml/h-anim/landmarkInfo.html http://ovrt.nist.gov/projects/cardlab/vrmlhead.htm , 2003.

[20] Rosson, M.B. and Carroll, J. Usability Engineering: Scenario based development of Human-Computer Interaction. NY, Morgan Kaufmann, 2002.

[21] Salzman, Marilyn C., Dede, Chris, Loftin, Bowen R., & Chen, Jim, "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning". Presence: Teleoperators and Virtual Environments, vol. 8, no. 3, pp. 293-316, 1999.

[22] Sutcliffe, A. and Faraday, P. "Designing Presentation in Multimedia Interfaces", Proc. CHI '94, 92-98, 1994.

[23] Systems Biology Markup Language (SBML) http://www.sbml.org , 2003.

[24] Ware, Colin Information Visualization: Perception for Design. Morgan Kauffman, NY, 2000.

[25] Ware, Colin "Design as Applied Perception". In: HCI Models, Theories, and Frameworks: Towards a Multidisciplinary Science. ed. J.M. Carroll, Morgan-Kaufmann, San Franscisco, 2003.

[26] Xj3D Open Source Web3D Tookit; administered by Web3D Consortium Source Code Group http://www.web3d.org/TaskGroups/source/ , 2003.