

© Copyright 2012
Wayne Warren

Towards an Extensible Atlas-Based 3D Visualization Framework for Biomedicine: Biolucida

Wayne Warren

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Washington
2012

Reading Committee:

James Brinkley, III, Chair

Linda Shapiro

Suzanne Weghorst

Program Authorized to Offer Degree: Medical Education and Biomedical Informatics

UMI Number: 3521577

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3521577

Copyright 2012 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature: _____

Date: _____

University of Washington

Abstract

Towards an Atlas-Based Generalized 3D Visualization
Framework for Biomedicine: Biolucida

Wayne Warren

Chair of the Supervisory Committee:

Professor James F. Brinkley III

Biomedical Informatics and Medical Education

Today's biomedical research endeavors often entail exposure to a daunting amount of expressive data, much of it spatially attributable. These data must be effectively comprehended and communicated, and it has been well established that visualization is a powerful means of doing so. Because these data are often associated with structures of anatomy that can be represented as computer-generated 3D models, it is possible to create a biomedical atlas which can be leveraged as an effective visualization motif. This style of presentation not only conveys spatial relationships inherent in the data, but also provides a natural representation which is easily understood by a wide audience. Many systems have been produced which can be considered computer-based biomedical atlases, some of which have been used to visualize data and to illustrate spatially complex concepts. However, the development of these systems has been costly due to the fact that many, while similar in features and design, have been built from the ground up as single-purpose applications. A next-generation visualization system, Biolucida, has been developed as a generalizable framework which is designed to meet the 3D visualization needs of the biomedical community through its comprehensive feature set and its extensible architecture. Biolucida has been implemented as a client/server Java-based system, which allows developers to create plugin modules which govern the functionality and content of immersive 3D visualizations, requiring only a fraction of the time it would take to develop a use-case-specific system.

TABLE OF CONTENTS

| | Page |
|--|------|
| List of Figures | v |
| List of Tables | vii |
| Chapter 1: Introduction | 1 |
| 1.1 Overview..... | 1 |
| 1.2 Problem Statement | 3 |
| 1.3 General requirements of a domain-wide visualization framework | 3 |
| 1.4 Biolucida as an extensible visualization framework | 4 |
| 1.5 Thesis outline | 4 |
| Chapter 2: Background and Context: Biomedical Atlases as Visualization Platforms | 5 |
| 2.1 The data-driven imperative of biomedicine: more efficient comprehension and communication..... | 5 |
| 2.2 Visualization has consistently proven itself as a method of presenting large datasets and complex ideas in an effective and efficient manner. | 5 |
| 2.3 Visualizations as effective communication media for biomedicine..... | 7 |
| 2.4 Three-dimensional visualization applications beyond the confines of traditional clinical imaging..... | 7 |
| 2.5 The development of content-based visualizations..... | 9 |
| 2.6 Converting scenes into data visualizations: the use of the atlas motif..... | 9 |
| 2.7 From GIS to biomedical visualization..... | 11 |
| 2.8 Overlay biomedical atlases providing high-fidelity representation of spatial phenomena..... | 12 |
| 2.9 Object-based biomedical atlas visualization..... | 14 |
| 2.10 Addressing the lack of generalizability in anatomically-based display, presentation, and visualization systems..... | 15 |
| Chapter 3: Preliminary Design Phase: Functional Requirements, Architectural Features, and Arising Challenges | 17 |
| 3.1 Underutilization of current systems..... | 17 |
| 3.2 Application domain specificity related to technology adoption..... | 17 |
| 3.3 Functional requirements determination through a survey of other systems..... | 18 |
| 3.3.1 Use-case-based requirements analysis..... | 19 |
| 3.3.2 Required features of the visualization environment..... | 20 |
| 3.4 Implementation choices and justification..... | 21 |
| 3.4.1 Use of an X3D-based VR environment for rendering and scene management..... | 21 |
| 3.4.2 Abstraction of rendering technology from the core system..... | 22 |
| 3.4.3 Model catalog selection..... | 23 |

| | |
|---|-----------|
| 3.5 Challenges to be met in implementation..... | 23 |
| 3.5.1 Creating an agile yet robust system..... | 24 |
| 3.5.2 Providing ample functionality via library elements and API access..... | 24 |
| 3.5.3 Providing support knowledgebase support | 25 |
| 3.5.4 Implementing adequate mechanisms for data attribution..... | 25 |
| 3.6 Conclusion of preliminary design analysis..... | 26 |
| Chapter 4: System Design and Implementation..... | 27 |
| 4.1 Overview..... | 27 |
| 4.2 System Description..... | 28 |
| 4.2.1 Basic Scene Construction Components..... | 28 |
| 4.2.2 Authoring/Viewing Client..... | 29 |
| 4.2.3 Biolucida Central Server..... | 29 |
| 4.2.4 Biolucida Database..... | 30 |
| 4.3 Basic Scene Construction Workflow..... | 30 |
| 4.3.1 Data Acquisition and Management Systems..... | 31 |
| 4.4 Beyond Inference: Biolucida’s Ontological Framework..... | 32 |
| 4.4.1 Concepts and the Ontological Framework..... | 33 |
| 4.4.2 Server-side ontology management..... | 34 |
| 4.4.3 Client-side ontology management..... | 34 |
| 4.4.4. Rationale of custom ontology creation..... | 35 |
| 4.5 Plugin architecture..... | 36 |
| Chapter 5: General Client Systems, Processes, and Representations | 38 |
| 5.1 General concepts of client-side extensibility..... | 38 |
| 5.1.1 Extensible client applications..... | 38 |
| 5.1.2 Representations defined..... | 39 |
| 5.1.3 Simplified plugin architecture..... | 39 |
| 5.2 Client architecture in detail..... | 40 |
| 5.2.1 User Interface..... | 41 |
| 5.2.2 Event Processing Pipeline..... | 42 |
| 5.2.2.1 Supplied default controllers..... | 43 |
| 5.2.3 Client communication systems..... | 45 |
| 5.3 Workflow Customizations: Initialization Operations..... | 45 |
| 5.4 Flexibility and Longevity Through Abstraction: The 3D Framework Architecture..... | 46 |
| 5.4.1 Detailed description of the Web3D-based 3D representation framework..... | 46 |
| 5.4.2 Default 3D framework components..... | 47 |
| Chapter 6: 3D Representation Construction Processes..... | 49 |
| 6.1 Introduction..... | 49 |

| | |
|---|-----------|
| 6.2 Client/server process execution through direct communication loop..... | 49 |
| 6.3 The main processes in scene creation..... | 49 |
| 6.3.1 Example: scene initialization..... | 50 |
| 6.3.2 Inserting Entities into the scene..... | 52 |
| 6.3.3 The event and node registration process..... | 53 |
| 6.4 Example of processes and architecture in action: the ASA Client..... | 54 |
| 6.4.1 Navigation..... | 54 |
| 6.4.2 Changes in position and appearance..... | 54 |
| 6.4.2 Changes in position and appearance..... | 54 |
| 6.5 Semantic Level of Detail with respect to SAI..... | 55 |
| 6.6 Server-side synchronization..... | 56 |
| 6.7 Creating dynamic content..... | 56 |
| 6.8 “Publishing” the Representation..... | 57 |
| Chapter 7. Using Data to Drive the Visualizatin Process..... | 59 |
| 7.1 User-driven visualizations vs. data-driven visualizations..... | 59 |
| 7.2 Variety and scope of biomedical data..... | 59 |
| 7.3 The Dataset creation process..... | 60 |
| 7.3.1 The data management process primitives..... | 60 |
| 7.3.2 Anatomy of a dataset..... | 62 |
| 7.4 Semantic attribution: aligning records with anatomical concepts..... | 62 |
| 7.5 An example of the data-driven scene construction process..... | 63 |
| 7.5.1 Illustrating the role of ontological frameworks in data-driven visualizations..... | 64 |
| 7.5.2 Native concept selection and insertion..... | 65 |
| 7.5.3 Model selection..... | 65 |
| 7.5.4 Shortcomings of simplified attribution processes..... | 66 |
| 7.5.5 An alternative and superior data-driven strategy: data propagation and subsequent scene population..... | 67 |
| 7.6 Data propagation system architecture and execution process..... | 67 |
| 7.7 Semantic Level of Detail’s role in data-driven visualizations..... | 69 |
| 7.8 Moving from propagation to inference: Data Transformation..... | 69 |
| Chapter 8: Determing Visualization Characteristics: Visualization Schemas..... | 71 |
| 8.1 Introduction..... | 71 |
| 8.2 An overview of Visualization Schemas..... | 71 |
| 8.3 The visualization schema process pipeline..... | 72 |
| 8.3.1 Relationships between VLTs, VLEs, and values..... | 73 |
| 8.4 Record resolution..... | 74 |
| 8.5 Customization of VLEs and VLTs..... | 75 |
| 8.6 Extending the functionality of the VLT system beyond simple data..... | 76 |

| | |
|---|-----|
| 8.7 Persistence for visualization schemas..... | 76 |
| Chapter 9: Results and Evaluation | 77 |
| 9.1 Introduction..... | 77 |
| 9.2 Custom Biolucida client systems..... | 77 |
| 9.2.1 The Anatomy Scene Authoring Client..... | 78 |
| 9.2.1.2 The Interactive Scene Generator (ISG) Representation – A full-featured intelligent scene generator and animation system..... | 78 |
| 9.2.1.3 The Procedural Animation (PA) Representation – A full-featured intelligent scene generator and animation system. | 82 |
| 9.2.2 The DataGen Client (DGC)..... | 86 |
| 9.2.2.1 Server-Side Components..... | 86 |
| 9.2.2.2 Compatible Data Visualization Representations..... | 87 |
| 9.2.2.2.1 The Simple DataViz (SDV) Representation..... | 88 |
| 9.2.2.2.2 The Data Browsing Visualization (DBV) Representation..... | 88 |
| 9.2.2.2.3 Display of heterogeneous media in a single environment: The Medical Record Visualization (MRV)..... | 90 |
| Chapter 10 Discussion and Future Work | 93 |
| 10.1 Revisiting Biolucida’s Purpose..... | 93 |
| 10.2 A review of the prototype applications developed as plugins to the core Biolucida system..... | 93 |
| 10.2.1 Discussion of the evaluation results..... | 93 |
| 10.3 Forward-looking research topics and design developed from further analysis of the atlas motif from the perspective of GIS..... | 96 |
| 10.3.1 Placing biomedical atlas visualizations within the context of GIS..... | 97 |
| 10.3.2 The use of a range of content in atlas-based visualization: the real, the approximate, and the high utility of the iconographic..... | 98 |
| 10.3.2.1 Abstractions can be more effective than exact depictions within visualizations..... | 99 |
| 10.3.2.2 Both geographical maps and biomedical images are subject to error..... | 99 |
| 10.3.2.3 Atlas-based visualizations as subjective content, hypotheses, and social constructs..... | 99 |
| 10.3.2.4 A discussion of information and evidence in relation to accurate and iconographic visualizations. | 100 |
| 10.3.2.5 Domain-specific applications are still relevant for the analysis of <i>information</i> to produce <i>evidence</i> | 101 |
| 10.4 Future work: research endeavors informed by the concepts of distributed cognition..... | 101 |
| 10.4.1 Effective communication: collaborative visualization through shared worlds..... | 103 |
| 10.4.2 Supporting information resource awareness: plugin resource centralization..... | 103 |
| 10.4.3 Reduction in cognitive load by specialization: Linked server domains..... | 103 |
| Works Cited | 104 |
| Appendices | 116 |

LIST OF FIGURES

| Figure | Page |
|---|------|
| 2a. Types of two-dimensional data visualizations of microarray studies..... | 7 |
| 2b. GIS visualizations using an overlay approach, and semantic attribution..... | 10 |
| 2c. Images of the EMAGE system, the Allen Brain Atlas, and Brain Explorer 2, and MouseBIRN | 13 |
| 2d. Overlay visualization in biomedical atlas-based visualization. | 14 |
| 2e. Semantic attribution in biomedical atlas-based visualization..... | 15 |
| 4a. Example Scenes From Biolucida..... | 27 |
| 4b. Workflow in Basic Scene Construction and in Data Visualization Construction | 28 |
| 4c. Components of the Central Server used in scene construction..... | 28 |
| 4d. Authoring/viewing client interface..... | 29 |
| 4e. Workflow of the Client and Central Server Building a Scene..... | 30 |
| 4f. Components of the Central Server Used in Data Visualization Construction..... | 31 |
| 4g. Selection of the Biolucida Database Relating to the Ontological Framework..... | 33 |
| 4h. Depiction of the Server-Side Ontological Framework and Concept Instance Management..... | 34 |
| 4i. The ontology editing utility provided within Biolucida..... | 35 |
| 5a. The ASA client and the Datagen client | 38 |
| 5b. Depiction of the deployment strategy and contents of Representation plugins..... | 39 |
| 5c. Two different clients, the Anatomy Scene Instruction client and the Datagen client..... | 40 |
| 5d. Two sample representations and their implemented UI panels..... | 42 |
| 5e. Schematic showing the control event processing pipeline..... | 42 |
| 5f. The grab and rotate functions of the ASA’s controllers in action..... | 44 |
| 5g. Pictorial representation of Semantic Level of Detail..... | 45 |
| 5h. Graphical depiction of Biolucida’s MVC implementation of the rendering module..... | 46 |
| 5i. Schematic of Biolucida’s 3D Rendering System Architecture..... | 47 |
| 5j. Schematic of the Web3D Scene Management Module..... | 48 |
| 6a. Schematic depiction of a typical command response cycle in Biolucida between the server and a client... .. | 49 |
| 6b. Schematic depiction of the interaction between the client and the server when an empty 3D Representation is loaded. | 50 |
| 6c. X3D node hierarchy of an empty scene..... | 51 |
| 6d. Schematic depiction of the process of adding new Entities to an active 3D Representation..... | 53 |
| 6e. The various navigation modes of a Biolucida scene. | 54 |
| 6f. Appearance change control UI in the ASA Client..... | 55 |
| 6g. Depiction of the node structure used to implement Semantic Level of Detail..... | 55 |
| 6h. Schematic depiction of the scripting and relationships between nodes when Actions in a Sequence are translated to standard X3D..... | 58 |
| 7a. Graphical depiction of the data retrieval process and its components..... | 61 |

| | |
|---|-----|
| 7b. Graphical depiction of a dataset and its components..... | 62 |
| 7c. Graphical depiction of the data attribution process..... | 64 |
| 7d. Summarized graphical depiction of the data propagation process..... | 67 |
| 8a. Depiction of a Visualization Schema Set..... | 72 |
| 8b. Schematic of the process of a visualization schema set acting upon an Entity and upon the scene itself..... | 74 |
| 8c. Description of the VLT evaluation process within a visualization schema..... | 75 |
| 8d. Depiction of the record resolution operation..... | 76 |
| 8e. User interface allowing the customization of VLT contents..... | 76 |
| 9a. The ASA Client, showing a basic scene, plane clipping, and contextual content..... | 79 |
| 9b. The ISG Representations displaying the cardiovascular system..... | 80 |
| 9c. Biolumida scene showing pin placement | 81 |
| 9d. Transitions between semantic levels of detail in the ASA client. | 82 |
| 9e. The PA Representation using a syringe as the instrument pointer..... | 84 |
| 9f. PA user interfaces allowing the construction of animations by managing discreet actions..... | 84 |
| 9g. Output X3D File with semi-transparent animation selection panel..... | 85 |
| 9h. The Datagen Client interface which allows an author to set the parameters for the extraction of a dataset which forms the basis for a subsequently produced visualization..... | 87 |
| 87 | |
| 9i. The SDV and DBV Representations showing positive gene expression for a given queried gene name..... | 90 |
| 9j. Screen captures of four of the schemas described above, depicting positive gene expression..... | 91 |
| 9k. The MRV Representation..... | 92 |
| 9l. Multiple ways of displaying DICOM images within Biolumida..... | 93 |
| 10a. Overlay and Semantically Attributed visualizations in Brain Explorer 2..... | 100 |

LIST OF TABLES

| Table Number | Page |
|---|-------------|
| 3a. Descriptions of the model sets loaded into the Biolocida prototype system..... | 23 |
| 6a. Listing of the main nodes of an empty 3D Biolocida scene/Representation..... | 51 |
| 7a. Example record set imported into Biolocida..... | 63 |
| 7b. Results of data propagation performed on source data from table 7a..... | 68 |
| 7c. Listing of data types which could produce informative visualizations using a variety of relations..... | 69 |
| 7d. Data transformation results during data propagation..... | 70 |
| 7e. Further data transformation results following initial transformations..... | 70 |
| 9a. Descriptions of datasources used for the Biolocida prototype of the DGC client..... | 87 |
| 9b. Data extracted by the Practice Fusion Data Handler..... | 91 |

ACKNOWLEDGEMENTS

I wish to express my deep and sincere gratitude to my advisor, Jim Brinkley, for his support and encouragement throughout the process of bringing this undertaking to fruition. I am grateful to the members of my committee for their willingness to devote their time to the guidance of my efforts and I would also like to thank my family for their support and patience over what has proved to be a long but rewarding endeavor.

DEDICATION

This work is dedicated to our teachers and mentors, to those who take their time and energy to impart not only their knowledge and experience onto those who need comprehension, but also their patience, guidance, and caring to those who need so much more.

CHAPTER 1

Introduction

1.1 Overview

Today's biomedical research endeavors often entail exposure to a daunting amount of data, the digestion of which is often intractable not only because of its sheer volume [114], but also because current research protocols offer highly expressive results which demand more sophisticated analyses than can be performed without equally sophisticated tools. Significant biomedical research endeavors are no longer limited to investigating the behavior of a single biomarker molecule in vitro. Instead, today's researchers employ methods which yield data characterizing the behavior of thousands of entities throughout an organism, and therefore the extraction of salient insights is challenging and often takes a great deal of time and resources.

The need to analyze data more effectively is second only to the need for more efficient and expressive communication. From new surgical procedures to descriptions of complex physiological phenomena, the modern biomedical researcher has a need not only to comprehend, but also to communicate complex concepts with both colleagues and with the public. There is a prominent opinion in biomedicine that informaticists must develop ways to better accommodate both these current needs and those yet to come [74], and to enable communication and analysis in a manner where the speed of comprehension is markedly increased. Doing so would not only allow researchers to speed up the pace of innovation, but also to gain insights which would otherwise remain hidden.

It has been well established that visualization is a powerful conduit for analyzing, representing, and communicating complex concepts and dense datasets. Visualization techniques have been applied to a growing need for searching through dense collections of information for the purpose of conceptualizing relationships among the data by filtering via visual cues, thus revealing trends that would have otherwise gone unnoted. Such "visual data mining" activities prove especially useful when connected to real-time interactive interfaces which facilitate the search and filtering processes. The field of computer graphics applied to visualization has given the biomedical research community a collection of tools with which it can represent data using a number of styles, as it has traditionally done with figures, graphs, and tables. However, static 2D visualizations lack the illustrative power to sufficiently describe concepts which have precise and expressive spatial characteristics, such as in the case of illustrating the functional activation patterns of the brain while performing a particular task, animating the choreographed steps of a surgical procedure, or the displaying graphical output of a physiological simulation.

One of the more well-established styles of 3D visualization, supported by decades of developments in medical imaging and grounded by our familiarity with the visible structures of the body (and of model organisms), is the biomedical atlas. The traditional view of the biomedical atlas has changed quite a bit throughout its history of more than 450 years. Early on, beginning with the printing of Vesalius' *De Humani Corporis Fabrica*, published in 1543 (Cushing, 1943; Saunders and O'Malley, 1950) early atlases primarily had an anatomy education focus, with

particular interest in the human organ systems. The atlas has since evolved beyond the printed classics penned by Henry Gray and Joseph Hyrtl [60, 44] to the realm of computer graphics and animation, significantly increasing an atlas' capability to impart information onto and offer interaction with the user. The use of such atlases has been able to grow beyond their traditional confines within a hospital's radiology department, to the biomedical research community through innovations in computer graphics which have raised the capabilities of typical desktop machines [51].

Whereas 3D display capabilities were previously the domain of only high-end professional workstations, such as those used by radiologists and CAD designers in the 1980's and early 1990's, later developments in more affordable hardware acceleration and available software libraries have made personal computers effective 3D display platforms. The application of 3D computer graphics to the rendering of scenes of human anatomy on average desktop computers was largely pioneered in the 1980's, and has since produced a number of computer-driven atlases of anatomy. Originally, these applications were distributed with predetermined content and curricula via CD-ROM [1, 42, 54, 92, 108], and while somewhat interactive, were composed of static content which was, for the most part, "what you see is what you get". To effectively utilize such technology in the biomedical research field required more customizable content, interactive environments, and the creation of a new form of visualization that puts a new perspective on an old art: *interactive* 3D biomedical atlases.

Many in the biomedical community tend to relate biomedical atlases to collections of images for educational purposes, an interface to a framework of information organized and accessed spatially or pictographically, or simply a collection of illustrative representations. The transition of atlases from pen and ink to computer graphics has led Biomedicine to create atlas-based applications serving a number of purposes: education, data analysis via visualization and as query interfaces to datasets. Electronic atlases of all kinds, from compendiums of radiological images to collections of representations of non-human model organisms have been built to serve a variety of users throughout the biomedical research community.

A number of recent atlas implementations have been deployed through Virtual Reality (VR) based systems. The release of graphics libraries such as OpenGL [150] and Direct3D [11], have allowed those with enough technical knowledge to access low-level 3D display APIs. Software developers, informaticians, and computer scientists have responded by creating a variety of systems which produce 3D representations of a wide array of different datasets. VR systems (or interactive 3D systems, for the purposes of this discussion), offer a method of implementing both sophisticated visualization techniques and highly interactive control over a customizable 3D world. These VR systems not only offer free navigation as well as interaction with the scene's contents, they can also display a variety of multimedia data, offering a degree of flexibility which suits the growing types of data which are found within the biomedical information landscape. The potential returns of properly utilizing these technologies within biomedicine are well documented.

While there exists a common theme between the biomedical atlas applications developed to date, their use of content and implementation strategies lack a strong sense of consistency. Therefore the adoption of such systems has been

slow, a phenomenon possibly due to the high level of effort required to develop the software and the learning curves associated with mastering their use. The biomedical informatician suffers from the lack of systems and frameworks which allow the construction of visualization workflows easily [125]. Without in-depth knowledge of scenegraph APIs and proprietary syntaxes associated with lower-level APIs, the construction of a 3D visualization application is a difficult, if not impossible task.

The existence of these challenges is surprising, given the amount of previous work completed in the informatics field. One would expect that the development of atlas-based visualization systems to be serial in nature, with earlier projects powering the advancement of more modern systems through the re-use of their model content, rendering technology, and scene management modules. This, however, has not been the case. Software developers have since made less than ideal gains in the advancement of the field by developing systems in a semi-parallel nature, without the reuse of code or the leveraging of common libraries and frameworks. This phenomenon has occurred despite the existence of such toolkits [36].

This fragmented development strategy has negative effects on the users as well as the developers. The user, or biomedical researcher, experiences the costs of learning to use tools which have been developed as approaches to address only very specific biomedical questions. As very specialized applications, these systems are limited in their content and scope and can be used to analyze only specific types of data. Moreover, the same software cannot be leveraged to help communicate these findings through the production of animations and other helpful media and cannot be applied to other model organisms or even different anatomical structures than what the system already contains. Thus, one may have to become familiar with a number of visualization tools to investigate research topics in any given biomedical discipline.

1.2 Problem statement.

Clearly, the biomedical research community would benefit from a common and generalizable visualization system which has a low barrier to entry from the perspective of both the developer and the user. Additionally, the system should have the appropriate expressivity and extensibility to apply itself to the application needs of biomedicine's subdomains: education, task-based training, data visualization, and simulation. Biolumina is a biomedical atlas-based visualization framework that aims to meet these requirements.

1.3 General requirements of a domain-wide visualization framework.

Clearly, the challenges to building a domain-wide biomedical visualization system lie in creating a 3D application framework that is: (1) expressive – it can display whatever content is needed, (2) accessible – the system must be easy to work with in terms of making extensions to its functionality, (3) flexible – the system can be changed to adapt to new technologies, content, and data to ensure longevity, and (4) have wide appeal – the system should be functional on a wide array of platforms and should be easy to operate for its end users. The system's architecture

will also have to exhibit features which enable the level of agile development required to handle a potentially massive number of individual application needs.

1.4 Biolucida as an extensible visualization framework.

Because Biolucida aims to serve as a generalizable atlas-based visualization system, yet a central development team cannot be aware of all possible current and future application needs of the biomedical research domain, Biolucida is deployed using an extensible plugin architecture, where a great deal of focus is placed on the design abstractions which make the system's core functionality accessible, yet plugin development is constrained in ways that guarantee the core system's reliable operation. Several plugin visualization systems have been developed as proofs of concept. Each of these visualization clients differs in the data it can represent, as well as in the features and tools that it offers the user. Together, they represent a broad demonstration of Biolucida's capabilities.

1.5 Thesis outline.

The following chapters describe the significance of this work, Biolucida's implementation details, and the results of these efforts. Chapter two describes previous works in the field of visualization, which are precursors to and seminal examples of atlas-based visualization. Chapter three outlines the requirements for a generalized atlas-based biomedical visualization system, and goes on to describe the challenges associated with the implementation of the proposed software. Chapter four illustrates the architecture of the Biolucida system at a very high level, while chapter five gives a more specific description of the client and the characteristics that enable the creation of different client permutations to meet a variety of application needs in biomedicine. Chapter six contains detailed descriptions of the components and processes of the client and server involved in the creation of virtual anatomical scenes, and other core operations in the visualization process. Chapter seven describes the operations on the server that are responsible for the processing and handling of datasets which drive the production of visualizations, and chapter eight elucidates the mechanisms through which developers and users control the attributes of those visualizations. Chapter nine describes the development and evaluation of four types of visualization plugin systems which leverage the Biolucida core as a toolkit. Finally, chapter ten discusses the broader significance of this work, philosophical considerations, and possible future work to extend Biolucida.

CHAPTER 2

Background and Context: Biomedical Atlases as Visualization Platforms

2.1 The data-driven imperative of biomedicine: more efficient comprehension and communication.

The pace of biomedical research data production has evolved to the point where the amount of information to be processed, represented, and comprehended by the research community each year has become very daunting. In many disciplines, the volume of data being produced by today's high throughput protocols, imaging methods and meta-analyses has grown far beyond our ability to comprehend these findings unaided. No matter how fast our instruments and computers can analyze and present these data, the pace of discovery will be limited by the viscosity of uptake inherent in human users. Moreover, there exists a compounding problem of dealing with the cumulative data of any discipline, since researchers are often expected to be familiar both with seminal legacy findings and recent advances in the field. As the volume of data grows with each progressive step forward in the perpetual march toward discovery, so do the barriers to comprehension and an understanding of provenance which is critical to gaining new insights.

In addition to the increasing volume of research data, the complexity of the biomedical information landscape has increased considerably as well, especially when one compares the types of data produced today versus those of thirty years ago. Whereas researchers in the past were focused on endeavors such as translating nucleotide and protein sequences, we find ourselves now attempting to interpret the spatial distribution of biomarker molecules and inferring relationships among hundreds of genes which were sampled simultaneously. So, not only is there more information to digest, interpretation of these more expressive data has become a burdensome task as well. Clearly, biomedicine would benefit significantly from innovations in the way in which we manage, communicate, and analyze such information.

Endeavors which seek to augment our ability to understand large amounts of data have been cited as keys to future progress in biomedicine [26]. One author states that "the methods of portraying ... [biomedical research] results in ways that researchers can understand and interpret need improving" [109]. A possibility for such improvements exists in how the data are represented, because proper modes of presentation will not only increase the rate of information absorption, but will also augment the effectiveness of analyses and the degree of comprehension. Norman et al. recognize that our ability to describe various concepts is limited by the functionality and expressivity of the cognitive artifacts we use to represent them [100]. Like many of its fellow domains, the biomedical sciences have traditionally relied on figures, images, and tables to communicate complex information in publications. While these methods have sufficed for some time, there is now a demand for more expressive modes of illustration.

2.2 Visualization has consistently proven itself as a method of presenting large datasets and complex ideas in an effective and efficient manner.

Due to its slow serial-based method of absorption in the brain, the limitations of communicating information solely by narrative and discreet text are well documented [75]. Visualization is a high-speed transport mechanism for information to enter the mind, and it is especially efficient in allowing a user to appreciate densely populated information spaces [133]. This is because visualization of information reduces the cognitive load on the viewer, transferring much of the processing required to absorb information from the human cognitive system to the perceptual system. A reduction in the viscosity of uptake that we normally experience with our absorption of text is achieved by leveraging the parallel processing power of the human visual cortex [106], allowing users to devote more attention to understanding and analyzing the data. Moreover, a well-designed visualization has a high capacity to reveal trends in data because it can limit the information presented to the viewer to only that which is relevant.

The information visualization discipline has a well-established history using a variety of two-dimensional illustrative techniques to communicate insights from a chaotic dataset. Graphs, figures, and illustrations have been used to communicate insights for a majority of the existence of our scientific disciplines. There are several examples of projects which have proven their effectiveness in using visualization in such areas as software engineering [35, 34], and more arbitrary analytics [2, 80, 128, 137, 139, 143, 148]. While visualizations can be more effective than any other method of presenting the same data, this claim is especially true when the data lends itself to representation by spatial metaphor. Interestingly, the aforementioned types of data produced in modern biomedical research protocols lend themselves to representation using such spatial metaphors. Yet, the needs of modern biomedical researchers have pushed beyond simple two-dimensional representations, presenting a need for 3D illustrations and time-series animations revealing the salient characteristics of more complex sets of information. Therefore, informaticists are presented with the challenge of understanding in what manner 3D representation best illustrates a number of types of research data. As Johnson stated, a “top” problem in scientific visualization is the integration of scientific research and information visualization methods [69].

A perfect example of the aforementioned advances in biomedical research methods is the development of the microarray, a protocol which yields a relative analysis of the expression levels of thousands of gene products from a tissue sample in one experimental “run”. Such data are valuable in understanding clusters of genes which are differentially expressed (compared with a control) in samples representing disease states such as cancer, viral infection, and toxicity conditions. This information is extremely valuable for subsequent analysis, but its sheer volume requires that it often be comprehended by proxy – using some type of illustration which highlights features that differ significantly from the rest of the population and therefore may reveal potential conclusions. However, these abstract visualizations (such as those in figure 2a) carry the risk of ignoring critical insights that can be gained through the appreciation of not *what* data was produced, but *where* it came from: its spatial and/or semantic dimensions. Within these data is not only information describing numerical measurements, but spatial and temporal information which are best represented in a manner which departs from the established methods of data visualization.

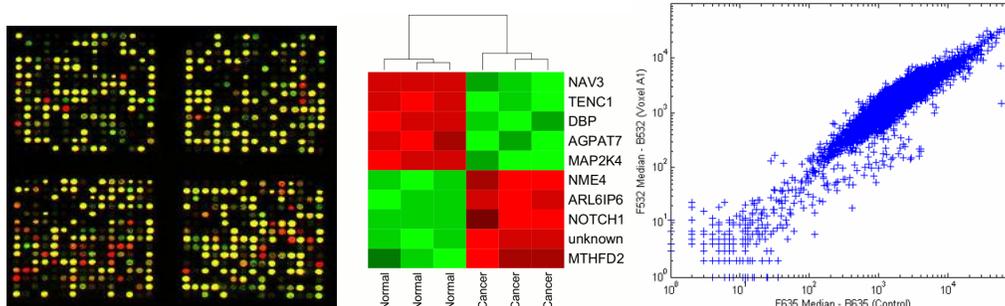


Figure 2a. Types of two-dimensional data visualizations of microarray studies.

2.3 Visualizations as effective communication media for biomedicine.

Visualizations can serve not only as a means of analysis, they can also become a method of communicating insights to an audience. Moving beyond molecular investigations, one finds that both progress in technology and shifts in culture have presented a need for many types of 3D visualizations which enhance communication between physicians, colleagues, and patients. Video and animation can be a very powerful method of *showing* rather than simply *describing* the details behind complex concepts [51]. Animations of medical procedures have been proven to enhance the trust felt between physicians and their patients [53] when shown in an effort to educate and obtain informed consent. There are also many well-documented cases of the effectiveness of sharing visualizations of procedures between colleagues [110, 68, 79] to gain a better understanding of new techniques. Lastly, one cannot ignore the increased attention which new 3D technologies have received within the domain of medical education, mostly in anatomically-based instruction [22, 122, 55, 90, 24, 147, 78]. These modes of communication and learning can also be applied to the task of presenting results from the aforementioned microarray studies, which may require an appreciation for spatial, time-dependent, and otherwise complex concepts when communicating with colleagues.

The evolution of 3D computer-based visualization technologies in research has been marching forward since the 1980's where pioneering systems such as the Application Visualization System (AVS) [136] and IBM's OpenDX [63] implemented the earliest examples of 3D scene rendering for data visualization purposes. To understand the power that such visualizations can bring to biomedical research, one must specifically grasp the importance of spatial analyses to the research field, as there are many applications which are best served through the use of 3D representation. Applications such as the calculation of proper angles of radiation beam paths used in the treatment of tumors [30], calculating penetrating wound trajectories in trauma cases [12, 104, 105], genetic analyses of both human and model organisms [99], and the visualization of simulation models [29] all involve 3D spatial analysis. Fortunately, some areas of biomedicine have already established 3D visualization as a core part of their discipline.

2.4 Three-dimensional visualization applications beyond the confines of traditional clinical imaging.

Three-dimensional visualization in medicine has become a key technology in clinical imaging, where the analysis and presentation of image data from acquisition methods such as CT and MRI [50] have spawned an entire

discipline devoted to the visualization of human anatomy to enable a higher standard of care. However, visualizations of these data have traditionally been very specialized in their scope due to the highly capable hardware they require to render images and the limited audience that these visages were designed to serve, specifically surgeons and radiologists. The high level of resources required to display 3D medical images arises from the expressive manner in which they are represented and rendered, compared with the simpler way in which general 3D computer graphics are handled.

Two types of 3D rendering strategies most commonly in use today [123] are volume rendering and surface rendering, found most commonly in medical imaging and general graphics applications, respectively. In volume rendering, the visualized space is represented as a matrix of three-dimensional cubes, or voxels, as in the case with CT and MRI data where serial sections of 2D image data have been interpolated to produce the voxel matrix. Rendering a view involves establishing the position and orientation of a virtual “camera” and casting a series of rays for the displayed image’s pixels - lines of sight between the visualized content and the camera. Surface rendering, associated with more general 3D graphics applications such as simulation and gaming, involves representing and rendering only the surface of 3D shapes. A “mesh”, consisting of points in 3D space and the edges which connect them, produces an array of polygons which are imbued with appropriate appearance attributes such as color, texture, and transparency. The resulting mesh is shaded by a rendering algorithm which considers the angle and distance of light sources to their constituent polygons in the production of a final image [76].

Volume and surface rendering differ greatly in their demand for processing power when rendering images to the user’s screen. Volume rendering requires a great deal more time and resources to produce images [134], where interactive viewing of these 3D datasets may not be possible on some platforms. Interactive, or “real time” animation and navigation require that a 3D display system render at least 24 frames per second, so that the user has the impression of smooth movement, rather than perceiving changes in screen content from one frame to the next, just as in viewing a video or film. Such is the cause for medical images being confined to specialized hardware, that is, until fairly recently.

Advances in the graphics capabilities of desktop computers [20], innovations in software engineering, and the increased availability of medical image content have catalyzed the creation of many software systems designed to allow biomedical researchers, educators, and even patients to view 3D medical image data for a variety of purposes on common hardware. A number of such software projects have brought valuable tools to the research community in the last twenty years [115]. In the late 1980s, general 3D volume viewing tools such as the commercial package Slicer Dicer from Visualogic [126], funded by Apple Computer and the NSF, allowed a user to view many types of 3D data including clinical images. Interactivity and control over the rendering was limited, but the ability to view such datasets on an average machine represented a quantum leap forward at the time. Subsequent software projects catered more to the biomedical field’s needs, such as offering tissue coloration and interactive threshold control. Slicer [111] and Osirix [132] are two popular 3D image rendering packages used in biomedical research which incorporate some of these features. Still, other systems changed the manner in which the application could be

delivered to a user [84], or the way in which the user was able to manipulate and interact with the displayed 3D image [67].

2.5 The development of content-based visualizations.

Equipped with the ability to display and interact with virtual anatomical structures in 3D, software developers have produced systems which addressed very specific challenges in biomedicine, mostly in the area of medical education to start. A 3D scene of a virtual human body offers the possibility of replacing (or at least reducing) the resource-intensive practice of classical dissection using cadavers. Yet, applications which seek to present didactic material on the subject of anatomy must be deliberate in regard to the content presented to the user. No longer implemented as simple viewers, these systems met the challenge of delivering both a set of content, and a means for display and interaction.

Some anatomical instruction/representation systems such as EVA [110] were designed to focus on a particular area of the human body. Still, others would seek to widen the scope while constraining the amount of detail, such as The Visible Man [73], and the 1993 prototype that would eventually become the well-known Voxel-Man project [121, Schiemann, 2000, 56]. More recent endeavors have focused on creating more realistic renderings [47], or on providing a more comprehensive set of features and fine detail [103]. Other anatomy scene generators sought to enable the user as an author of more customizable scenes. Both the Dynamic Scene Generator [149], and the SCAN project [19] used a knowledge-based model which allowed for user-specified inclusion of various surface models representing anatomical structures, from clinical sources and the visible human female, respectively, into a fully navigable scene.

More advanced anatomical visualization systems sought to go from simply representing a fixed library of canonical content which may have limited utility [86], to introducing a measure of realism by incorporating the concept of anatomical variability from one individual body to another. Such representations were attempted through predefined discreet changes in morphology [102], or by implementing parametric models which use randomness to create slight dissimilarities between instances [46], as suggested by Trelease, et al [131].

2.6 Converting scenes into data visualizations: the use of the atlas motif.

While the aforementioned systems can produce some rich and informative scenes of human anatomy, it is our opinion that they don't qualify as visualizations in their own right. A visualization is a graphic depiction which uses visual metaphor to represent the attributes of underlying data. While not qualifying as visualizations per se, the aforementioned projects' ability to display 3D anatomical objects have contributed to the evolution of a form of data visualization using the structures of both model organisms and humans as a framework on which to illustrate the characteristics of attributed datasets: biomedical atlases. Looking beyond the borders of computer science and graphics, maps and the collective works of atlases have been used to produce some of the most effective visualizations in human history, and much can be gained by applying their concepts to creating both software

systems and the types of visualizations that the biomedical research community clearly needs. A number of recent successful system deployments within the biomedical informatics community have been somewhat based on the atlas motif [138, 38, 77]. However, we believe that a closer look at the acclaimed assets of cartography and GIS can provide necessary guidance for the creation of still more effective visualizations.

Maps, used in human society for thousands of years, are visualizations which use the arrangement of various items of interest to convey their relationships primarily through the use of a spatial metaphor [8]. Traditional maps depict regions of land, countries, and cities as the items of interest, while distance and direction can be thought of as data attributed to those items. Illustrations such as these mimic the previously discussed anatomical scenes produced by educational and radiological applications, where the concept of “place” equates to anatomical structure. However, a map’s power can be augmented by the inclusion of several other dimensions of data attributable to the “mapable” items of interest – data such as population, climate, and gross domestic product.

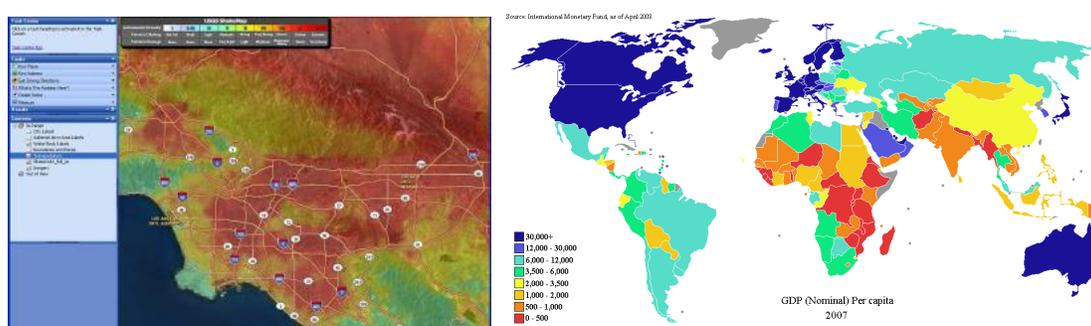


Figure 2b. GIS visualizations using an overlay (left) approach, and semantic attribution (right)

Maps can be thought of as substrates for visual analytics when their spatial data are combined with other information, adding additional dimensions. These foreign data can be correlated with the map in two very distinct ways: via coregistration or by object-based attribution. The former method involves overlaying an image containing spatially-oriented data, such as a weather satellite photo, onto a map representing the same physical area. The combination of the two sets of data produces an informative visualization from which a viewer can glean insights, and in this case, of the weather patterns of various cities, states, and countries. Object-based attribution involves the association of data directly with regions of interest within the map, treated as whole entities (i.e. countries, states, etc. – the objects). The retinal attributes (color, transparency, etc.) of these entities can be used to represent information that is attributed to the spatially designated entity. This style of data visualization offers tremendous flexibility because the data need not have any inherent spatially relevant information by themselves. Only a reference to a spatially relevant entity in the map is needed. Such is the case when information such as disease incidence rates for various countries is represented via coloration. As a result, non-spatial datasets are given spatial significance.

When visualizations are formed from the convergence of a map with abstract data, they can make for very effective representations. In object-based maps, a level of summarization and simplification is offered via the semantics. In

the case of a GDP map, the countries have now become icons that simply refer to the GDP data, conveyed by coloration. Visual representation of these datasets makes appreciation of trends and relationships almost trivial, and has led to the formation of an entire discipline, Geographical Information Systems, or GIS. GIS allows the illustration of various datasets using the framework of the earth's surface and its features – physical, political, and otherwise. The use of common geospatial references can integrate otherwise heterogeneous data, allowing the viewer to perceive relationships among previously unrelated information, and appreciate general trends in these data which may otherwise lay beyond cognitive reach.

2.7 From GIS to biomedical visualization.

It can be argued that the types of analyses and representations required of the biomedical research community today necessitate the use of tools like maps, but in an entirely different scope. The concept of “location” has become much more complex than in the days of John Snow, who is best known for his novel use of primitive GIS methods to identify the contaminated water wells which spread cholera throughout London in the 19th century [70]. Physiology and microbiology have created their own microcosmic definitions of “location”, necessitating the analysis of spatial trends involving the body and its tissues. It is important to note that biomedicine is not entirely unfamiliar with such artifacts. Chromosomal and plasmid maps, medical illustrations depicting various aspects of human physiology, and even diagnostic scanning technology could qualify as samples of modern biomedical “maps”. The concepts of modern biomedical maps and atlases have been brought to fruition in a number of recent development projects. Those projects that have the most in common with modern GIS techniques involve the fusion or overlay of multiple datasets.

Technological advances in medical imaging have yielded a number of different acquisition techniques, each designed to reveal different information about the structure and function of anatomical area of interest. It is common practice in cancer treatment to merge a set of multimodal images such as CT and PET scans, in order to establish the treatment volume of a tumor [61]. Reminiscent of our aforementioned map and satellite photo combination, the different modalities contribute structure, tissue attributes, and metabolic information to a composite image which allows a clinician to determine the appropriate treatment volume for a tumor. This is a task which is not possible to accomplish by reviewing any of the images by themselves. Other projects have integrated less common data into the fusion process, producing novel visualizations. For example, MEDIMAG is designed to merge simulated electrical fields with an anatomical atlas [49]. These examples of data fusion mimic image overlay practices that form the foundation of GIS research. Biomedicine, however, has additional challenges which are not encountered within GIS.

In the above biomedical examples, all data are taken from the same patient. Therefore, while proper registration of these images is critical to the composition's accuracy, it is a tractable task. Biomedical atlas visualization begins to depart from the foundational concepts of GIS when visualizations begin incorporating images taken from different subjects. While there is one canonical representation of the surface of the earth, there can be drastic differences in morphology between individual organisms. Such anatomical variation occurs less in simpler model organisms, but

is more common in higher primates, as they exhibit an appreciable degree of morphological heterogeneity making proper image registration a challenging task. However, a great deal of insight can be gained from visualizations which combine images from different subjects, as noted in Barrillot's work [6].

Visualizations involving data taken from different subjects are currently produced in both the domains of developmental molecular biology and clinical medicine, involving model organisms as well as human subjects. The action of warping a set of models so that the morphology of one organism matches another's is a common approach to achieving proper registration between two individuals. However, many of the analyses we see today involve samples from a great number of individuals. A common approach for the combination or comparison of data from such a large group is to register the images to a canonical set of morphology, representing an "average" organism. Implementations of both approaches are utilized in a number of projects that create atlas-based visualizations which combine data from disparate individuals.

2.8 Overlay biomedical atlases providing high-fidelity representation of spatial phenomena.

A great deal of biomedical research protocols result in datasets which contain raw spatial information, such as images and point clouds. Some of the most prominent examples of these data are *in situ* hybridizations and fMRI studies. These types of data require an overlay and registration approach to produce an atlas-based visualization, unless either computer vision technologies or human intervention is employed to attribute the sampled points in the data to the name of the anatomical structure which that point in space represents.

A recent explosion of image data taken from *in situ* hybridization (ISH) studies which reveal the distribution of expressed gene products throughout an organism at a point in time has given researchers the ability to understand the actions of the chemical cascades which are responsible for proper morphological and physiological development of an embryo into a functional organism. Recently, there have been launches of many systems which allow the management, analysis, and visualization of these data, from whole-mount protocols in which the entire organism is visualized [38] to those with resolution at the cellular level [145].

One of the most prominent examples of the combination of anatomical models with ISH molecular studies is the EMAGE project [4], and its derivatives. The EMAGE/EMAP system, based on the mouse model *Mus musculus*, combines a comprehensive developmental Theiler stage atlas [129] of canonical 3D models derived from optical projection tomography (OPT), an ISH image repository, and tools which allow contributors to specify arbitrary regions of gene expression by "painting" data from their research findings onto scanned serial sections from which the Theiler atlas is derived. The resulting 3D regions are stored in a searchable database which now allows for spatial querying [138] and arbitrary geometry-matching capabilities [113] which allow a researcher to draw a region of interest and initiate a search for expression patterns which most closely match the query shape.

The development of the EMAP/EMAGE system has given the research community one of its first examples of a common biomedical atlas framework on which novel types of visualizations can be built by extending aspects of the

system. The EMAP/EMAGE system has therefore spawned a number of other OPT/ISH projects which seek to model the developmental gene expression of other model organisms such as the chicken, the *medaka* killifish, the *Xenopus* frog, and zebrafish [27, 9]. The BIRN project, a multi-institution collaboration centered at UCSD, has developed a mouse atlas system [17, 91, 153] which, like EMAP and EMAGE, visualizes mouse anatomy in both 2D slices and 3D reconstructions and gives the user the ability to view gene expression patterns which reside in the BIRN database, or from additional datasets which can be uploaded at runtime. Going a step further, several projects have applied the same approach to the analysis of human development. OPT/ISH studies using human embryo material have analyzed gene expression in the brain [120, 85], producing informative visualizations as a result.

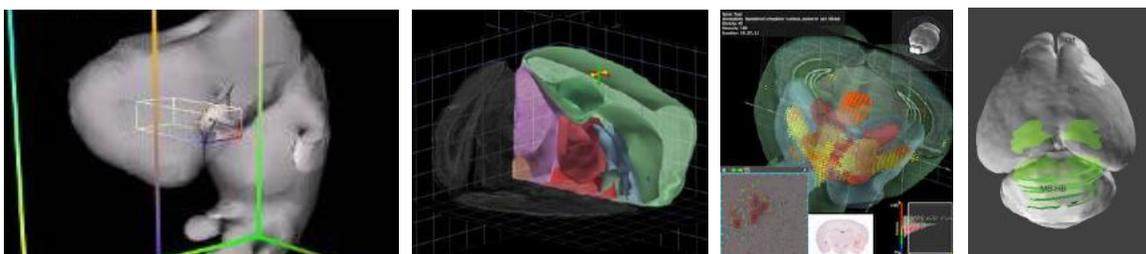


Figure 2c. Images of the EMAGE system (left), the Allen Brain Atlas (center-left), and Brain Explorer 2 (center-right), and MouseBIRN (right)

Another ISH-based atlas visualization system, the Allen Brain Atlas project [25], has created its own visualization platform which takes a very similar approach to the EMAGE system but uses a greater number of experimental protocols to derive gene expression data which focuses on the neural development in mice and in humans. Like the EMAGE project, the Allen Brain Atlas has spawned its own set of spin-off projects which use the Allen Brain Atlas framework to produce novel visualizations, including applications which offer similar spatial search capabilities [83].

Any discussion of atlas-based visualization systems would be remiss without the mention of the innovations made by functional MRI (fMRI) visualization software projects. Based on the idea that imaging the increase/decrease presence of water can be analogous to measuring blood flow, fMRI images the fluctuations in activity of various areas of the brain while the subject is alive, awake, and functional. Elimination of the requirement of euthanization, or even sedation of the subject, allows for these measurements to be taken while performing various tasks which help answer questions regarding which areas of the brain are activated during the performance of different types of activities. Systems, such as BrainVoyager [41] and ImageSurfer [37], are among many systems which create visualizations fusing multimodal data with an atlas of the brain's anatomical structures, as well as provide tools for the query and analysis of the data in the visualization.

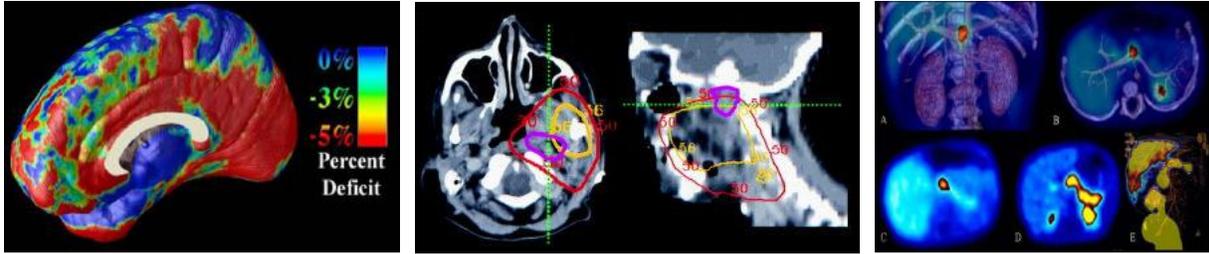


Figure 2d. Overlay visualization in biomedical atlas-based visualization. An fMRI image fused with atlas geometry (left), a clinical tumor volume composite image (center), and a multimodal image composed of CT and PET scan data (right)

2.9 Object-based biomedical atlas visualization

While there are many projects which focus on fusing image data to create an overlay presentation, showing insights through the spatial correlation among the datasets, other systems take a more object-attribution oriented approach. This approach, referred to in this work as *semantic attribution*, endeavors to assign data to discrete entities in the scene. Instead of simply registering a satellite photo showing heat distribution over a political map, *semantic attribution* might assign average temperatures to zip codes or counties, treating these as discrete entities in the resulting visualization. One can see how such an approach would be appropriate for a biomedical atlas, especially in the case where its contents were composed of discrete surface-based models.

Many biomedical visualization projects have taken the approach of using *semantic attribution* in the creation of atlas-based visualizations. IBM's Zurich lab uses an atlas-based visualization as an interface with the electronic medical record [62]. The project is attempting to use technology similar to Google Earth [43] as an exploration interface for the human body. This would allow physicians to interact with a visualization of data found in a patient's EMR by constructing a visualization which attributes appropriate information to the corresponding anatomical structures. Likewise, HealthCyberMap [72] is a system developed to visualize available information resources containing information related to various concepts in the human body. It uses an iconographic depiction of the human body, to which color is applied to represent the area of relevance and the number of sources of information available to the user.



Figure 2e. Semantic attribution in biomedical atlas-based visualization. Visualizations from from HealthCyberMap (left), IBM's visual medical record system (center), and 3D choropleth rendering showing Garb's discreet operative failure rate regions of the human colon.

Garb, et al, used GIS methods to visualize failure rates, or lesion frequency, of TEM surgery for colorectal polyps and the locations at which the polyps were present [39]. This system was responsible for mapping the data from over 4000 patients to a universal coordinate system. The authors coregistered data from over 4000 patients by dividing the rectum into four identifiable anatomical regions, and summary data was collected and visualized by attributing the frequency of lesions to given regions. This gave the plot a somewhat object-based style which circumvents the difficulties posed by anatomical variation between patients, yet still allows the viewer to appreciate trends in the data. ANET, an automated system for calculating blood flow through microvascular networks [118], was designed to model the damage to microvasculature following ionizing radiation treatment for cancer. Scenes were assembled via digitizing frames from fluoroscopic video images. Data was attributed using an object-based method where section of the vascular network was treated as a node which was populated with attributable data from an external simulation database. Bartling, et al, have created a system that is capable of building a scene using individual patient anatomic structure to create annotatable dental charts [7], implemented in a GIS system. The anatomical entities within the chart are populated (or annotated) with clinically relevant information, thus producing a complete patient chart which, rather than being represented using ideal anatomy which does not account for anatomical variation, duplicates each patient's unique morphology.

2.10 Addressing the lack of generalizability in anatomically-based display, presentation, and visualization systems.

We have briefly reviewed a number of software systems which are responsible for the 3D display of virtual anatomical entities and the creation of novel visualizations to drive the process of discovery in biomedicine. While the atlas visualization systems produced to date are very effective in approaching the research questions for which they were designed, it is unfortunate that they lack the required extensible architecture which would allow these systems to be utilized in related areas of research. For example, an ISH visualization engine sits upon technology that could be leveraged by another system which manages tumor scans in the same model organism. In fact combining the data might reveal very informative results.

The construction of so many disparate visualization projects has its detrimental effects on the pace of innovation within biomedical informatics. Nevertheless, the field of biomedical visualization has not fully embraced the adoption of generalized visualization toolkits, even though they do exist [36]. It is possible, that those already in use in other fields lack the full scope of features which allow them to be general graphics toolkits, and research applications as well. Therefore, many projects have continued to “reinvent the wheel” with respect to the creation of a number of subsystems that they have in common with one another. Based on the current trend in software engineering of leveraging libraries and toolkits, one would expect that many of the systems described would incorporate modularized components, libraries, and toolkits of earlier systems. Closer examination of these systems reveals that they are all constructed with the following modules in common:

- *Scene Rendering and Management System* – The system responsible for the drawing of screen images based on the parameters of the scene, visual attributes and geometry of the models, and viewpoint of the camera. This framework also manages the creation, deletion, saving, loading, and management of the contents of a scene.
- *User-scene interaction manager* – The system which manages events such as object selections, camera movements, “hover” events, and other user-generated interactions with the scene, so that the system can respond appropriately
- *Content repository* – The storage mechanism by which content is made available to be placed into the scene, as well as serving as a container into which content may be added to or removed from the system’s catalog

The more feature-rich systems which are responsible for producing data-driven visualizations have a need for these additional modules:

- *Data access and attribution system*– The system responsible for acquiring data and managing its attribution to an anatomical concept, whether implemented as an automated process or as a utility requiring human operation
- *Concept knowledgebase* – An ontology system that allows for inference with regard to how the anatomical structures in the scene relate to one another

An appreciation for the identification and description of the above application modules is a critical part of constructing a general biomedical visualization system. The proposed Biolucida system must not only implement these modules but also must provide programmatic access to their functions if the system is to be extended by external software developers. In the next chapter, we will discuss the requirements and challenges that Biolucida faces in offering what seems to be a full feature set capable of meeting the visualization and communication needs of the biomedical research community.

CHAPTER 3

Preliminary Design Phase: Functional Requirements, Architectural Features, and Arising Challenges

3.1 Underutilization of current systems

In the previous chapter, we reviewed many projects which were considered pioneering endeavors in the field of 3D visualization in the biomedical sciences, and highlighted their salient characteristics. While these systems have each contributed value to an array of research endeavors, they remain underutilized for a number of possible reasons. In fact, in 2008 a literature search for references to the visualization system GenePaint [3] revealed only 37 citations versus the 4000 qualified subject area articles that the authors felt should have interest in the system [27]. These findings were not accounted for by a lack of use of online tools and data repositories. The aforementioned study found over 5000 citations for the Gene Expression Omnibus (GEO) [33]. Reasons for the lack of use of a number of prototype visualization systems of the past could include steep learning curves or lack of awareness of the systems. However, we believe that the true causes are more design-oriented: limited usefulness beyond a focused application domain, limited features, and a lack of customizability.

3.2 Application domain specificity related to technology adoption

Biomedical research has used 3D visualization for a variety of purposes, from simply displaying 3D medical images and the implementation of anatomy education instruction systems, to visualizations showing functional data overlaid with structural models. All of the surveyed visualization systems were created to solve specific research problems, where the authors had intimate knowledge of the proper approaches to the problem, of the data sources involved, and of the project's stakeholders. Specialization allows for a group of developers with this knowledge to implement the right solution. Specialization has its advantages, but in Biomedicine, it also has its costs.

From the potential user's perspective, the gene expression visualization systems developed to date offer utility to researchers in a very specific field. For example, consider EMAGE. The system will prove useful only to those investigating the genetically-related developmental phenomena in the mouse. Out of luck are the researchers who may be interested in enhanced features such as animations to communicate any findings, different data sets such as fMRI images *combined* with gene activity in an adult mouse, or the expression data applied to an atlas with much finer detail. With so much invested in the EMAGE system and the enhanced functionality it offers, it is unfortunate that it is limited in scope only by the fact that its architecture is not that of a framework, instead of an application with a limited degree of extensibility.

One would assume that, given modern software engineering practices, many of the aforementioned systems would share technology, each building on advances made before it. However, this is not the case. Even though much of the software projects have common features, each system's technological underpinnings are quite unique, despite the

availability of toolkits and other frameworks [36]. The more recently developed gene expression visualization systems do not use the same high-level rendering technologies, 3D asset management systems, or even the same set of 3D models in cases where the atlas subject is the same species.

While a great deal can be said for allowing a high number of projects to blossom in hopes that one will arise superior to all others, more can be said for avoiding the reinvention of the wheel. For each application produced there is a sizeable labor investment in working with lower-level rendering libraries, such as DirectX [11] and OpenGL [124], to construct content and render the results. Unfortunately, many of the applications which were surveyed have reproduced similar approaches for doing so, but do not share code. While there are some shining examples of general purpose systems such as VTK [140], most toolkits and libraries exhibit a learning curve which is very much like that of gaining familiarity of working with the lower-level graphics rendering engines themselves.

A system developed as a toolkit or framework, provides the capability to “wrap” a core set of modules which offer common general functionality within a system which has implemented more specific features to solve an application-specific problem. For instance, a core atlas system may display the anatomical parts of a mouse, but an ISH application allows additional regions of gene expression to be visualized within that atlas. Designing a visualization system as such would abstract away the details of 3D rendering and scene manipulation, model repository management, and many other lower-level aspects of such a system. Doing so would allow developers who have an intimate familiarity with the unmet functional needs of a potential set of users to focus on higher-level application features, without having to re-implement the common features of the core framework. Frameworks such as Drupal differentiate between “core” development and “contrib” development [32], where the latter builds upon the former. The widespread successful use of many libraries, toolkits, and frameworks is a testimony to how this strategy works well, and should therefore be used more often in biomedical applications, especially in more complex and sophisticated implementations such as 3D visualization systems.

3.3 Functional requirements determination through a survey of other systems

To better understand what is entailed in the construction of a common framework which has the ability to form the foundations of many types of systems, one must first understand the feature set that must be offered, and gain an appreciation for the manner in which it will be used when released to its diverse population of users. We have therefore looked to the large body of literature comprised of prototype system descriptions. Our survey found that the informatics and computer science fields have supported the development of over 200 different visualization systems (see Appendix A) associated with biomedical applications over the last 25 years. An analysis of these systems reveals that they tend to fall into three categories in terms of their use and purpose:

1. **Volume data visualization and fusion.** Many systems have been designed which simply render DICOM image volume datasets. They represent exploratory efforts using novel platforms, simultaneous rendering with complimentary image modalities, or incorporation of novel user-interaction capabilities.

2. **Instruction, task performance, and evaluation.** These systems focus on the educational value of visualization, where a system is designed to “show and tell” or provide an environment for a user to perform a task and the means to subsequently assess whether the task was performed correctly.
3. **Fusible and Attributable data visualization.** A host of systems have been developed for the purpose of creating novel visualizations of domain-specific data which has some relation to the anatomical structures which can be rendered in a 3D scene. From microarray data, to ISH images, and metabolic pathway visualization systems, all are used for specific purposes.

3.3.1 Use-case-based requirements analysis

Assuming that the systems reviewed above were created in response to the needs of their proposed users, they serve as testimonials to a latent collective set of features which a common visualization framework must offer. However, it is best to not only understand the features required of the system, but also who will be using the application and for what purpose. Therefore, it seems a most informative approach is to create several use cases to “flesh out” the manner in which Biolucida might be used. Four typical use-cases were identified which serve to best describe how a generalizable visualization system might be used in the biomedical research community, listed below.

1. **Anatomy education.** A professor of gross anatomy is instructing medical students in the identification of the components of the cardiovascular system, and in their function together. The professor desires to create a visualization of the parts of the human cardiovascular system, display it during class, identifying each part, using custom coloration and other visual cues to show areas filled with oxygenated/deoxygenated blood.
2. **Informed consent.** A surgeon is presented with a case where the patient can choose between three interventions to remove a tumor located in his or her skull. In order to get informed consent for one of the three options, the surgeon desires to produce an animation showing the series of events in each intervention. The animation is to be in a format that the patient can take home with them and use as a discussion tool with his/her family.
3. **Visualization of gene therapy targets.** A researcher is developing an approach to deliver a therapeutic cDNA molecule via an attenuated virus vehicle to combat Hepatitis C. There are potential complications with the treatment if the cDNA interrupts functions in other bodily structures. The virus targets a particular protein receptor known to be expressed in the cells of the liver. The researcher must understand if the protein is expressed anywhere else, as this will inform the choice of delivery and/or targeting method for the cDNA treatment.
4. **Multilevel visualization of patient data from the medical record.** An ER physician is testing out the latest augmented-reality optics, and desires to see a visualization of a recently admitted patient, derived from their medical record. The record may contain demographic data, basic vital signs, 3D and 2D images, and recent prescriptions received. The visualization is designed to give the physician a gestalt appreciation for the patient at first, followed by detail, if desired.

3.3.2 Required features of the visualization environment

When these use cases are analyzed in detail, a set of requirements presents itself that will be discussed specifically in later chapters. The high-level requirements of the proposed visualization system are as follows:

- **Anatomy education**
 - a. *The system must be platform-independent.* Compulsory requirements for either authors or students to use specific hardware to use the system will limit its appeal
 - b. *The system must allow for interactive real-time 3D displays.* It has been proven that real-time interaction with the scene is necessary for users to gain the full benefit of learning in 3D environments [65].
- **Informed consent**
 - a. *The system must provide a means for publishing universally viewable content.* The proposed system must produce output which is readily viewed by a number of programs, and not limited to proprietary products. The latter condition would limit its use for obvious reasons.
- **Visualization of gene therapy targets**
 - a. *The proposed system must allow for the capture of data, yet be independent of the data used.* The system must provide a general set of mechanisms for data importation, yet allow developers to determine which data is to be parsed for visualization, and how it is processed. This removes the burden of providing parsing methods and means of acquisition for innumerable types of data that may be sought after in the biomedical research field.
 - b. *The system must allow for the attribution of data to the objects in the scene.* The connection of the data to the 3D anatomical objects in the scene forms the basis for producing meaningful visualizations. However, such correspondences are neither always one-to-one, nor exact. Therefore the proposed system must provide a method for handling such situations.
 - c. *The system must allow user control over the visualization parameters used to produce a desired representation.* The developer community can be counted on to do only so much. The users should be given a measure of control over selected parameters of the visualization. Some datasets, especially large datasets like gene expression databases, can be represented in a number of different ways to highlight their salient characteristics.
- **Visualization of patient record data**
 - a. *The system must provide a means for transforming the data into new and relevant data types when necessary.* Translational medicine has revealed that communicating knowledge across domains is a critical process that speeds discovery and delivery of innovations [119]. The proposed system should allow for the transformation of data, such as test results into diagnoses.
 - b. *The system must support the display of multiple types of media to the user within the scene.* Interactive visualizations may include the playing of sound (such as a heartbeat recording or a

physician's recorded notes), display of volumetric and model content (from CT, MRI, or PET scans), as well as images and movies (ultrasound, X-Rays, etc.).

This brief list of requirements has not only helped to identify the selection of platform, display technology, and informed other architectural choices, it brings into sharp relief some areas of data visualization research which will be addressed in subsequent chapters. However, a high level description of the resulting putative visualization system reveals a novel application which uses interactive 3D displays to allow the user to build scenes of human anatomy as a substrate for rich visualizations.

3.4 Implementation choices and justification

Informed by the aforementioned analysis of other systems and the subsequent derivation of required features, a set of decisions were made during the design phase of constructing the Biolucida prototype. What follows is a discussion of those decisions and their justification.

3.4.1 Use of an X3D-based VR environment for rendering and scene management

Visualizations prove especially useful when presented within real-time interactive interfaces which facilitate the search and filtering processes [2, 28, 64]. Virtual Reality (VR), a term coined by Jaron Lanier, describes an interactive and immersive environment produced by a computer which allows a user to interact with information in a true three-dimensional manner. VR systems offer a method of implementing both sophisticated visualization techniques and highly interactive control, where the basic goal is to create an interface that offers a more intuitive operation by attempting to be an imitation of a real-life scene.

VR systems provide the user a very intuitive representation using an established notion that high usability arises from designing the conceptual model of a system based on users' previously learned mental models [100]. In this case, the users' mental model is the world around them, or reality itself. In other words, VR systems mimic reality, and thus they are easier to use [40]. Scene description languages which are VR-based, such as X3D [144] and VRML [141], offer flexibility in terms of the interactions they offer with the user, and the type of data which can be displayed. These VR systems can also display a variety of multimedia data, offering a degree of flexibility which suits the growing types of data in the biomedical research information landscape.

While VR systems attempt to encourage more intuitive operation through an imitation of reality, they have also been criticized for their poor ease-of-use compared to their two-dimensional counterparts. While such a criticism is well-founded, the comparison is unfair. Much of the usability of two-dimensional interfaces is due to the greater maturity of the 2D interaction techniques and to a wider user familiarity with 2D interfaces due to a high market penetration of desktop-style user applications and operating systems. The shift from command-line interfaces to the now ubiquitous graphic desktop metaphors originated by Alan Kay allowed more widespread use of computers by groups who lacked the knowledge required to control a command-based textual system. The ease of use of these GUI environments is bolstered by the fact that the control "widgets" with which the user interacts, such as scrollbars and

combo-boxes, have undergone a form of standardization over the last twenty years. This allows users to move from one system to another with the ability to make assumptions that the controls of one system will behave similarly to another's. This is not the case with 3D systems, given their lack of maturity and their limited use by research institutions and large-scale industrial endeavors. Recently, however, 3D technology has achieved a lower barrier to entry and has become more popular among both researchers and consumers. These observations suggest that dismissing 3D interfaces due to low usability might be premature. Rather, we can take two alternative approaches. First, allow sufficient time and development for their interaction metaphors to mature. Second, design interaction metaphors which already match the current expectations of prospective users.

Interactive virtual environments (IVEs) take advantage of a user's naturally developed sense of how to move and navigate through a given world, and how to interact with its objects. Such interactive visualization systems offer complete control over the user's point of view (POV). Since 3D representations may be inspected from arbitrary points of view, it is quite possible that the user can navigate to particularly disorienting viewpoints, or become disoriented in their current POV, making proper navigation challenging. While other earlier 3D systems constrained navigation by requiring the user to specify the camera's position using an alternative method of input such as manually entering the coordinates and view angles, or choosing a location on a two-dimensional proxy, this design leads to a time consuming repeating cycle of specify → evaluate → specify where the user arrives at a correct POV only after substantial trial and error.

It is clear that interactive navigation is the preferred way-finding method for the proposed system, and the aforementioned navigational challenges can be overcome using a few simple controls allowing the user to reposition the camera and establish canonical views. Canonical views, however, have more benefits than simply making navigation easier. Norman, et al [101] investigated the advantages of using interactive navigation with novice learners in the subject of teaching anatomy. The authors suggested using a novel hybridization of a fully navigable virtual anatomical scene and canonical views. Their "canonical views with wiggle" allowed the solidification of posterior, superior, and other views in the student's mind while allowing the "wiggling" of the structure to convey a true sense of its spatial attributes. This hybrid method of navigation led to the highest performance in anatomical knowledge tests compared with free navigation and the control group.

3.4.2 Abstraction of rendering technology from the core system

One possible explanation for the repeated development-from-scratch phenomenon that has been the strategy of so many visualization systems is the persistent rapid evolution of 3D display technologies. Whereas earlier systems were built upon custom rendering software, this approach gave way to using OpenGL and Direct3D as APIs to the new GPU systems. On top of these abstraction layers have evolved many methods of describing models and entire scenes, such as XAML [94] and Collada [23], and many higher-level renderers and game engines like Unity [135], LWJGL [88], and Jogle [66]. Now the development community is witnessing the birth of still other technologies such as the new WebGL canvas specified in HTML5 [81]. When one considers the pace of innovation and the variety of technologies already in existence, it is clear that the module which handles Biolucida's rendering must not

only be flexible, it must be abstracted away from the larger system so that it can be replaced, if necessary, with newer technology.

3.4.3 Model catalog selection

Much of our society's discoveries about the basic processes of life are driven by a curiosity to understand the inner workings of our own body. Therefore, it seems only natural that a majority of the endeavors in biomedical research focus on concepts which, in some way, eventually relate to human function, and therefore, human structure. Today, it can be said that a significant percentage of the research data produced has direct relation to human physiology and structure. From human diseases to the basic functions of ion channels, an overwhelming portion of concepts in biomedicine can be related in some way to human form and function.

Many modern research protocols produce datasets which are best represented in the context of the three dimensional human anatomy. For instance, a researcher who desires to investigate the clinical outcomes of surgery on tumors based on the depth of location within the colon would do well to visualize such data within the context of the colon itself. Therefore, Biolucida's model set was built from a number of human-based sources available to us at the time of the prototype's construction.

| Dataset Name | Description | Creation Method | Models |
|---------------------------|---|------------------------|--------|
| Zygote, Inc. – Human Male | A set of commercially produced surface-mesh models depicting structures throughout the entire body. | CAD Drafting | 694 |
| UW Human Skull Series | A set of models produced from CT scans of a live patient's skull at UW Medical center. | Tiling Serial Sections | 30 |
| UW Human Thorax Series | A set of models produced from serial sections of a cadaver thorax at UW Medical center. | Tiling Serial Sections | 125 |
| UW Human Brain | A set of models produced from serial sections of a cadaver brain at UW Medical center. | Tiling Serial Sections | 30 |

Table 3a. Descriptions of the model sets loaded into the Biolucida prototype system.

3.5 Challenges to be met in implementation

The implementation of a general atlas-based visualization system, given the design decisions discussed above, clearly presents challenges which were identified during the design phase. These informatics challenges composed the main research questions of the Biolucida project, and their resolutions are described in subsequent chapters.

3.5.1 Creating an agile yet robust system

Biolucida is designed to be a general tool for use within the biomedical research community, which may present innumerable application needs. A single core development team of any size would not be well suited to approaching such diverse requirements, and would lack a sufficient familiarity with biomedicine's subdomains to meet such needs. Instead, Biolucida has been developed as a framework or toolkit, to enable those closest to biomedicine's problems to develop appropriate solutions, without the substantial investment in building applications from the ground up. This toolkit approach depends on the involvement of external developers to extend the capabilities of the system through a process called crowdsourcing.

Crowdsourcing, a term originally coined by Jeff Howe in the popular technology periodical *Wired* [57], is also known as "distributed participatory design" [31] in the informatics field, though the latter term is less focused on public participation in the performance of tasks. Crowdsourcing is the offering of a task or problem in an "open call" fashion to the public. It is thought that working in this fashion tends to encourage the participation of those who are most fit, equipped with novel approaches, and having a vested interest in a solution participate in the effort. In the world of software development, crowdsourcing tends to be successful in circumstances where a central development team produces a system with sought-after core capabilities and outside developers produce complementary software modules (plugins) which depend on the core. These modules then extend the system's capabilities in ways where the central team was unaware of such a need, lacks the domain or technical knowledge to develop a solution, or lacks access to a market of users to whom such a module can be delivered. By exposing the APIs of their software systems, a development team can realize the benefits of leveraging the talents of the vast numbers of software developers world-wide. Some shining examples of successful crowdsourcing have come from plugin development for many standalone software applications such as Blender [116], and a large number of frameworks such as Drupal [18]. Biolucida, therefore, implements a plugin architecture which supports such distributed development efforts through crowdsourcing.

The primary challenge behind creating an agile framework such as Biolucida, is developing a system architecture which isolates the components which represent customizable application-specific modules from those that form the common core capabilities of the system. A visualization system is particularly complex where doing so requires significant design effort, and a familiarity with how such a system can be used in a number of subdomains within biomedicine.

3.5.2 Providing ample functionality via library elements and API access

To serve as an effective visualization framework, Biolucida must provide a range of functionality to the developer which provides a great deal of functionality for manipulating elements within the 3D visualization scene, building datasets, and driving the visualization process. Moreover, encouraging adoption of the framework entails constructing many "default" elements, such as pre-made GUI objects and default illustration styles which a developer only references to build entire visualizations.

3.5.3 Providing support knowledgebase support

In reviewing several atlas visualization systems associated with ISH, de Boer et al [27] cited the importance of the use of ontologies in these systems. The implementation of ontologies within an atlas-based visualization application allows for inference with regard to not only how the system's concepts relate to one another, but also how they potentially relate to other concepts from external systems and datasets. The role of such knowledge within a visualization system ranges from intelligent scene generation and manipulation to providing inference capability during the data attribution process. Ontologies have also been implemented in the Hudson Atlas (uses the Puelles ontology), and within EMAGE (both a custom ontology and the Gene Ontology [9]). Given that Biolucida is designed to process and visualize arbitrary datasets, the inclusion of an ontology system is critical to its successful implementation.

3.5.4 Implementing adequate mechanisms for data attribution

The structure of the human body provides an intuitive framework on which to organize heterogeneous data to produce informative visual output. Yet, there are challenges which need to be addressed in order to (1) properly associate data with a human anatomical framework, and (2) create proper and accurate visual metaphors to illustrate the attached heterogeneous data. These challenges represent worthwhile research areas which promise to advance the medical informatics and data visualization fields, fueling discovery in the process. Biolucida is forced to grapple with data attribution to its anatomical models and concepts using a variety of different methods discussed in chapter seven, given that it does not use simple overlay techniques to construct its visualizations.

Biolucida uses an alternative to the common combinatorial overlay approaches that GIS typically uses to create its visualizations. Instead, Biolucida uses a method which will be referred to in this work as *semantic attribution*, shown in figure 2b. Semantic attribution refers to attributing the data to anatomical structures not by spatial colocation, but through identification of identical structures in the data and in the atlas. For example, in the case of the OPT/ISH studies, instead of warping the ISH image directly to attempt to match the atlas models, images are instead interpreted, creating datasets that state the activity of studied genes in the various tissues represented in the atlas. Actually, several ISH projects already perform these tasks, storing the results in an accessible database [3, 142]. Attributing data in this fashion not only circumvents any registration challenges, it offers a number of other benefits.

Semantic attribution confers a number of advantages onto a visualization system in that it allows (1) the use of data that is not explicitly spatial in its raw format, (2) the use of surface models instead of volumetric representations for more efficient interactive scenes, and (3) allows the use of inference for integrating incoming data. Given the advantages of using semantic attribution, and the fact the many systems which use overlay technology end up investing human labor in semantic attribution, Biolucida has been implemented to use this approach in its data visualizations.

3.6 Conclusion of preliminary design analysis.

Analysis of the requirements of the proposed visualization system have presented a wide scope of features which must be implemented if Biolucida is to serve as a generalizable visualization framework for biomedicine. Initial analysis of this feature set have guided a number of design decisions which narrow the scope of questions during the implementation phase. A number of challenges, however, present themselves and must be overcome in order to properly implement the proposed system. The subsequent chapters discuss the implementation of Biolucida, describing the software architecture and processes involved in addressing these challenges.

CHAPTER 4

System Design and Implementation

4.1 Overview

The Biolucida visualization system enables the knowledge-based creation of interactive three-dimensional scenes composed of a set of anatomical structures, relying on a novel set of visualization methods to construct them. These scenes can be used for a variety of purposes, from the depiction of a surgical procedure for informed consent to the animation of blood flow throughout the body for educational purposes. Biolucida is ideal for any application where the main purpose is to convey spatial relationships, as standalone applications offering dynamic display abilities and true 3D representations have been found to be powerful for such goals in the CAD/CAM community [71, 13]. However, Biolucida departs from traditional anatomy visualization applications by allowing the user to create custom content. The application allows the user to specify many parameters which will fit their needs, such as anatomical entities' appearances and discreet camera movements in an animation, in addition to allowing the importation of data which drives the creation of data visualizations.

When used as a basic scene generator, Biolucida's key differentiator is that it uses knowledge-based scene creation methods. Knowledge-based scene creation is a departure from the standard methods of selecting anatomical structures from a database in "line-item" fashion. Since such an iterative process can become laborious, a layer of abstraction allowing the user to specify the contents of the scene in a more intuitive fashion has been implemented using an ontological framework. Commands such as "show me all of the parts of the heart", and "show me all blood vessels continuous with the aorta" are used to construct a scene. These intuitive commands query an anatomical knowledgebase internal to Biolucida, the results of which describe a list of anatomical structures which fit the query criteria. The data required to produce a basic prosaic, or plain face-value, scene include a list of anatomical structures from the knowledgebase, the necessary information specifying the transform and 3D shape of the structure from the model database, and a set of default appearances from the central server.

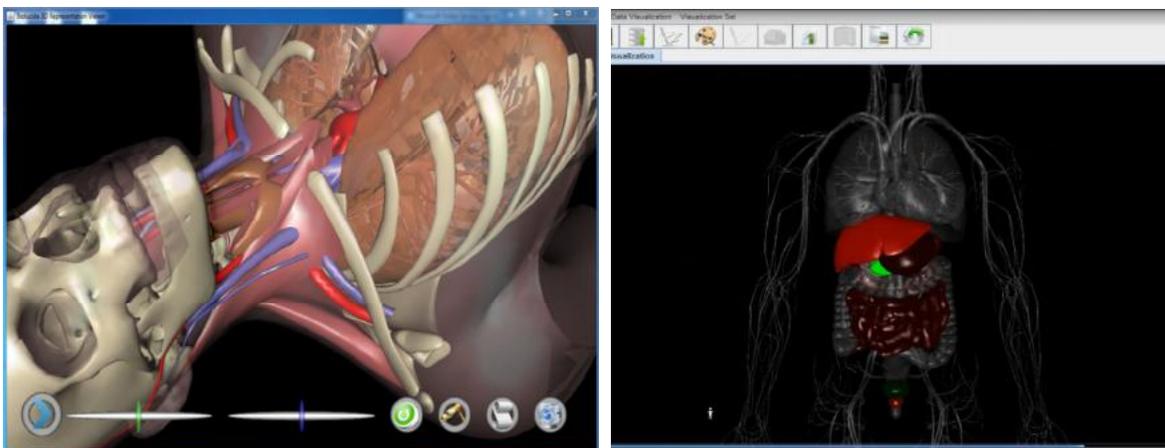


Figure 4a. Example Scenes From Biolucida

As a visualization system, Biolucida leverages similar composition methods used to create 3D scenes via user interaction in order to create rich 3D data visualizations driven by anatomically attributable datasets. Data is acquired, parsed, and brought into an analyzable format where records are attributed to anatomical structures. The resulting models have appearances and behaviors differentially applied according to their attributed data, creating informative and distributable interactive visualizations.

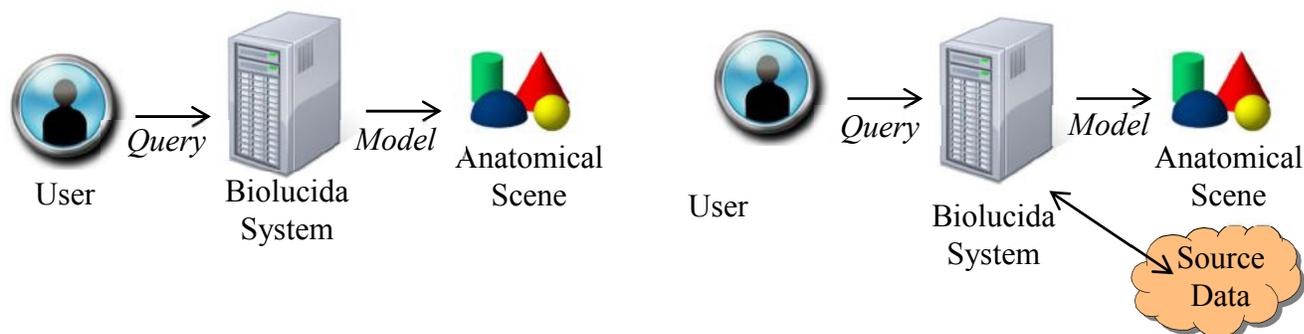


Figure 4b. Workflow in Basic Scene Construction (left) and in Data Visualization Construction (right)

4.2 System description

Biolucida is a follow-on to earlier work, the Dynamic Scene Generator (DSG) [149], in which the idea of constructing scenes from 3D anatomical primitives was first proposed and implemented. The DSG was implemented in LISP and run on an IRIX platform, which constrained its portability and maintainability. Moreover, the previous DSG lacked functionality supporting the aforementioned use-cases, which became apparent after the implementation of the system, such as a higher level of interactivity, more authoring capabilities, and a better user interface. Biolucida's early stages of development addressed these issues.

4.2.1 Basic scene construction components

The Biolucida system is implemented in a traditional three-tier architecture, composed of: 1) The authoring/viewing client, 2) the central Biolucida server, and 3) the Biolucida database. All components cooperate to assemble and render a virtual scene according to an anatomist author's design.

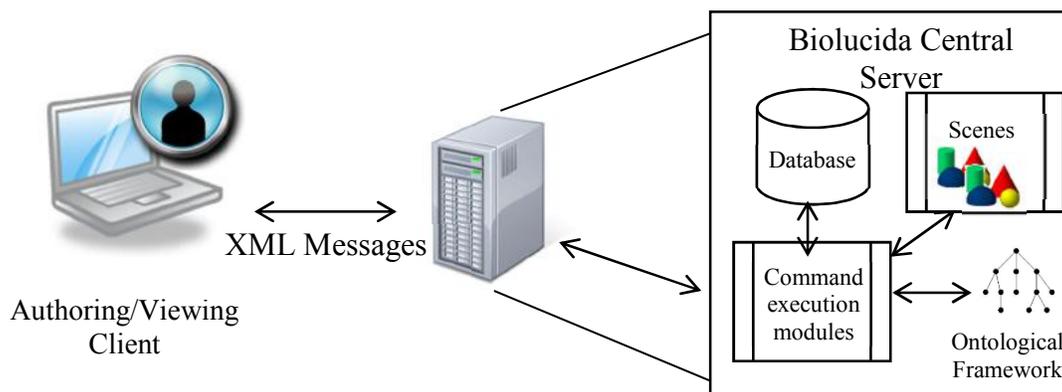


Figure 4c. Components of the Central Server used in scene construction

4.2.2 Authoring/viewing client

The client is a Java application, which is deployed as either an Applet or a Webstart application. The client is responsible for both rendering the scene, and for allowing the author to manipulate the scene via a set of Java Swing control panels. From the user's point of view, the authoring interface is composed of two distinct panels. One panel serves as a canvas on which to render and interact with the scene. The other panel represents a display area for a variety of control modules (fig 4d), offering a level of interactivity and control over the scene that is best managed by two dimensional widgets and text. Selecting from the menus, a user can control a variety of modules which create animations, alter the appearance of objects in the scene, add annotations, and execute many other authoring tasks.

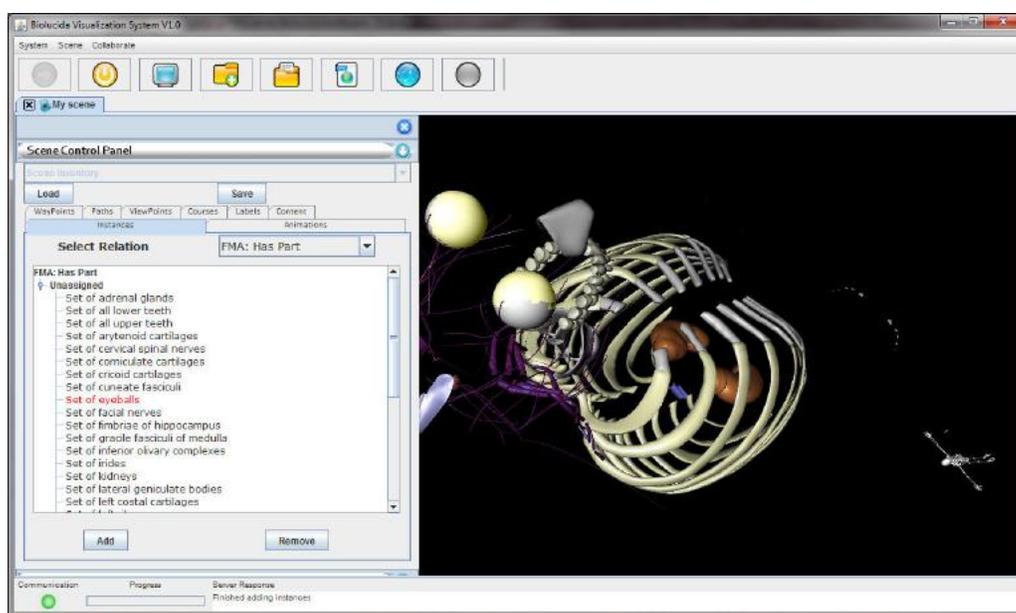


Figure 4d. Authoring/viewing client interface

Both the server and the viewing client are implemented in Java because of its platform-independent nature, and because of the utility of the SAI (Scene Access Interface) of X3D, which allows external applications to easily communicate with the X3D scene. The scene itself is rendered using the open source XJ3D library provided by Yumetech [58]. Xj3D is a Java-based, open source API for creating VRML 97 and X3D applications.

4.2.3 Biolucida central server

The central server, which is implemented in Java, manages communication between the client, the in-memory representations of the working scene, and the Biolucida Database. The central server is responsible for managing the scene's content and state. It manages additions and deletions to the scene, determines appearances of anatomical structures, and processes the abstract scene graph into format-specific content which is sent to the client to be

rendered, as well as acting as a translation apparatus when the client requests to download its content in a particular format.

4.2.4 Biolucida database

The Biolucida database is implemented in PostgreSQL, and includes information such as the concept catalog for the entire system, the ontological framework used for inference, correspondences between concepts and models, as well as information describing user accounts and privileges. With regard to 3D models, the database stores assignments between concepts and specific model file paths. Moreover, the storage of summary spatial information regarding the surface mesh representations of these “primitive” anatomical structures (leaf nodes) allows for rapid spatial calculation, such as transform and “bounding box” data. The current prototype database contains 759 anatomical models, representing structures found within the skull and chest cavity, as well as engineered models from the Zygote corporation [154].

4.3 Basic scene construction workflow

In order to illustrate how the Biolucida application works, it is best to use an example. Consider an anatomist using the system to create a scene containing the basic parts of the skull.

First, a query is issued from the client, requesting the addition of “all structures which are part of the skull”. Concepts with assigned 3D models are inserted directly into the scene, while those which lack such representations are inserted as grouping nodes or empty leaf nodes in appropriate places in the mereological relation which forms the underpinning of the scenegraph (found within the ontological framework). The collection of entities is then transmitted to the client where they are then translated by a content factory, which translates each into format-specific content (i.e. VRML, X3D). The formatted content (in this case VRML97) is then rendered in the scene for the author/user to view.

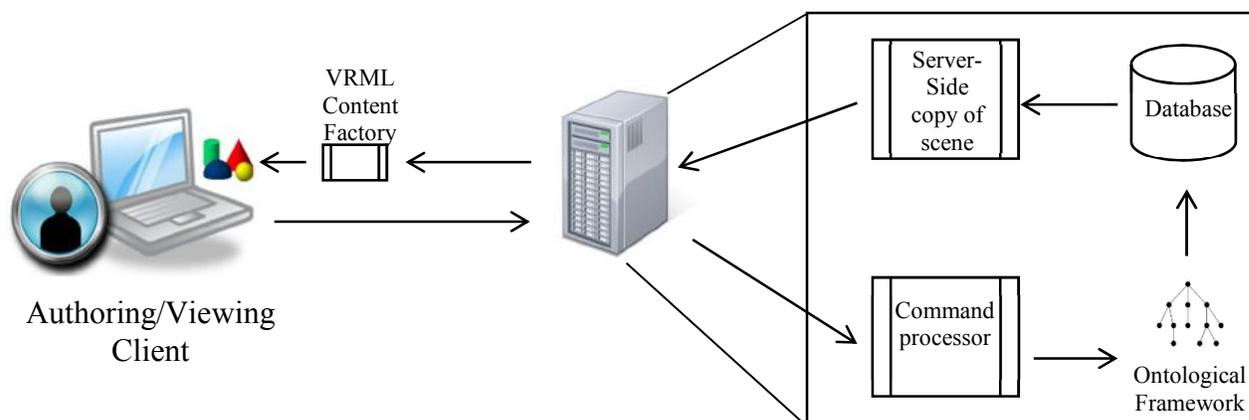


Figure 4e. Workflow of the Client and Central Server Building a Scene

4.3.1 Data acquisition and management systems

When building data-driven visualizations, the central server is responsible for acquiring the source data, which is then parsed and analyzed before becoming attributed to the anatomical concepts which will become the structures populating the scene. There are several components which play a role in this process, depicted in figure 4e. While the fundamental process of building a visualization scene is similar to the one described above, the data, not the user, becomes the primary driver in selecting the set of concepts on which to construct the scene.

The data visualization process consists of the user sending a command to the system which instantiates a data handler, thus beginning the process of data extraction and parsing. The scene is then built by inserting the models which represent the anatomical concepts referenced by the records in the resulting dataset. Each of the following components has a role to play in this process.

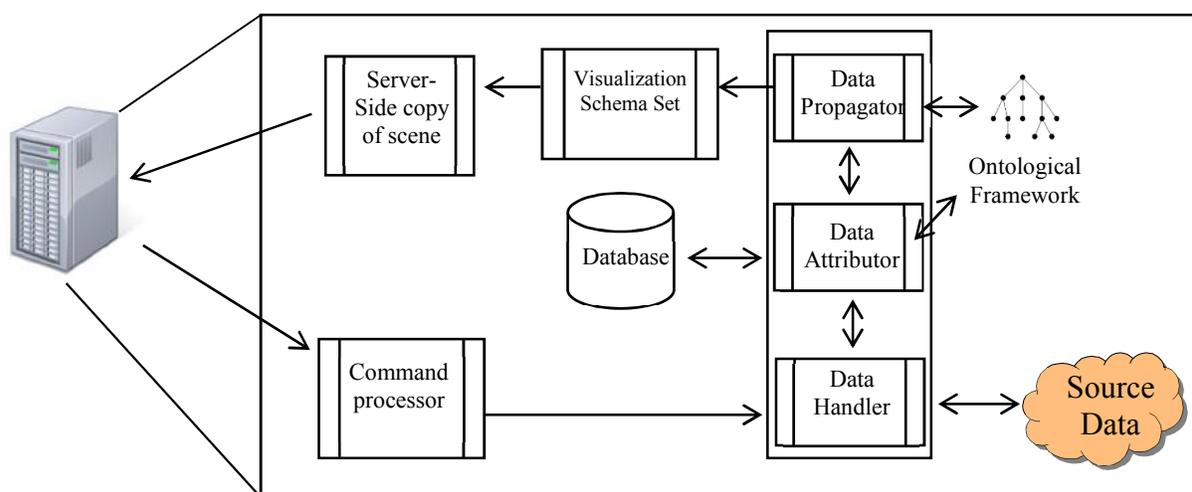


Figure 4f. Components of the Central Server Used in Data Visualization Construction

Data handlers. Data Handlers are primarily responsible for accessing a data source and extracting the relevant content to be brought into the Biolucida server in a tabular format. The Data Handler is also responsible for assigning a standardized field definition to each column. These field definitions contain metadata such as the data type, name, description, and other information. This metadata becomes important to Biolucida while executing various processes during the construction of a visualization.

Data attributors. The data attributor is responsible for assigning a Biolucida-native concept to each record in the processed data set. In order to perform this task, access to the ontological framework is required where inference and customizable program functionality are used to determine the correct concept to assign to each record. Once

such attribution has been performed, the anatomical concept is injected into the growing scene with its attributed data in tow. If the anatomical concept already exists in the scene, the record is attributed to the existing entity.

Data propagators. The data propagator's functions are described in detail in chapter 7. In summary, the Data Propagator creates new records from source records already attributed to entities, moving between nodes along a single Biolucida ontological relation, spawning new data according to rules set by the developer of the propagator.

Visualization Schema Set. When all data have been processed and assigned appropriately, each entity is then passed through a customizable set of visualization schemas, which implement the variables (color, position, animation) of the resulting visualization. Visualization schemas analyze the data attributed to the entities in the scene and, in response to certain conditions set forth by the developer of the schemas, affect changes in the appearance, location, size, selected-3D model, companion content selection, and dynamic behavior of each entity. A visualization schema set is composed of the following schemas.

1. **Appearance schema.** Makes appropriate elections for emissive, diffuse, and specular color, along with alterations in transparency, shininess, and ambient intensity.
2. **Transform schema.** Enacts changes in the scale, position, and rotation
3. **Behavior schema.** Applies the appropriate dynamic behavior to apply to an entity
4. **Contextual schema.** Selects and inserts “companion” 3D models into the scene to provide context
5. **Instance depiction schema.** Selects a more appropriate model in response to the data attributed to an entity, rather than simply using the model associated with the concept alone. This schema is used only when data affects the morphology of the model, as in the case with sequence data suggesting genetic mutations such as birth defects.

4.4 Beyond inference: biolucida's ontological framework

An internally stored ontology allows Biolucida to perform inference in both scene manipulation and data attribution operations. Biolucida uses its ontological framework for a variety of such purposes, such as: (1) intelligent scene creation, (2) proper referencing of entities using semantic level of detail, (3) bulk manipulation of entities such as changing transform attributes or appearances, (4) proper selection of entities from which to compose a visualization, and (5) making proper inferences for assignment of the incoming source data to aid in the process of visualization creation. This wide variety of roles required of Biolucida's ontological framework makes it necessary that it is extensible and flexible in its design. Therefore, the administrator is not only allowed to create an arbitrary number of relations, but to edit them whenever needed to meet the needs of the application domain for which Biolucida is used.

4.4.1 Concepts and the ontological framework

All things, which are either grouping anatomical terms, anatomical terms to which data can be attributed, abstract terms, or terms which are represented using 3D models, are considered *concepts* in the Biolucida system. As within most ontologies, concepts, such as “biventricular part of the heart”, and “skull” exist as unique and describable ideas

or entities within the given domain of application. In the prototype system’s case, the domain of application is human anatomy. Biolucida implements its concepts as entries in the system’s *inc_blc_concepts* table, where each concept is assigned a unique numerical identifier, along with its name and description. Each concept’s numerical identifier is used to link 3D models, appearances, textures, and other relevant information to the original concept, and to create relations between concepts.

Biolucida’s capability to describe relations between concepts lends its users, among other things, the ability to quickly build scenes, and to make inferences about attributing data to its concepts beyond what is described in the incoming source data. Biolucida implements its ontological relationships using a very simple approach: via an entity-attribute-value table. The *inc_blc_relations* table serves as catalog of the relations currently known to the system where the relations are assigned a name, description, and attributes such as transitivity. Each relation is given a unique ID, and therefore the catalog can store an arbitrary number of relations. The *inc_blc_concept_relations* table stores the relations between concepts, using a “from concept id”, “relation id”, “to concept id” entry.

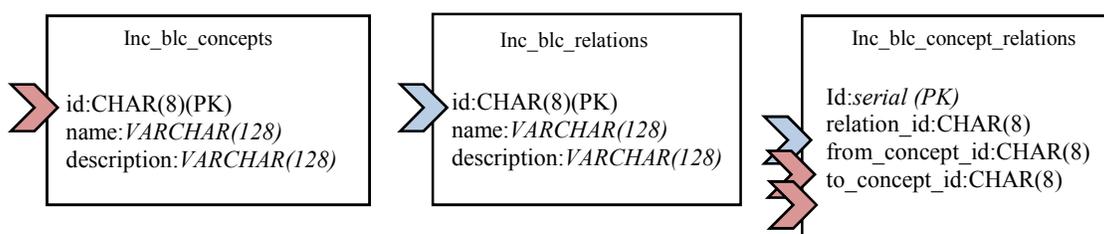


Figure 4g. Selection of the Biolucida Database Relating to the Ontological Framework

For example, a typical mereological entry may be “left atrium”-*-“part-of”-*-“heart”, though numerical IDs would be used. While such an approach may be inefficient for storing large numbers of relations, Biolucida is not designed to manage and use the entirety of a reference ontology, such as the FMA [117], but rather uses a smaller view. The ontological framework must exist both on the server and on the client, both because of the speed required for ontological query execution while displaying an interactive 3D scene, and because the server is responsible for managing the data-centric operations in the production of Biolucida’s visualizations.

Semantic Web technologies are a viable alternative to our chosen database implementation of concept-relational knowledge representation. Syntaxes such as RDF are expressive, as well as readily parsed and queried by a number of stable libraries which could be integrated into Biolucida’s core. However, such technologies were not as mature, when Biolucida’s prototype server module was reaching completion of its design and development phase, as they are now. The storage method used in the current implementation maps to the concept of “triples” within RDF quite well, and therefore the database can be exported to such a format. However, the database backing store offers a level of security and stability that a file-based method of storage may not offer.

4.4.2 Server-side ontology management

Ontological relations managed by the server-side are loaded into memory as graph data structures. On startup, the concept relations table is processed in its entirety and each graph built containing all appropriate concept nodes and relational edges. This initialization need only happen upon startup of the server, as each graph is serialized into XML and stored for subsequent download by the clients. If a relation is altered by the administrator at any point, the server must be restarted for those changes to take effect.

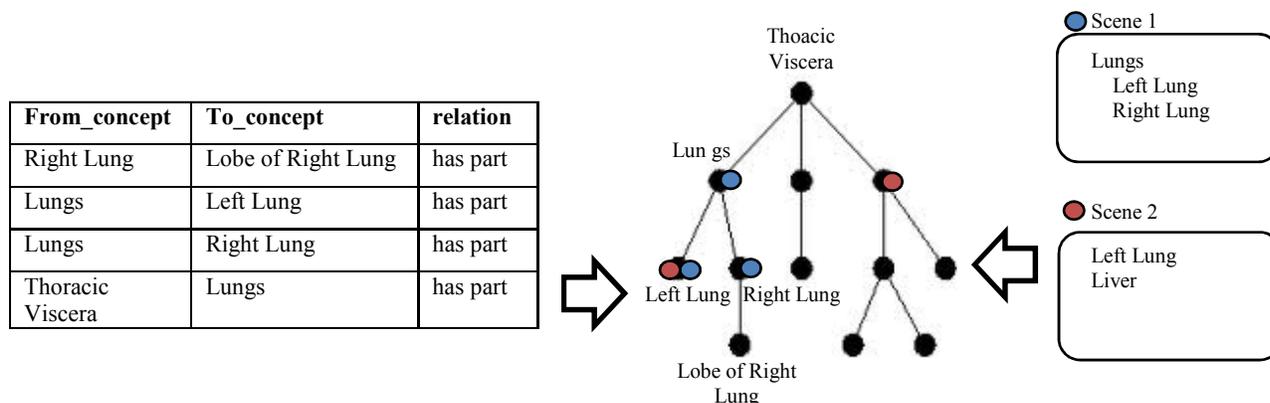


Figure 4h. Depiction of the Server-Side Ontological Framework and Concept Instance Management

Each concept node is designed as a container which will later hold scene-attributable instances, when the server is given a command to insert such an instance into a given scene. So, the server is responsible for holding a graph containing the instances of all scenes, while the client copy of the graph structures is solely responsible for instances contained in its own scenes.

4.4.3 Client-side ontology management

Like the server, the Biolucida Client keeps its ontological framework as a set of in-memory graphs for speedy access during navigation and manipulation of its scenes. The initial material from which the in-memory graphs are built is downloaded from the server at sign-on in the form of an XML stream which is parsed appropriately. Given that Biolucida must support an unknown number of and types of relations, the default client supplied by the server contains an ontology editing module used to create and edit the contents of an arbitrary number of relations.

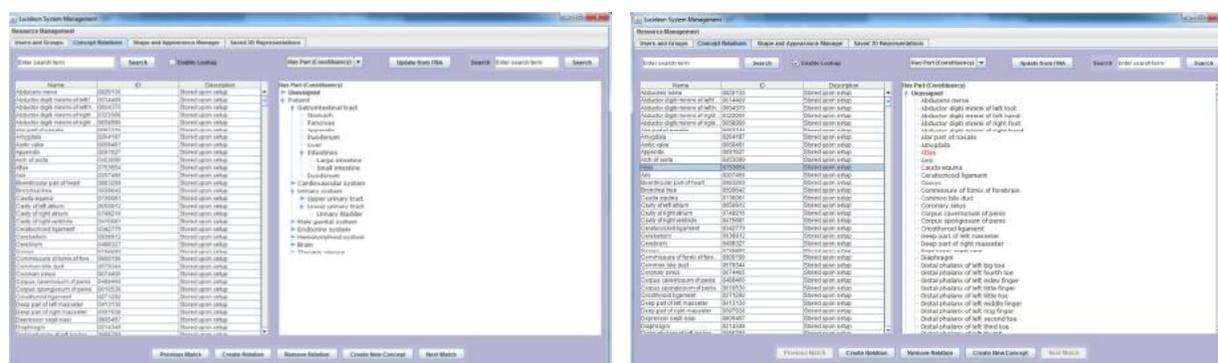


Figure 4i. The ontology editing utility provided within Biolucida.

4.4.4. Rationale of custom ontology creation.

As mentioned earlier, Biolucida is a generalizable biomedical visualization system which is designed to be used within many domains of application from surgical simulation research to molecular biology studies of human tissue samples. Given the unpredictability of the needs of such a wide array of users, and of the knowledge which they will endeavor to represent, it seems only fitting that Biolucida be equipped with the ability to create, edit, and copy ontological data. There are significant arguments as to why the “single truth” of a reference ontology should be used as the framework on which to organize a visualization’s contents and the attributed data. However, this view is incompatible with how the system would be used in the following situations.

1. **Novel concepts.** Consider a researcher who is identifying areas of the colon which exhibit different risk levels for surgical removal of tumors. These regions represent discreet concepts to which source data can be attributed. Yet, as a part of a legitimate research endeavor, such concepts would not be part of the established knowledgebase and therefore would be absent from any reference ontology. Moreover, such novel concepts may represent ideas which may not integrate properly into established relations occurring within a reference ontology.
2. **Novel relations.** While a reference ontology such as the FMA accounts for standard relationships such as “continuous with”, “is a”, and “regional part-of”, one cannot amass all possible desired relations to be included in a research tool no matter how exhaustive the reference ontology may be. The most direct example might be what we will call “phenomena-based” relations. Consider a visualization illustrating a model of metastasis for a type of cancer. While connectivity relations for the blood vessels and lymph ducts may allow some inference to suggest where cancer cells may travel, other relations may prove interesting to include in the analysis, such as “provides invasion path to” in order to suggest the potential for cancer cells to arrest fluid-born travel and propagate through the lumen of one structure in order to gain access to another.
3. **Defunct or disputed data.** While reference ontologies are meant to represent the current state of knowledge in a given domain, there are no guarantees that such knowledge is either current or unanimously

agreed upon. Users may wish to update their own ontologies to reflect evolutions in knowledge and change requests that have been approved, rather than wait for the appropriately responsible development team to update a widely-distributed reference ontology. Though it is quite contestable to claim that separate groups should be able to maintain disparate ontologies to reflect disagreements in the representations, we feel that freedom should be given to the authors to answer this question.

4. **Hypothesis testing/experimentation.** Biolucida’s data propagation and transformation capabilities may, in addition to aiding in the construction of informative visualizations, serve as a framework with which to test hypotheses. Consider a dataset which gives initial conditions of a puncture wound received at a given point on the heart, and the goal of the user is to create a time-series visualization of the injury propagation resulting from the initial wound. It may be necessary to maintain several version of an ontology to test hypotheses about how injury conditions propagate, comparing the different versions to assess which is able to best explain observed phenomena.
5. **Modeling abnormal conditions.** While the prototypical human may be the focus of most research endeavors which may utilize Biolucida, it is safe to say that modeling aberrant conditions such as disease, developmental abnormalities, and genetic mutations is of significant interest to a subset of potential users. These situations may present conditions where a “normal” reference ontology must be altered in some way to appropriately represent the abnormalities of the condition of interest.

This is not to say that Biolucida is incapable of standardization using reference ontologies, for there are mechanisms implemented within the system that allow for view extraction from such ontologies as the Foundational Model of Anatomy. The FMA is an ontology, represented in Protégé, that consists of over 70,000 concepts and 110,000 anatomical terms. Its 168 relationship types yield over 1.5 million relations between its concepts. The FMA is made available to other programs by means of a query service, where requests are formulated in the SPARQL query language and the service provides RDF responses. The Biolucida Server is designed to extract views based on the set of seed concepts already resident in the concepts catalog, thus creating a view of the reference ontology used in construction of scenes, data propagation, and in manipulating scene contents in real time.

4.5 Plugin architecture

In order to offer critical domain-specific functionality, a generalizable system like Biolucida must allow its feature set to be extended and enhanced by developers who may know more about a given domain’s requirements than the core authors. After all, the original development team cannot “solve the world” by anticipating all possible needs. Therefore, Biolucida has been developed with a plugin architecture to support the incorporation of new reference ontologies, Data Handlers, Visualization Schemas, Data Attributors, and Representations.

To review, the table below describes the primary responsibilities of each class of plugin, along with the functionality which the developer is able to customize. Each plugin class is developed using a library which is a small subset of Biolucida’s codebase, primarily composed of the interfaces and abstract classes which the new plugin objects will implement and extend, respectively.

| Plugin Class Type | Customizable Functionality |
|--------------------------------|---|
| Reference Ontology Handler | Interfaces with a reference ontology to extract a functional view based on the concepts contained within Biolucida's concept catalog. The Reference Ontology Handler is responsible for querying its designated data source and inserting novel relations into Biolucida's ontological framework. |
| Data Handler | Data Handlers, described above, can be modified to access data sources which are remote, file-based, database-driven, or even accessed through customized APIs. The developer has full control over the execution of all aspects of the data parsing process. It is expected that the data handler yield a dataset in Biolucida's native format for subsequent presentation to the other classes in the data processing pipeline. |
| Representation | A representation is a user interface definition, along with functional capabilities afforded by the Representation's event processing logic. For example, Biolucida has basic 3D representations, which offer simple navigation within a 3D visualization, and "Constructive" 3D Representations, which allow an author to alter every aspect of the scene to their liking, using input events such as keystrokes, menu selections, and interactions with objects in the scene. |
| Visualization Schema/Schemaset | Visualization Schema Sets are described above. When made available to the system, a user can choose to include these schemas in building their own visualizations. |
| Data Attributor | Custom coding to the data attributor can create alignment between unknown terms which may occur in a dataset, and native terms which are resident in the Biolucida system. |

All externally developed plugins are exported into jars which bear the fully qualified name of the main class which implements the appropriate plugin interface. The jars, when copied into the designated subdirectories of Biolucida's installed location, are loaded upon server initialization and all constructors are called via reflection, so that instances of each type of plugin are loaded into memory and made available to the system.

CHAPTER 5

General Client Systems, Processes, and Representations

5.1 General concepts of client-side extensibility

As we discussed earlier, biomedical research is a broad field with numerous potential domains of application for a visualization system such as Biolucida. Therefore, Biolucida is designed with extensibility in mind, so that it can be applied to many areas of research using straight-forward extensions to its abstract user interfaces, event model, and representation frameworks. In doing so, a developer is able to create innumerable possible client applications with minimal investment. It is understood that these extensions need not be made by those who are familiar with the core application code, but can be constructed by external developers who have access to Biolucida's abstract frameworks and well-defined interfaces. This chapter describes the core architecture of Biolucida's client systems, with a focus on extensibility. There is a great deal of abstraction used throughout the client architecture and what follows is a description of these design conventions. First, it is proper to explain the relationship between clients, data, and Representations.

The Biolucida system is composed of both client and server components, communicating via XML-based commands and responses. The primary task of the client is to allow the user to create and interact with a Representation – an interactive visage of a dataset (or blank dataset) which may consist of an interactive 3D world (the focus of this project), contained within the client's Java interface containers. The contents of a Representation are left up to the developer. Representations function as small semi-independent modules, defining much of their own UI elements, event processing and user interaction logic, and visualization conventions. Clients can manage a number of Representations, allowing the user different views of the datasets which they describe.

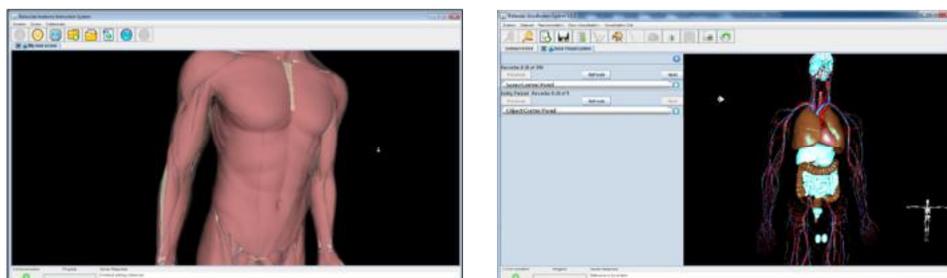


Figure 5a. The ASA client (left) and the Datagen client (right) are two of the many derivative clients that can be constructed by extending the Biolucida visualization framework.

5.1.1 Extensible client applications

A novel Biolucida Client can be developed by extending an abstract client class which is defined in Biolucida's core package. This abstract class offers a wide array of methods used to perform basic tasks such as communicating with the server, managing representations, etc. Additionally, clients may use a number of standard UI elements available

within the core library to construct toolbars, control widgets, and entire panels of data displayed for the user. Clients may also offer custom functionality implemented by the developer to serve the needs of the domain for which the client is being developed.

5.1.2 Representations defined

Biolucida is designed with the idea that a dataset can be represented in a variety of ways. It can be said that a “Representation” is a visual abstraction of its underlying data, rendered in its own unique way. Earlier in this thesis, we covered two-dimensional visualizations such as tables, graphs, and a vast collection of pictorial representations which help the viewer appreciate various salient attributes of the data. Though Biolucida was developed with three-dimensional visualization in mind, it is adequately capable of building such two-dimensional depictions, simply because a “Representation” is a library composed of a well-defined set of classes which, when implemented by the developer, enable the presentation of, exploration of, and interaction with graphic abstractions of the data. Developers are free to draw (no pun intended) resources from graphing libraries, GIS mapping packages, and charting services, rather than use Biolucida’s included 3D visualization environments. Developers may still include rules for building these graphic abstractions, and an event processing pipeline to provide interactive functionality to the user. However, the remainder of this work will focus on the use of Biolucida’s 3D Representation environments.

5.1.3 Simplified plugin architecture

Biolucida implements its extensible framework via a simplified system of using Java *reflection* to invoke instances of Representations, DataHandlers, and several other custom designed modules on which much of its functionality depends. The deployment of a plugin consists of a jar export, named using the fully qualified object class name which is to be instantiated. This jar is placed into the correct server directory, denoting which type of Biolucida resource it represents. In the case of Representations, the Biolucida server loads all appropriate Representation definitions by scanning the its *representations* directory, and using reflection to instantiate the description classes of each Representation, thus building a catalog of available representations for the users to select from. Each Representation archive will contain a *description* class which helps to define its capabilities both on the client and server side.

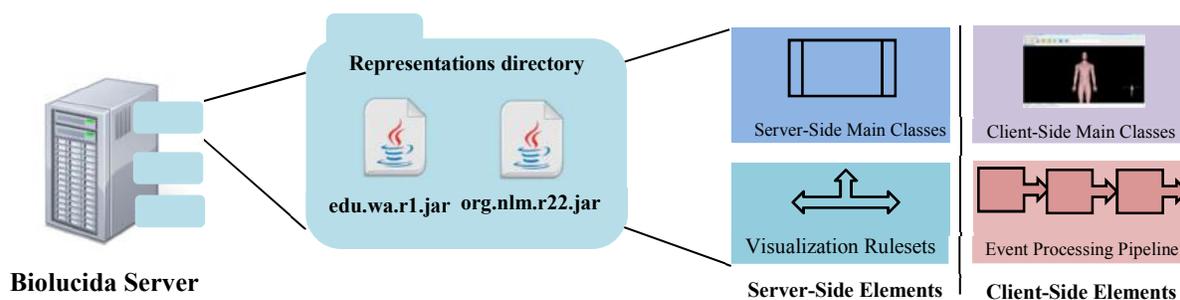


Figure 5b. Depiction of the deployment strategy and contents of Representation plugins

As you can see in the above figure, a Representation is composed of both client and server implementations. The server-side implementation usually contains classes which are necessary to maintain the state and persistence of the Representation's contents. All visualization construction routines which happen in response to the import of a dataset are implemented only on the server-side. Since the representation is actively viewed only on the client-side, UI elements are defined only for the client-side implementation.

Representations are implemented through the extension of abstract Representation classes which define many of the rudimentary capabilities to deliver an interactive 3D scene. These pre-constructed frameworks are especially helpful in 3D representations, due to the complexity of creating and managing a scenegraph, all of which is handled by the Biolucida core client classes (and the 3D graphics module) and incorporated Xj3D libraries. The core system is capable of not only displaying the scene's entities, but also provides a mean of handling events issued to and from the scene itself. The features of each Representation offered to a user differ by means of the event processing pipeline logic and the required user interfaces which are implemented by the developer.

5.2 Client architecture in detail

The framework of the Biolucida client defines the following components which determine the appearance, behavior, and functionality of the client application:

- **User interface.** A Biolucida Client must define a set of user interface classes which will be used as the main display window, the function menu, and the function options panels.
- **Event logic.** The client must define two event processing pipelines which determine behavior and functionality when the user interacts with the scene, or issues events by using the keyboard or other non-navigational input devices. Each client, therefore, has the ability to handle events in a different manner, calling core standard Biolucida functions as a result.
- **Initialization operations.** It is assumed that each client must execute a set of preliminary functions in order to achieve an initial "ready" state. Such functions are defined in the client's design and are executed upon initialization.

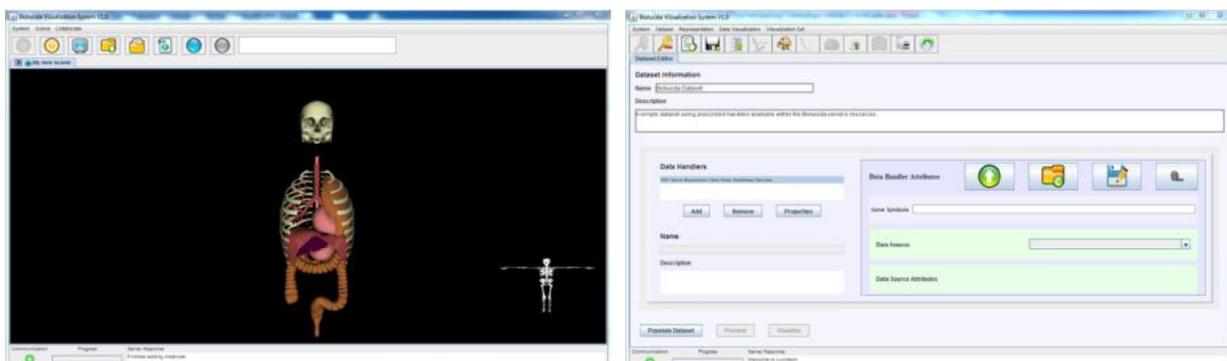


Figure 5c. Two different clients, the Anatomy Scene Instruction client and the Datagen client, developed as extensions to the Biolucida core client system.

5.2.1 User interface

The standard client's implementation framework defines a set of configurable user interface systems which allow a developer to create core pieces of functionality and program behavior which can differ significantly from client to client. These interface modules are composed of four main classes which are Java Swing containers: (1) the main UI toolbar, (2) the representation panel, (3) the functional options panel, and (4) the ancillary options panel. Developers have the option of selecting predefined command objects and UI elements from the Biolucida library, or may define their own.

Main UI toolbar and menu. The UI toolbar and JFrame menu define high-level operations such as signing in/out, saving and loading datasets, saving/loading representations, and many other basic client functions. Implementing the ability to customize these UI elements was prompted by the fact that not all client applications require the same set of these basic features. For instance, clients which focus exclusively on building and interacting with anatomical scenes, such as surgery simulators, do not require the creation of backing datasets, and therefore do not present such functions to their users.

Representation panel. The representation viewing panel is responsible for rendering the scene, and for handling all user interaction with the entities contained therein. Again, the developer of a client application may use the standard client panels provided within the Biolucida library, or may opt to include a different rendering UI object. However, it is required that such an object is capable of being hosted within a Java UI container. More detail on the rendering component's architecture which is used for 3D Representations is given later in the chapter.

Functional options panel. The abstract UI framework which is extended to build a Biolucida Representation provides a "popup" UI panel which is displayed in response to a right-click event issued by the user from within the scene. The panel is designed to contain a menu of options such as navigation mode selection, entity insertion/removal, and other functions which are specific to either view control or the scene's contents.

Ancillary options panel. For operations which require a user interface that is more expressive than the smaller functional options panel, and for operations which involve non-scene-related operations such as viewing source data, the client UI framework also includes an ancillary options panel. Functions such as appearance change controls, scene inventory tables, and animation construction elements are placed into this container. The developer supplies the contents of this panel in order to implement a Representation's custom functionality.

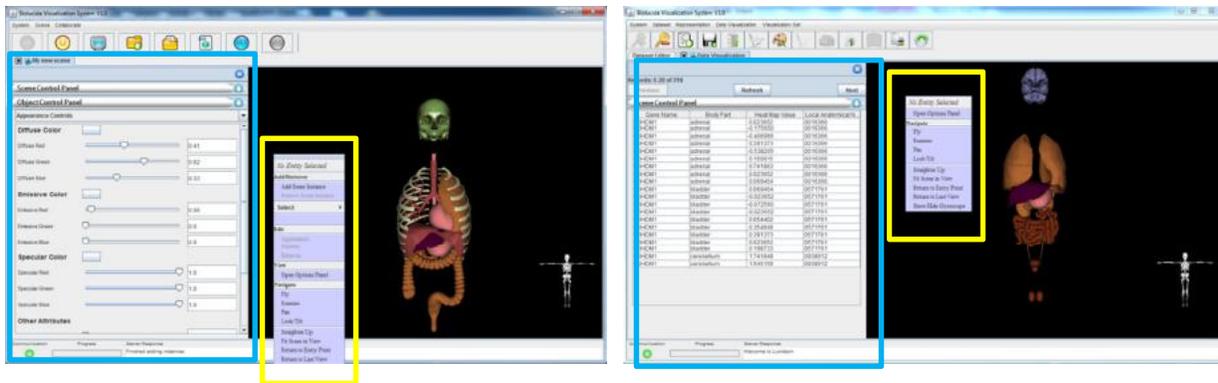


Figure 5d. Two sample representations and their implemented UI panels. The representation panels are delineated in yellow, while the ancillary options panels are outlined in blue. The Anatomy Scene Authoring (ASA) Representation (left) shows clear differences in functionality when compared with the Datagen Representation (right). The ancillary options panel is highlighted in blue, while the functional options panel is highlighted in yellow.

5.2.2 Event processing pipeline

The customizable UI elements designed for user interaction do not offer any functionality without the appropriate event processing pipeline which is responsible for taking action on user commands, and reacting to events which arise from within the representation itself. There are two distinct event processing systems which are implemented in Biolucida's 3D representations: the controller event pipeline and the scene event pipeline.

The controller event pipeline is designed to handle keystroke and mouse events issued by the user. This system processes these events appropriately into commands acted on by the Biolucida client. There are several input controllers which are implemented in a Representation. Although the Biolucida system offers generic implementations of each, the developer has the option of either extending and customizing them, or developing their own from scratch, provided that the appropriate interfaces are implemented.

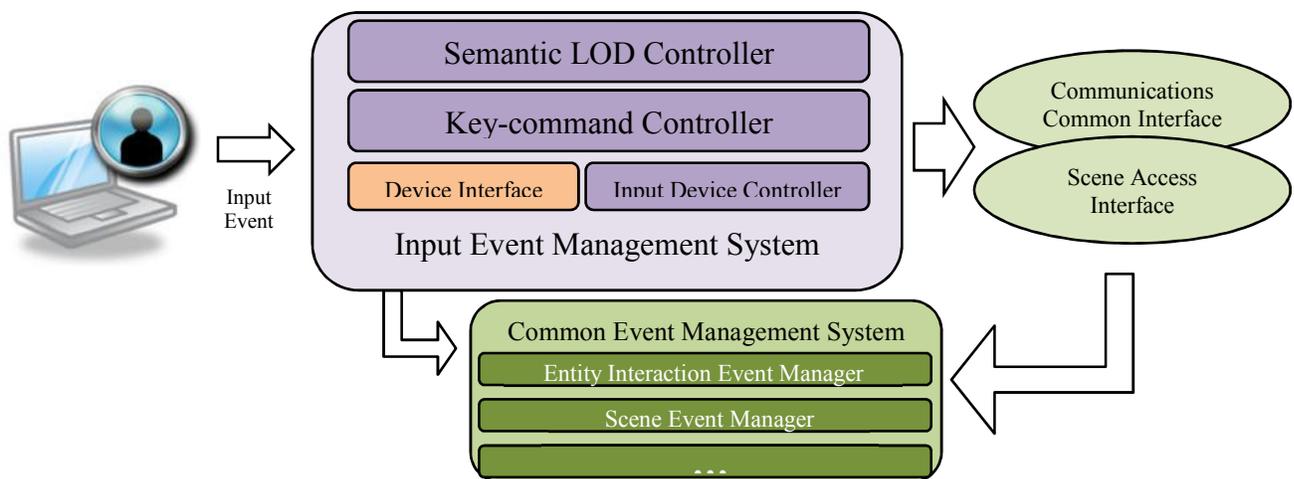


Figure 5e. Schematic showing the control event processing pipeline

5.2.2.1 Supplied default controllers.

Key command controller. The key command controller registers itself as a key stroke listener with the main Representation panel, so that keystroke commands can be used to move the camera, grab and move entities, insert and guide a cutplane, and many other in-scene operations.

Input device controller. The system must be able to react to a variety of events which arise from the pointing device operated by a user. Biolucida was designed not only to support a variety of display systems and their appropriate user input devices (such as mice, 3D mice, haptic instruments, etc.), but also to be resistant to inevitable changes in technology. A “Device Interface” abstraction layer is used receive and translate events from a pointing device to the actual Input Device Controller. The Input Device Controller then issues events to the scene, communications system, or to other modules in the form of “Pointer Clicked”, “Pointer Right-Clicked”, “Pointer Entered”, “Pointer Exited”, and “Pointer Moved”.

Though the aforementioned input device events offer such a general description of what has occurred when the user has , Biolucida has implemented a set of Event Managers which are responsible for broadcasting more specific events to objects which have been registered as listeners. For example, the Entity Interaction Event Manager uses the combination of events from the scene of a 3D Representation (“Entity Moused-Over”) and the events from the Input Device Controller (“Pointer Right-Clicked”), to issue an “Entity Right-Clicked” event to the listener pool. The client system then takes appropriate action. In this case, an entity-specific Representation Options panel is displayed. These more specific event managers are grouped into a Common Event Management System, which make these events easily accessible to other classes, simply by registering themselves as listeners with the appropriate manager. The following Event Managers offer any class within the Biolucida Client the ability to take action on the following events:

| Event Manager | Description |
|--------------------------------------|--|
| Animation Manipulation Event Manager | Manages events where the contents of animations are added to, pruned, or otherwise updated, keeping registered listeners informed of the new information |
| Camera Viewpoint Event Manager | Manages events where the camera’s position and/or orientation has changed, alerting registered listeners with the updated transform information. |
| Dataset Event Manager | Manages events where the dataset associated with the user’s current session is created, removed, or updated, informing listeners of the change. |
| Entity Change Event Manager | Manages events where an entity is added, deleted, moved, rotated, or its appearance changed, keeping listeners informed of these changes. |
| Entity Interaction Event Manager | Manages events where an entity is selected, moused-over, double-clicked, and other device-based interactions that originate from the user. |
| Ghost Interaction Event Manager | Manages events where a ghost is added, deleted, moved, or rotated which is usually associated with changing the attributes of an action in an animation. |

| | |
|--|--|
| New Content Event Manager | Manages events where new content has been added to the scene, which includes contextual content as well as entities. |
| Representation Event Manager | Manages events where a new representation has been created, destroyed, saved, loaded, etc.. |
| Representation Keystroke Event Manager | Manages events where the user has pressed a key to invoke a particular action, announcing the execution of the action to its listeners. |
| Representation Mouse Event Manager | Manages events where the pointing device has been “clicked”, moved, “right-clicked”, “double clicked”, etc, generally within the display of the currently active representation. |
| Scene Event Manager | Manages events where the camera has collided with, approached within a critical distance of, or gained or lost visibility of a given entity in the scene, |
| Temporary Object Event Manager | Manages events where a “temporary” object, such as a label, axes, or waypoint/viewpoint is added, deleted, moved, or rotated. |

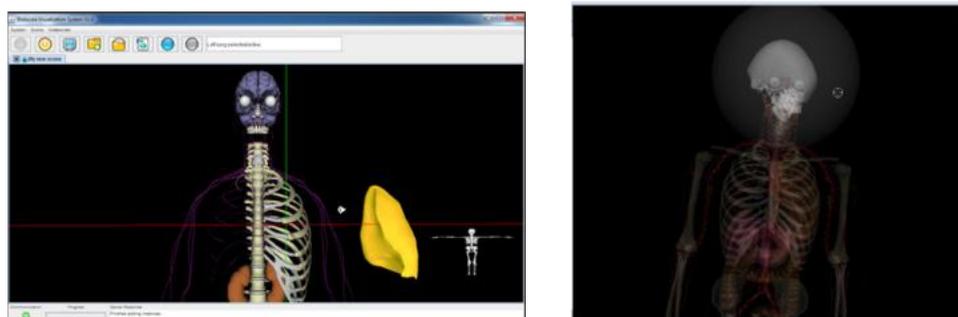


Figure 5f. The grab and rotate functions of the ASA’s controllers in action.

Semantic LOD controller. The Semantic LOD Controller was designed separately from the other controllers because of the amount and the complexity of the logic it contains, which manages the navigation between the semantic levels of detail (SLOD) within the current scene. SLOD is parallel to the established concept of graphic level of detail, often used in computer graphics and in GIS technologies [97]. Graphic level of detail is used to provide varying levels of fidelity to graphic representations of an object which are dynamically toggled based on the appropriateness of each rendition to certain conditions [21], such as the distance of the camera from the object. SLOD, however, represents dynamic *semantic* groupings of models in a scene, allowing the system to change the granularity with which the user interacts with and views a set of concepts.

Described in the next chapter, the display and navigation of semantic level of detail is a required capability of any extendable visualization system to be used within Biomedical research. It describes a conceptual hierarchy of entities, often modeled by a mereological relation, which the user can focus upon for manipulation or viewing. The storage and management of ontological relations was discussed in the previous chapter, and SLOD depends upon a single chosen relation among those modeled within Biolucida, acting as a framework where the active selected and displayed concept in the Representation is changeable by moving between concepts along that relation. For

example, in a scene composed of the thoracic viscera, the user may have selected a bronchopulmonary segment of the left lung, but can easily move from this concept to a lobe of the left lung, the left lung as a whole, or the lungs as a pair.

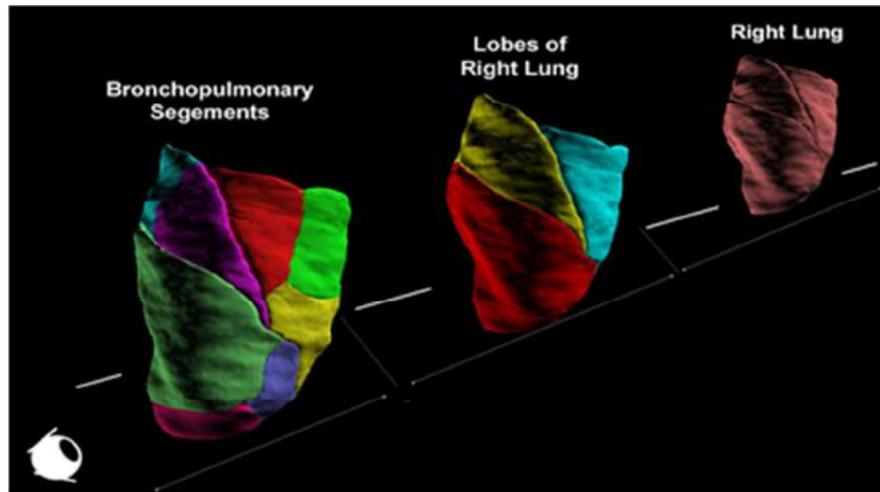


Figure 5g. Pictorial representation of Semantic Level of Detail

The Semantic LOD Controller currently uses keys to change the current SLOD, with a timed highlighting event to show the user that the current SLOD has changed, and to provide a visual queue as to what structures the current SLOD might include. The underlying details of the management of SLOD navigation and proper display of concepts at different SLOD levels is discussed later in the text.

5.2.3 Client communication systems

As previously mentioned, the Biolucida System operates using a *client command/server response* methodology. All classes within the Biolucida Client have static access to a set of Command Sources, which are responsible for sending XML commands to the server, where they are processed by Command Processors, and the appropriate XML responses sent to and parsed at the client. This set of commands forms a basic API where the Biolucida Server could possibly be used as a service by more clients than those designed under the constraints of the Biolucida Client framework. The available commands and their descriptions is available in Appendix B.

5.3 Workflow customizations: initialization operations

Initialization operations can differ from client to client, where a series of predefined command objects are invoked after the client has been properly authorized. The developer of a given client may automate certain initial operations such as the creation of a given scene or representation, the adding of certain entities to the representation, or the application of behaviors to those entities. In the case of the most basic scene construction client, the initialization operations include the creation of a blank dataset (since no representation can be without one), followed by requesting a 3D representation of that dataset – a world containing no entities. The author is then free to begin

constructing a custom scene. In the case of the Datagen client, the user is prompted to specify a Data Handler, and a number of attributes relating to the incoming dataset and the resulting Representation before a scene can be viewed.

5.4 Flexibility and longevity through abstraction: The 3D framework architecture

As previously discussed, all technologies will continue to evolve due to the nature of innovation. Systems which are designed to be resistant to obsolescence must be designed to incorporate these changes easily. One of the best methods of implementing this philosophy in software is through modularization and abstraction. While abstraction is required in a number of functional areas within the client to ensure flexible modular design, one of the modules which promises to benefit most from such abstraction is the system responsible for rendering and managing interactions with the scene. In chapter 3, we noted that graphics technology was rapidly evolving and that Biolucida must remain unbiased toward its implemented graphics-systems to ensure its own longevity. In an effort to separate its 3D systems from other elements of the client core, Biolucida uses the Model-View-Controller (MVC) development paradigm to provide convenient and well-defined ways of accessing and controlling the scene without tying itself too closely to the renderer. Such an architecture allows a fairly simple substitution of one 3D technology (such as Xj3D with OpenGL), for another (such as LWJGL).

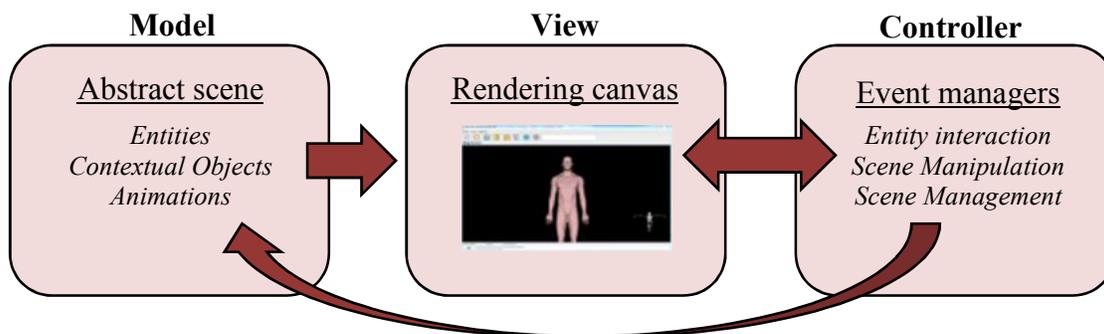


Figure 5h. Graphical depiction of Biolucida's MVC implementation of the rendering module

5.4.1 Detailed description of the Web3D-based 3D representation framework

The default 3D Representation framework offered for implementing a variety of types of interactive visualizations is based on Web3D standards, or more specifically, on X3D. X3D is an XML-based ISO ratified scene description language which provides file syntax, interface definitions, and viewer requirements for the definition and display of interactive graphics. X3D is an open standard so that it can support the widest array of domains and their applications.

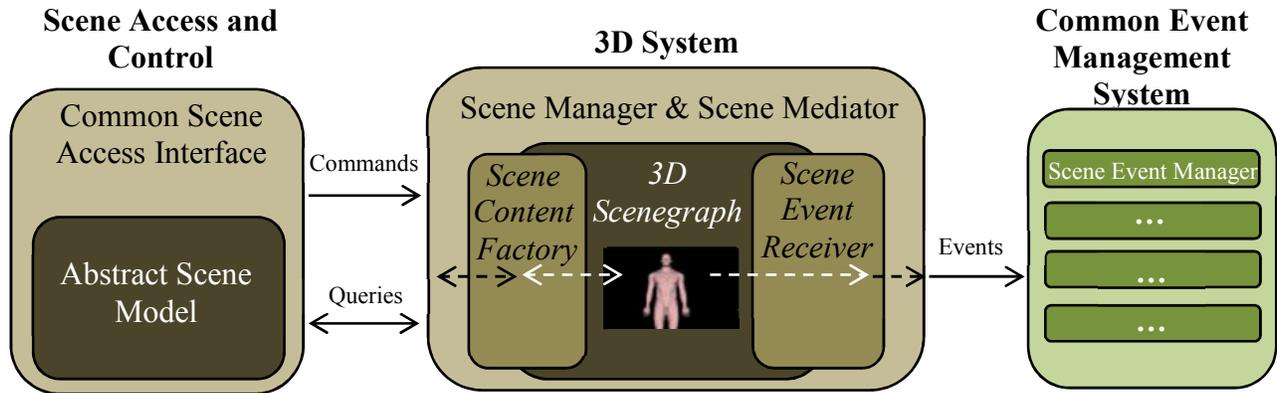


Figure 5i. Schematic of Biolucida's 3D rendering system architecture

X3D has a rich set of componentized features that can be tailored for use in engineering and scientific visualization, CAD and architecture, medical visualization, training and simulation, multimedia, entertainment, education, and more. X3D [16] has evolved from its beginnings as the Virtual Reality Modeling Language (VRML) [141] first implemented in 1993. Unlike other scene description languages which confine their scope to geometry and appearance, X3D includes a complete event model, a standardized Scene Access Interface (SAI), and a scripting component enabling logic specification within the scene itself. As a means of expressing 3D scenes, X3D clearly meets our earlier listed requirements as it supports dynamic behavior, image and movie textures, programmatic access to the scene, and currently is undergoing standardization on a volume-rendering component within its specification.

5.4.2 Default 3D framework components

The next chapter will describe key processes of creating and interacting with 3D scenes produced by a Biolucida client. In order to get a better understanding of this process, it is best to first describe the components involved.

Xj3D browser. Having an expressive scene description accomplishes very little without a means of displaying it. Xj3D [58] is an Open Source X3D and VRML browser produced by the Web3D Consortium's Source Working Group, with Yumetech, Inc. and The Naval Post Graduate School leading the effort. Xj3D is deployed as a highly componentized Java toolkit which provides application developers an embeddable viewer for Web3D content, as well as various graphics file and scene analysis utilities. The default 3D framework uses Xj3D embedded within a Java Swing container to render the content of the 3D scene. Direct interaction between the user and the scene, such as in moving the viewer's position and orientation, is accomplished by functionality inherent in the Xj3D browser. Programmatic interaction with the scene, such as changing the position and appearance of an entity in the scene, is accomplished via Xj3D's implementation of the SAI.

Scene Access Interface (SAI). The SAI is part of the ISO conformance standard for X3D browsers, where it specifies a standard set of interfaces, methods, and services which are to be made available to the developer so that they may access and modify the scene graph at runtime. The means by which these services are made available are not specified by the standard and so therefore there can be different SAI “flavors” available, such as a Java SAI, Javascript SAI, or a C# SAI. In the case of the standard 3D Biolucida framework, the Java SAI is used to enable the X3D Scene Mediator and its components to interact with the scene.

X3D Content Factory. The X3D content factory is primarily responsible for the translation of the abstract scene model, composed of Biolucida’s native primitives such as Entities, Animations, etc., into X3D syntax so that they can be inserted into the Xj3D browser’s scenegraph.

Scene Event Manager. The scene event manager is responsible for the registration of sensors contained within the X3D specific nodes of the scenegraph, located both within Entities populating the scene and within the virtual world at large. The scene event manager is responsible for broadcasting events received from these sensors to the appropriate event managers within the Biolucida client.

X3D Node Cache. Speed is a critical factor in an interactive 3D display system, and searching for entities embedded within the scenegraph can consume valuable resources leading to poor performance. Such access is required for manipulation of Entity appearance, position/rotation, and for the registration of interaction events with an Entity’s touch and proximity sensors. To optimize Biolucida’s ability to locate Entities within the X3D scenegraph, “handles” to the X3D implementation-specific nodes are stored within a hash map for rapid access.

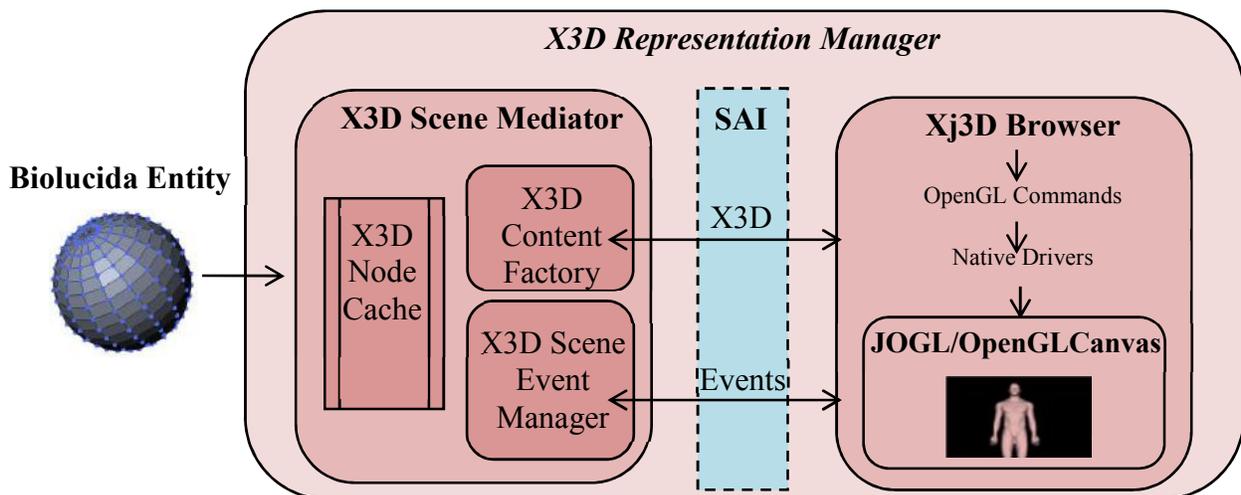


Figure 5j. Schematic of the Web3D scene management module

CHAPTER 6

3D Representation Construction Processes

6.1 Introduction

At its most basic level, Biolucida is an interactive 3D scene generator which is capable of building an immersive virtual world containing specific entities that have a variety of desired spatial and appearance attributes selected by the author. Scene construction involves a series of coordinated operations which makes the task of intelligently assembling Entities in an interactive 3D world easier for the user to accomplish than with other more expert-oriented applications. The previous chapter discussed the modules and the architecture of the client, in addition to the default 3D representation framework. Here we will describe the processes of building and customizing 3D scenes.

6.2 Client/server process execution through direct communication loop

Every client/server interaction involves the same basic communication and execution process, which moves from the client module making a request or issuing the command, to a statically-accessed Command Source which assembles a Command object and serializes it into XML for transmission to the server. The server then receives the Command and parses it to return it to its object form. The Command contains the command name, all necessary parameters, and the name of the processing class which is to handle it. The server routes the command to the appropriate processing object using this class name, where it is acted upon, and a Response object is created, sent back to the client, and the client acts upon the encapsulated Response data in whatever fashion is necessary.

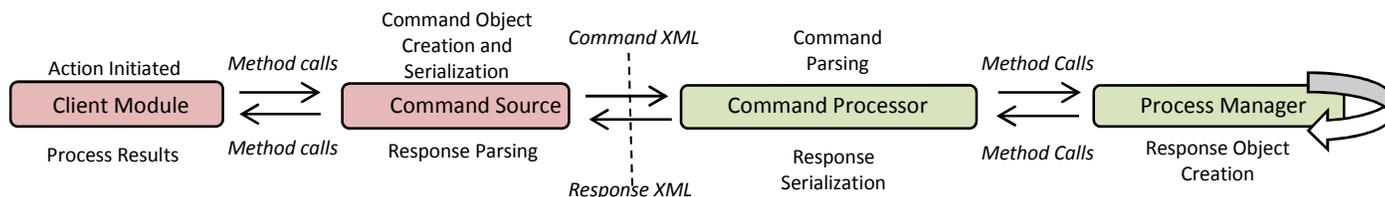


Figure 6a. Schematic depiction of a typical command response cycle in Biolucida between the server and a client

6.3 The main processes in scene creation

Now that a high level server architecture and a detailed client architecture has been previously described, it is now prudent to show in further detail how the two interact to allow an author to create and customize scenes of anatomical structures. These operations are best highlighted in the use of the Anatomical Scene Authoring (ASA) client, though it is important to keep in mind that these operations can and do occur in an automated fashion when building data driven visualizations using clients such as the DataGen Client.

6.3.1 Example: scene initialization

The first step in the process of building a scene (following login and initialization of the client), is to “load” an empty scene. This step may seem trivial at first glance, but it becomes clear that it is much more complex when considering the conventions of an X3D world, which forms the basis of the default 3D Representation Framework.

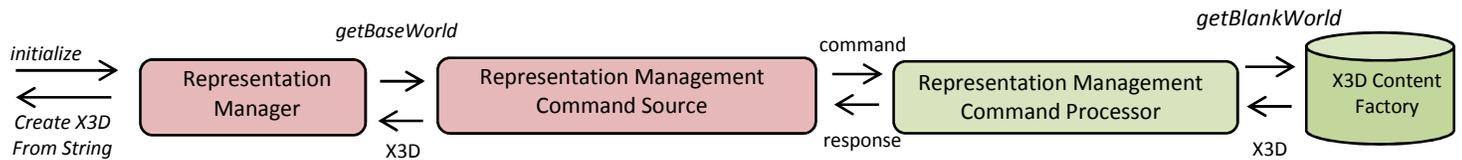


Figure 6b. Schematic depiction of the interaction between the client and the server when an empty 3D Representation is loaded. Client components are shaded red and server components are shown in green.

The request of the content necessary to load a base empty scene is one of the more basic processes in the Biolucida system, and it serves as a prototypical implementation of the aforementioned communication/processing loop. In this task, the Representation Manager (for any Abstract 3D Representation) receives a command from the system to initialize. The Representation Manager then calls a static method in the Representation Management Command Source to request the necessary X3D content describing a base scene for this Representation. The server receives and processes the request, where the Representation Management Command Processor pulls the necessary X3D content from the X3D Content Factory, transmitting it back to the client packaged in a serialized Response object. The Representation Manager then injects the new scene content into the Xj3D browser via a simple SAI method, *createX3DFromString*.

An empty scene contains some base content which not only provides a defined framework for the addition of entities, contextual objects, and temporary content, it also allows the Representation’s event model to register with certain sensor nodes in the scene which provide vital information for proper client function. This base content also contains scripting components to enable camera position updates and a gyroscopic compass to inform the user of their general orientation. The structure of the scenegraph of a base world is pictured in figure 6c.

The scenegraph structure gives us several well-defined containers (or grouping nodes) in which to place a variety of content. Each container requires its own Transform node to allow for unified motion of each content category independent of one another such as in the case of providing scaling/registration between contextual content models and entities, or moving a clip plane in relation to the rest of the objects in the world. A few special-purpose nodes allow us access to events and information which enable enhanced functionality in the client. These nodes are accessed through the SAI by either gaining reference to their exposed fields through the Xj3D SAI interface or by registering a listener which is kept informed of changes in the status of its monitored field/event.

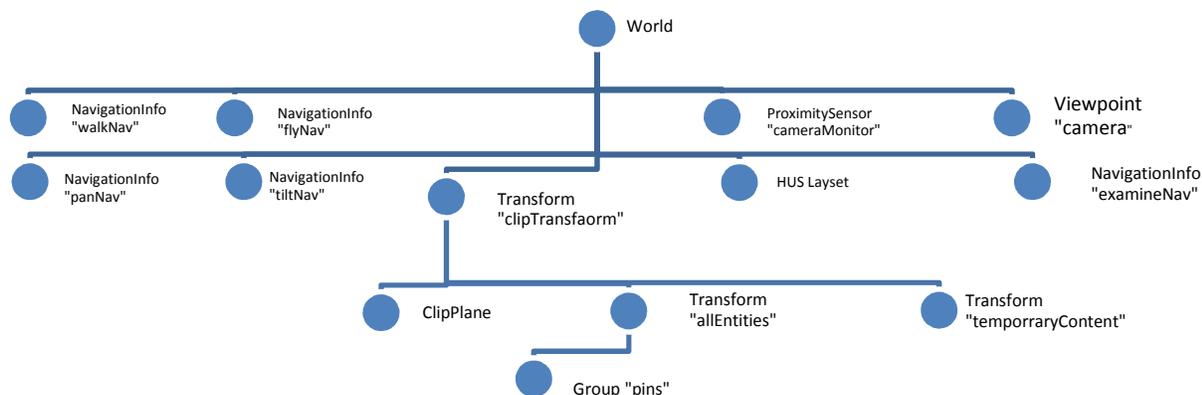


Figure 6c. X3D node hierarchy of an empty scene

| Node | Description |
|--|---|
| <i>Global Proximity Sensor</i> | Proximity sensor node which monitors the position and orientation of the user's camera view. A script node embedded within the base world transmits updates to the Camera Transform Node's translation and rotation fields. |
| <i>Camera Transform Node</i> | Stores updates of the camera's position and orientation for query |
| <i>HUD Layerset</i> | Stores 3D entities, such as the gyroscopic compass which remain in front of the camera in a "heads up display" (HUD) manner regardless of the user's change in position and orientation. |
| <i>Clip Plane Transform</i> | Holds the clip plane and enables it to be held in place while the other "containers" are translated along any of the three axes, independent of the clip plane itself. |
| <i>Global Entity Transform</i> | The container which holds all Entities in the scene, receiving such events from the scene mediator as "add children" and "remove children" |
| <i>Global Contextual Content Transform</i> | The container which holds all contextual content, or what are normally considered props – objects which have visual depictions in the scene but are not representative of concepts in the Biolucida system. |
| <i>Global Temporary Object Transform</i> | The container which holds all temporary objects, which are objects that are used as visual aids for client functions such as axes, entity and camera ghosts, waypoints, paths, and labels. |

Table 6a. Listing of the main nodes of an empty 3D Biolucida scene/Representation

References to these nodes are instantiated and cached when the browser signals it is done loading the world. Following this step, commands which pass to these nodes are then handled by the X3D Scene Mediator.

6.3.2 Inserting Entities into the scene

One of the most fundamental operations used during the construction of a scene, is the addition of new Entities. This operation involves a multi-stage process involving both the aforementioned direct communication, as well as an indirect inline method of communication meant to transfer larger amounts of data from the server's repositories to the client by means of a simple URL reference (since the *inline* Shape is the default Shape for Biolucida entities). This method is used to efficiently transfer the data contained in 3D meshes, large images, movies, and sound files from the server to the client.

The insertion of any Entity into the scene is an eleven step process which has been optimized for speed. When Biolucida was implemented in its first version, this process was vastly different and would take as much as 45 minutes to build a scene containing the human body described by 692 models. Now, that process reaches completion using the same hardware in under 3 minutes. The steps in the optimized process are as follows:

1. **Entity addition request made by a Biolucida Client module.** In this case the request is made when the user selects a set of concept names from a displayed catalog panel in response to requesting entities be added to the scene.
2. **Transmission of concept IDs.** The set of concept IDs is transmitted as parameters of the command requesting to add Entities to the current scene. Since the Biolucida Server manages multiple user sessions and therefore multiple scenes/Representations, the Representation ID is also a necessary parameter
3. **Command processing.** After the command is received and processed, the Biolucida Server uses its data access objects to retrieve shape and appearance information from the model database, and assembles a set of complete Entities.
4. **Server-side insertion.** The newly formed Entities are inserted into the server-side Representation where they are assigned unique IDs to guarantee that they can be referenced appropriately.
5. **Instance ID set delivery.** The new Entity IDs are sent back to the Biolucida Client, and the Command Source subsequently sends a command requesting the serialized entities by their new IDs
6. **Entity delivery.** The serialized Entities are reconstituted on the client-side and the Representation Event Manager is informed that new Entities have been added to the scene.
7. **Entity registration.** As a registered listener of the Representation Event Manager, the X3D Scene Mediator is notified of the new Entities. The Scene Mediator immediately caches references to each Entity's root node, storing it in a hash data structure for immediate reference when needed.
8. **Entity X3D translation.** The X3D Scene Mediator converts the new Entities into X3D nodes, using the X3D content Factory and referencing the Browser object of the Xj3D Browser to perform the conversion.
9. **Geometry import.** The new X3D nodes, since they contain inline references to their geometry hosted via URL, asynchronously download their geometry using the HTTP port via the Web Server on which the Biolucida Server is running.

10. **Sensor/event registration.** The X3D Mediator calls upon the X3D Event Registrar to register the X3D Scene Event Listener as a listener of collision, proximity, and input device events specific to each Entity. The Scene Mediator also retains references to the Material Nodes and Transform Nodes of each Entity for rapid reference when required.
11. **Appearance initialization.** The X3D Scene mediator sets each Entity’s appearance appropriately and inserts it into the rendered scene on the client side.

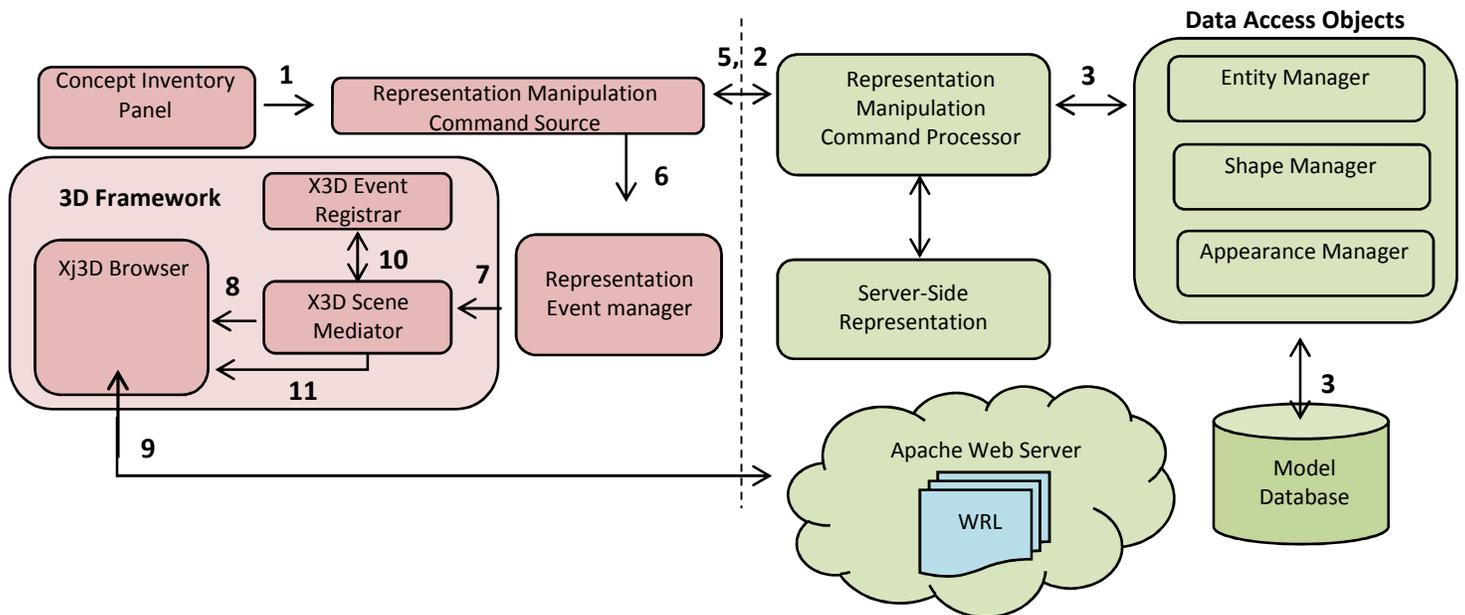


Figure 6d. Schematic depiction of the process of adding new Entities to an active 3D Representation

6.3.3 The event and node registration process

A critical piece of functionality implemented in the Biolucida Client is the ability to access, query, and create changes in the scene. Within the X3D Scene Framework, this is accomplished through the SAI system described in the previous chapter. The SAI is used to access the necessary X3D fields and events within the consistent node structure of each Entity (described further in Appendix C) and Contextual Object. However, these fields are not identified when a change or a query must be executed, but rather they are cached when the Entity is first inserted into the scene, using its unique ID as the key. It was found that, especially with large scenes, traversing the scene graph in search of the appropriate Entity and then the appropriate Node within the Entity was too resource intensive to ensure interactive speeds. Under ideal circumstances one may use the SAI to call a reference to a node by using its unique name, or DEF tag. Unfortunately, this operation was not found to be implemented in a way that would guarantee success in Xj3D 2.0RC. Therefore, a Node Accessor object is used to “find” the appropriate Entities and their fields/nodes using the Entity’s root node as a starting point. While each Node Accessor might be implemented differently to account for Representations which implement non-standard Entity node structures, the default class used caches the material, transform, image texture, and touch sensor nodes for instant access. Memory use for this

operation was found to be minimal, since the scene graph must be loaded into memory by default for Xj3D to function properly.

6.4 Example of processes and architecture in action: the ASA Client.

6.4.1 Navigation.

A three-dimensional interactive world has the distinct advantage over more “canned” depictions in that one can position the camera anywhere within the scene to attain a proper perspective view. Given that the Xj3D browser renders graphics at an interactive frame rate, the scene takes on the character of an immersive 3D experience, though this system will most often be used via desktop technologies. Navigation can occur via several different modes: flying, panning, examining (using the world’s origin as the center point), and looking/tilting (remaining stationary but altering the camera’s orientation only). Such interactive exploration makes Biolucida’s 3D Representations very expressive and versatile when used as tokens of communication for spatially intense uses such as with anatomy instruction, surgery simulation, exploration of temporal/spatial hybridization studies, and many more subject areas.

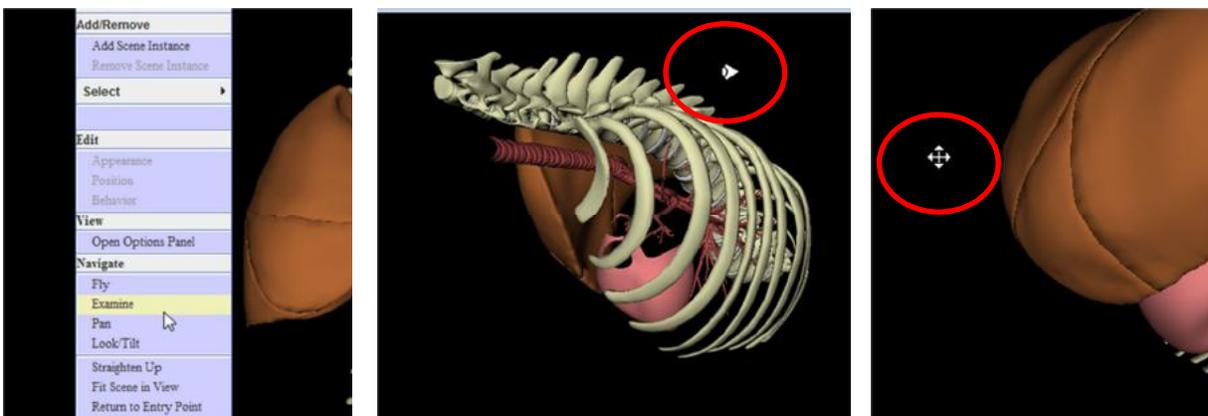


Figure 6e. The various navigation modes of a Biolucida scene. Icon-based cursors let the user know which navigation mode is active (center and right), functionality inherent in Xj3D

6.4.2 Changes in position and appearance.

After the objects have been inserted into the Representation, the author has the ability to customize the scene to their liking by changing the appearance and position of its contents. Again, use of the SAI factors heavily into the features which allow an author to perform these tasks. To minimize the learning curve for using the system, general 3D design UI and interaction conventions which are inherent in many commercial and Open Source software suites were used for these features. Controls which rely on key strokes to toggle them on and off and select the appropriate axis, along with mouse motions are used to move and position objects. Sliders and other standard UI objects are used to change the various attributes of an object’s appearance. What is different about Biolucida and these

software systems which can be considered to be 3D design systems, at least at a very basic level, is that Biolucida must propagate every change made by the user to the server's copy of the Representation.

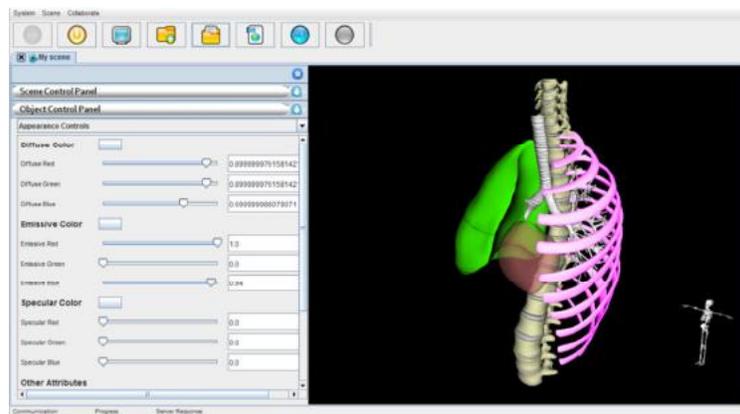
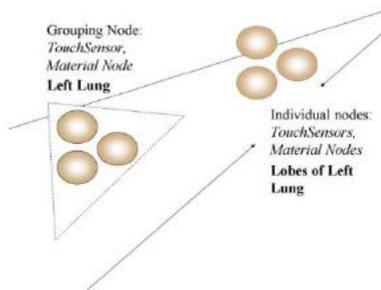


Figure 6f. Appearance change control UI in the ASA Client.

6.5 Semantic Level of Detail with respect to SAI

SLOD is particularly problematic with respect to appearance when one considers that children must inherit their appearance from their semantic parents when the Representation is in certain states. For example, a visualization may demand that the Heart be 50% transparent and its diffuse color be purple. However, the heart may have no actual 3D representation but instead relies on its children to provide its geometry – in this case it would be all of the valves, the atria, and the ventricles. However, the children may have their own required appearances in this visualization. As the user transitions between levels of detail, the system must be able to “swap” one appearance for another when display the parent or the child. Therefore, every Entity is given three separate material nodes: a “default” appearance node containing a set appearance attributes which were assigned from the values in the model database, a “scene” appearance node which dictates its proper appearance in the current scene, and finally a “current” appearance node which is changeable to enable, among other things, appearance queues for SLOD functions (as well as highlighting and in-client animation execution). The same three-tier representation is used for preservation of transform properties and behaviors between SLOD levels and edits which the author makes to contents in the scene.



6g. Depiction of the node structure used to implement Semantic Level of Detail

6.6 Server-side synchronization

Why keep a copy of the Representation on the server side current with all alterations of objects within the scene? There are two reasons: transmission efficiency and concurrent access. As the user creates a Representation and makes changes, it would be impractical to have to move the entire Representation and its contents up to the server to save it in the database for subsequent recall, or to share its content with other authors. The other reason for the server-side synchronization is to enable concurrent access by multiple users at one time. As will be discussed later in the text, Biolucida is built using technologies which are designed to enhance collaboration and enable shared experiences. Therefore, Biolucida's architecture is designed such that multiple users can connect to a single Representation and either experience a shared view of its contents, or have the ability to explore it from different points of view. Propagations to the server are managed by a set of synchronizers which register themselves as listeners for changes in the attributes of Representations and their contents. Such events will set off a command sequence much like aforementioned examples, which enact the same changes at the server.

6.7 Creating dynamic content

Biolucida is designed to create custom 3D content, and given that its environment is an interactive graphics system, animations are a key component of the Representations it allows an author to create. Biolucida has its own abstract model of animations and their atomic actions, which again allows the system to be somewhat graphics-platform independent due to the fact that these animations are simply converted by a Content Factory in the same manner that Entities and other abstractions are translated to the appropriate scene graph syntax. Animation objects are meant to describe a self-contained series of event sequences which can be played from start to finish to convey dynamic information to the viewer, whether it shows a surgical procedure, a time-series variance in data through visualization, or the careful review of anatomical parts as a portion of a complete lesson. Three main data structures compose the animation model within Biolucida:

1. **Animations** – Animations are groupings of Sequences, which all run in parallel when an Animation is started. Animations contain metadata such as a name and a description for purposes of saving and subsequent retrieval to incorporate into other scenes.
2. **Sequences** – Sequences are composed of a series of actions, which execute serially. Like Animations, sequences contain metadata for saving and retrieval.
3. **Actions** – Actions are the atomic objects of animations, they are responsible for enacting changes on a target over a defined period of time. A variety of Actions are available to authors within the Biolucida System. They are:
 - a. *Camera Motion* – Changes the camera's position and orientation over a specified period of time
 - b. *Entity Motion* – Changes an Entity's position and orientation over a specified period of time

- c. *Diffuse Color Change* – Changes an Entity’s diffuse color from one value to another over a specified period of time
- d. *Emissive Color Change* – Changes an Entity’s emissive color from one value to another over a specified period of time
- e. *Specular Color Change* – Changes an Entity’s specular color from one value to another over a specified period of time
- f. *Transparency Change* – Changes an Entity’s transparency from one value to another over a specified period of time
- g. *Text Report* – Causes a collection of text to appear in the HUD display, then pauses for a specified period of time
- h. *Delay* – Pauses the Sequence for a specified period of time
- i. *Sound* – Plays a chosen uploaded sound file (for example, a narration)

6.8 “Publishing” the Representation

If Biolucida is designed to encourage more effective communication and collaborative analysis, then it must have the ability to create artifacts which are self-contained and distributable. This requirement was one of the original reasons for giving focus to Web3D technologies. Since X3D and VRML are Open Standards, there are a number of freely available plugins, browsers, and custom applications which are capable of displaying Web3D content. Therefore it would be relatively easy for a collaborator or student to view a “published” world from an author in browsers such as Cortona, BS Contact, Instant Reality, Xj3D, and many others.

There is a difficulty that resides in translating all of the content which Biolucida is capable of producing into standard X3D. While the static elements of the scene are straight forward to translate into indexed face sets, materials, transforms, and other standard nodes, the animations are a bit more difficult. First, a UI object providing a means of choosing which Animation to play must be published along with the rest of the scene’s content. Second, a means of implementing the actions of the Animation without the aid of the SAI must be provided as well.

In order to provide the aforementioned UI when animations are present, the X3D Content Factory is capable of producing a HUD selection panel which displays the name and description of a selected animation and the controls to start it. Execution of the animations themselves relies on the scripting component of the X3D standard. By using ROUTE statements to link each action interpolator or custom script, the action is able to affect its desired target (Entity, camera, etc.) in the way that it was designed to do so. The duration of the action is control by a WorldClock node which issues *isActive* events to custom “Connection Scripts”, which serve to initiate the next Action upon its completion.

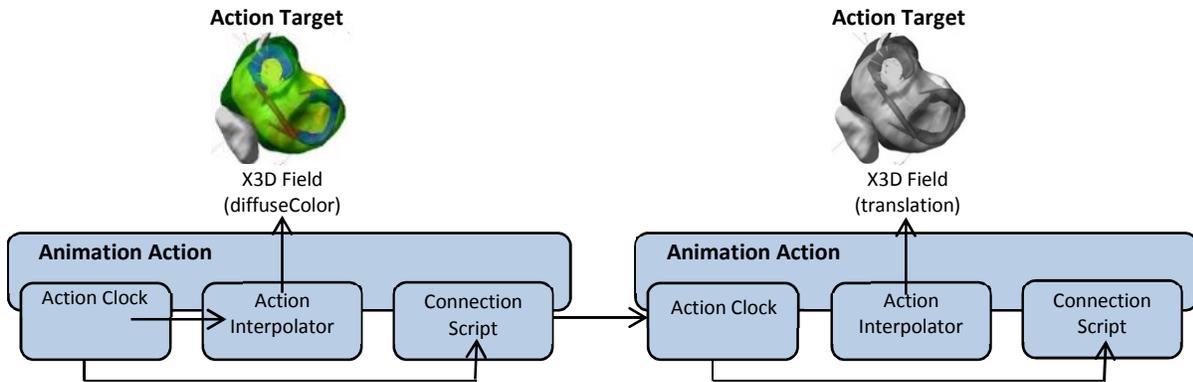


Figure 6h. Schematic depiction of the scripting and relationships between nodes when Actions in a Sequence are translated to standard X3D

CHAPTER 7

Using Data to Drive the Visualization Process

7.1 User-driven visualizations vs. data-driven visualizations

In the last several chapters, we've described how the Biolucida system constructs 3D scenes, or Representations. These types of 3D Representations were built and customized through actions of the author, and such *user-driven* Representations are valuable to communicate concepts which exist as mental models but lack adequate means of expression to be externalized. Anatomy lessons are perfect examples. Here, the author is the critical component in translating such ideas into to a comprehensible artifact – a 3D animation in this case. This chapter begins our description of a different type of 3D Representation, data visualizations, which use the scene construction processes formerly employed by a human author, but use them in producing *data-driven* scenes. In the next two chapters, we will discuss the processes behind the acquisition, parsing, attribution, and visual metaphor application which are responsible for creating a complete data visualization.

7.2 Variety and scope of biomedical data

Biomedical research is not only producing massive amounts of data for subsequent analysis by investigators, there are many initiatives to curate, standardize, and make this information available through both query interfaces for users and API mechanisms for software applications. In other words, fodder for analysis in biomedicine has never before been so plentiful and so accessible. As of this writing, Biomed central has indexed 2988 publicly available sources of biomedical research data [10], ranging from gene sequences and pathway information, to gene expression studies and medical images. It follows that a cross-domain visualization system should not only be able to illustrate the largest variety of data, it should also be able to access and process these data as well. Biolucida's data handling systems are designed with this goal in mind.

We have previously discussed the Biolucida Client's flexibility in presenting the widest variety of media types to the user within its 3D environment. Images, video, textual data, and 3D models are all rendered for the user within the same environment, according to the design intent by the Representation's author. These capabilities are supported by the rich environment of Xj3D and the X3D standard. With such capability, the scope of data which Biolucida can handle increases beyond what one would consider adequate for a typical domain specific system. However, this variety of data available to Biolucida brings with it three challenges: (1) the sheer number of potential sources of information make centralizing development of the access system an implausible prospect, (2) the current and growing collection of data sources lack a uniform means of access and format, and (3) the sources of data will evolve over time, changing their means of access and the format in which the data exists. While (1) is a question of the scope of the system, (2) and (3) are problems which have perpetually plagued the data integration field for some time. Rather than constrain the scope of Biolucida to the point where it would become yet another specialized 3D

system, or suggest a new method for standardizing access and account for schema evolution, Biolucida uses its familiar plugin/crowdsourcing approach to address these challenges.

As we mentioned earlier in Chapter 3, the fundamental advantages of crowdsourcing which are relevant to Biolucida are that small groups of stake holders are able to take ownership of the design and implementation of the application specific portions of the system. In regard to data sources, this entails (1) understanding and being capable of representing the application and data needs of a particular domain, (2) surveying the available sources of information regarding that domain's area of knowledge, and (3) maintaining valid access to those valued information assets. Given that Biolucida is designed with the intent of serving as a discipline-wide visualization engine for Biomedicine, the aforementioned responsibilities would be unreasonable for any one team to take on. Therefore, Biolucida has adopted a plugin architecture with respect to its data acquisition systems, similar to the manner in which it manages its collection of Representations.

7.3 The Dataset creation process

All Representations within Biolucida are produced from datasets. The Representations which are merely 3D scenes are no different. Rather, their datasets are unpopulated with source records. Biolucida employs a three-step process in populating its datasets: (1) acquisition of data through a Data Source, (2) parsing and normalized storage of the data into an appropriate Container, and (3) propagation and transformation of the data along a valid relation. The result of these three processes is a dataset containing records which can be attributed to concepts (anatomical structures in this case), and a visualization subsequently produced by inserting the Entities into a scene and applying various visual, positional, and behavioral metaphors to them.

7.3.1 The data management process primitives

It is best to describe the architecture of the data management system in order to gain an understanding of the modules involved in the process and their relationship to one another. The data management system is composed of four main components: (1) the Data Sources, (2) the Data Handler, (3) the Data Container, and (4) the propagation/transformation system.

Data Sources. Data Sources are essentially wrappers [82] for a resource or service, and are primarily responsible for the retrieval of data to be presented to a Data Handler for subsequent parsing/processing. Biolucida provides a set of basic Data Sources which are capable of retrieving data from a local file, local database, remote document via URL, remote database, and from Web Services (Servlet). The developer is free to extend and customize the functionality of any of these default classes to meet the source needs of their Data Handler. Below is a graphical diagram of the data population process and the components involved.

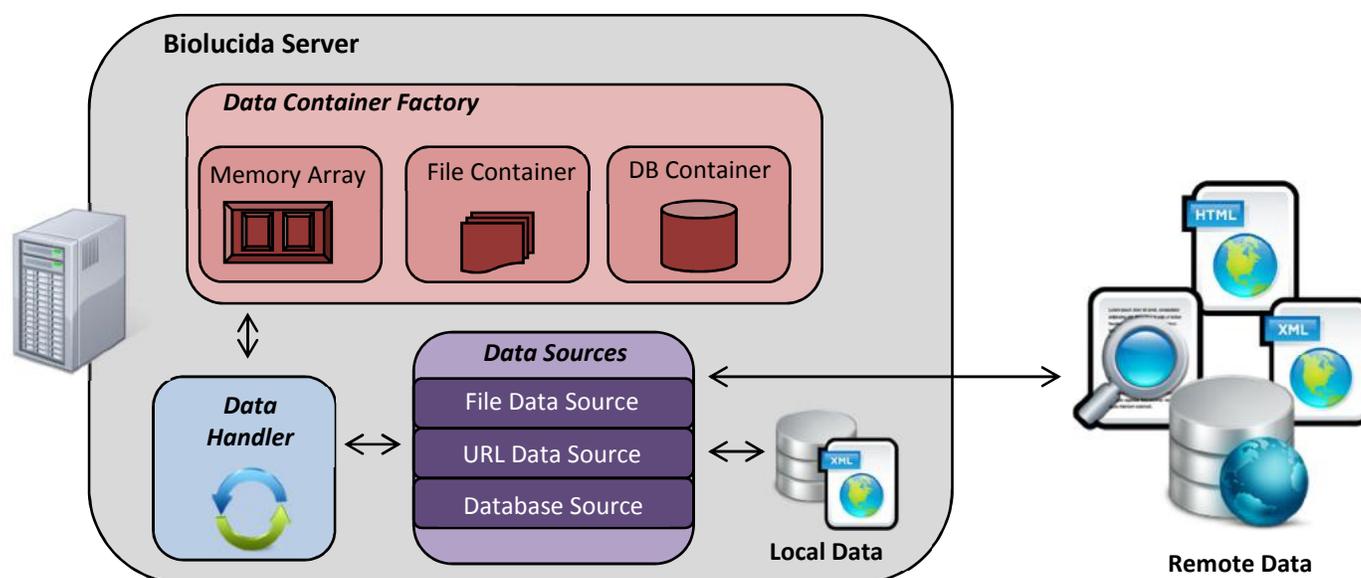


Figure 7a. Graphical depiction of the data retrieval process and its components

Data Handler. The Data Handler is responsible for the parsing of the data into a standard format for subsequent caching/storage. Data Handlers are by far the most custom piece of any of the components of the data retrieval system, as they are tailored to the specific resource(s) which they are to parse for the system.

Data Container. Data Containers are abstractions of persistence services, capable of storing and retrieving records produced by the Data Handler as it parses the raw data from its Data Sources. The Biolucida System uses a factory class to interrogate the resources which are available on the server. Given that there may or may not be a running available database on the Biolucida Server, or that the dataset which is to be stored is beyond the capacity of the working memory of the system, the Data Container Factory chooses the appropriate storage class to use given these parameters. Currently, there are four types of Data Containers available within Biolucida:

1. *Memory array* – Records are kept in memory using a list type data structure and is searched serially when queried
2. *File Container* – Records are written out to a generic file using XML. When queried, the file is searched sequentially from the beginning
3. *Indexed File Container* – Records are written out to a `RandomAccessFile`, along with metadata that includes pointers to records having indexed values.
4. *Database Container* – When available, a relational database is used to store records, and querying is performed via standard SQL.

Data Propagation/Transformation System. Data propagation is a complex process within Biolucida, enabling the system to make inferences about data which can be attributed to concepts which are not found in the original source data. This process mitigates some of the challenges which are brought about from incongruities between terms in

the original dataset, the native concepts in the Biolucida system, and the 3D models which are available within the Biolucida model database. Providing a data-centric Semantic Level of Detail feature where data can be investigated at multiple levels by the user also requires the functionality provided by the data propagation system. The data propagation and transformation processes are discussed in great detail in subsequent sections.

7.3.2 Anatomy of a dataset

The Data Containers are responsible for holding the contents of a Dataset, which is implemented using a simple table motif containing Records and Fields, which fill the same role as their name suggests. Fields contain the metadata necessary for identifying and classifying any datum which populates its appropriate place in the Record, such as name, description and format. The Fields are contained within a Record Manager, which besides serving as the metadata container for the Dataset itself, provides a set of convenience methods for locating the index of desired Fields, querying the size of the dataset, as well as adding and removing columns.

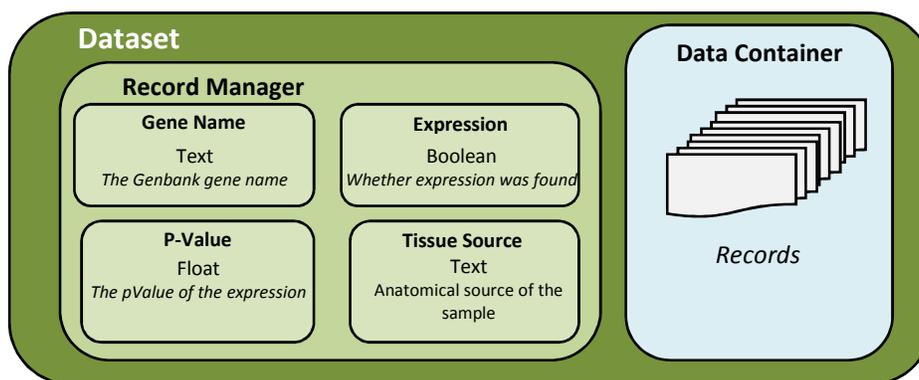


Figure 7b. Graphical depiction of a dataset and its components

7.4 Semantic attribution: aligning records with anatomical concepts

After source data are acquired and processed into the correct format and stored in a Data Container, the individual records are associated with Biolucida concepts. It is through this process that Biolucida will determine which Entities belong in the scene. Our ability to create an informative visualization of experimental datasets using Biolucida's object-oriented approach depends heavily on our ability to recognize semantic connections between a record in the dataset and a concept in the Biolucida catalog. Otherwise, it would be impossible to associate data with any of Biolucida's 3D models, and the system would lack the information required to identify the set of models with which to populate the scene.

Biolucida cannot assume that references to appropriate anatomical structures will already exist as available information in the dataset for three reasons. First, Biolucida is designed to digest arbitrary datasets of which one can make very few assumptions regarding their content. Second, information specifying the anatomical source of a

sample is not a compulsory requirement for prior submittal by researchers in many fields, leading to potential omissions in the data. For example, the MIAME standard [14] governing gene expression studies requires several items of metadata regarding the experimental variables, but does not require the anatomical source of the cell sample. Third, it is a virtual certainty that any anatomical structure referenced by a dataset will require some lookup function to find the equivalent native Biolucida concept for the “foreign” anatomical term. Such lookup functions must overcome three challenges in selecting the appropriate content with which to populate the scene.

1. **Proper concept selection.** The correspondence of anatomical terms in the source data won’t always be 1:1 with the concepts in Biolucida’s catalog. As mentioned previously, a developer will have to embed some term lookup logic inside of a Data Attributor to overcome this hurdle.
2. **Model availability.** Even though an appropriate native concept might be identified, this does not mean that this concept has an appropriate model reference within the Biolucida database. A developer must use Biolucida’s ontological relations (such as mereology, or part-of relations) to solve this issue.
3. **Context provisioning.** Small samples of source data may produce scenes containing the models of obscure anatomical structures which are difficult for any user, besides one who has background in gross anatomy, or worse, a scene may be produced with no valid models at all. This problem is compounded by the fact that the scene will be rendered within a navigable 3D world, where it is possible that the user may be viewing the scene in an unfamiliar orientation where it becomes impossible to identify such obscure structures. Providing context, such as cartographers providing their readers with smaller scale “location” maps inset within larger maps, is pivotal to properly orienting the user. In Biolucida’s case, a greater scope of model inclusion into the scene is used to provide context.

Biolucida contains a framework which implements lookup algorithms, ontological data propagation, and interactive semantic level of detail to overcome these challenges.

7.5 An example of the data-driven scene construction process.

It would be unproductive to abstractly hypothesize how a potential developer would deal with the anatomical references of a totally arbitrary dataset that they are attempting to digest and make available to the system. Rather, it is best to focus our discussion on the use-case examples mentioned in Chapter 3 and their potential gene expression datasets. We will discuss how the attribution of data to anatomical structures is performed in context of these data. To this end, suppose we have imported a set of records composed of the following data:

| Record ID | Gene Name | Expression | P-Value | Tissue Source |
|-----------|-----------|------------|---------|--------------------------------------|
| 1 | ESR1 | True | 0.03 | Cerebrum |
| 2 | VHL | False | 0.05 | Lungs |
| 3 | ESR1 | False | 0.04 | Left Fibrous Trigone of Mitral Valve |
| 4 | INSR | True | 0.07 | Blood |

Table 7a. Example record set imported into Biolucida.

Each record in our sample dataset is illustrative of four situations which the system may encounter when attempting to “assign” these data to the appropriate anatomical structure which will eventually populate the 3D scene. These situations each use a different strategy for such attribution.

7.5.1 Illustrating the role of ontological frameworks in data-driven visualizations.

The conceptual instance “nodes” which are found in a Biolucida visualization scene are organized into a directional acyclic graph (DAG). The resulting structure not only forms the basis of semantic level of detail, but also a framework for relations from concept to concept where data-related inferences can be made. Graphic scenes are often constructed using mereologies to create its fundamental structure. However, the DAG which forms the organizational structure of a visualization scene can be built on any ontological relationship stored within Biolucida. In fact, a visualization scene can contain any number of hierarchy relations. Here, we offer arguments as to why data propagation via non-mereological relations can be an effective precursor to concept selection, and later, model insertion.

Within a given visualization’s ontological framework, there are nodes populated with 3D models, while others do not have such visual representations. These nodes are either higher-level amalgams of their leaf children (which often have models), such as the spine is an amalgamation of its vertebra, or they are more abstract nodes existing as obscure leaf concepts or isolated nodes of the ontological framework entirely (such as the concept of a high risk operative area of the colon which may not exist within one of the standard ontological relations). In any case, each record which contains a foreign concept term must have a Biolucida-native concept term inserted into the record so that it may be attributed to that concept for the purpose of subsequently building the visualization.

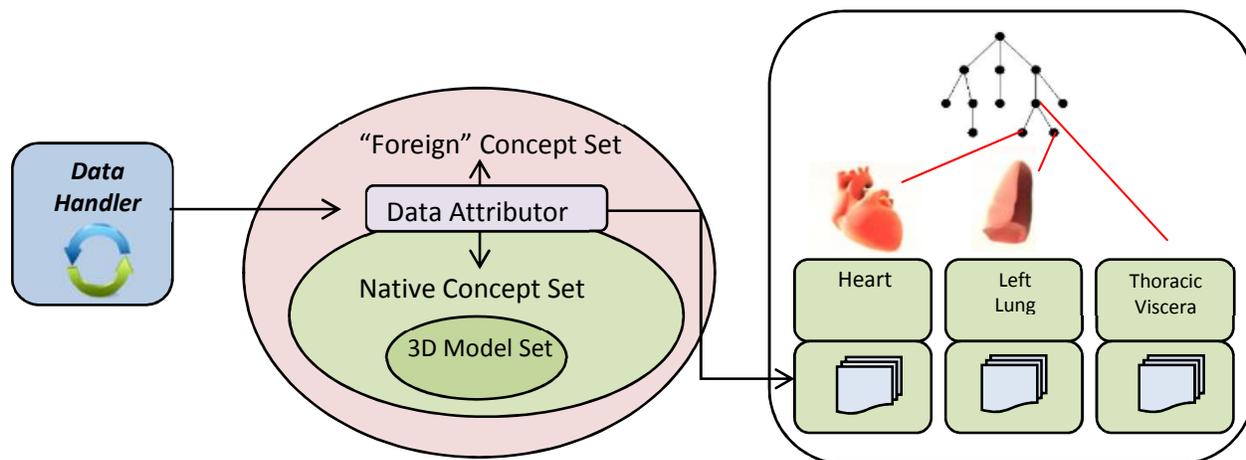


Figure 7c. Graphical depiction of the data attribution process. The construction of a data-driven visualization entails an alignment between the anatomical concepts noted in the source data, the concepts represented in Biolucida’s reference ontology view, and the set of models to which the system has access.

7.5.2 Native concept selection and insertion

The selection of the appropriate native Biolucida concepts from the foreign anatomical terms found in the source data is an endeavor far too large for a single system deployment considering that Biolucida is designed to interrogate arbitrary biomedical datasets. Therefore, Biolucida's plugin approach is perfectly matched to the task – where external developers are provided with an extensible framework to allow the implemented lookup algorithm to be specific to the dataset it is designed to interrogate. The data management system developed for Biolucida's gene expression datasets uses a simple correspondence table, where the foreign terms in the dataset are listed with an appropriate native concept, which is inserted into the produced records at runtime.

7.5.3 Model selection.

Following the identification of native concepts to insert into the source data, Biolucida must then determine which models to insert for the purpose of visual representation. In the event that all concepts are not represented by models found in the Biolucida 3D model database, appropriate action must be taken to ensure these concepts are visually represented in the scene in some manner. There are several different scenarios that the system should be prepared to handle.

1. **Correspondence selection.** Direct correspondence where the anatomical term in the data matches the name of a concept which has a representative 3D model is rare, but does occur. Looking at our dataset in table 7a, record 1 shows that we have data describing that the GMV5 gene is found to be expressed in samples taken from the Cerebrum. The Biolucida system contains a concept with the name “Cerebrum” in its catalog, and also contains an appropriate 3D model in its model database. In such cases, data is attributed to an anatomical instance and the model is simply inserted into the scene.
2. **Mereological selection.**
 - a. **Downward selection.** Downward ontological correspondence occurs when the anatomical source in the data corresponds with a Biolucida parent concept lacking an assigned model. Row 2 in the data specifies “Lungs”, and while Biolucida may have a concept called “Lungs”, or “Set of Lungs”, the concept itself does not have its own 3D model. Rather, Biolucida has models corresponding to the concepts “Right Lung” and “Left Lung”. Traditional use of a metrological relation for concept selection will require that Biolucida add the children to represent the parent. In this case, the parent concept is mereologically related to children which have representative models, and therefore the child nodes are inserted into the scene, along with the parent concept as a grouping node, for purposes of semantic level of detail. The records, in this case, are attributed to the parent grouping node.
 - b. **“Upward” selection.** Upward ontological correspondence occurs when the anatomical source referenced in the data corresponds with a very low-level (ontologically) entity which either exists as a concept without an assigned model, or is addressed by the developer of the Data Handler in question. This situation is common for anatomical structures noted in datasets that lack either

proper geometric representation, (such as cavities and spaces, such as the optic cavity of the skull), or formal definition (such as regions of colon). In row 3, the required structure, the left fibrous trigone of mitral valve, is a concept within the Biolucida system, but the entity is too small to warrant its own 3D model. In this case, the representative model can be the entire mitral valve, or perhaps the heart itself.

3. **Deliberate Selection.** Situations can arise where the data may reference a concept which either is not related to any native concepts found within Biolucida’s catalog, or the native concept lacks any appropriate geometric representation, such as “blood”. One may also encounter situations where expression experiments will have data points which are associated with structures with no names, such as certain small veins. In these cases, the developer must assign concepts to records in a deliberate manner.

7.5.4 Shortcomings of simplified attribution processes

While the aforementioned approaches can produce informative visualizations given a dataset of adequate size and breadth, it also suffers from potential shortcomings. These arise from the simple two-step process of native concept identification followed by mereological model insertion, which can mislead the viewer and produce suboptimal visualizations.

The visualization may not be representative of the data. Even though appropriate concepts may be identified for the corresponding anatomical terms in the dataset, simply including only the mereological children of those concepts gives the false appearance that the data apply to those structures, rather than the parent concepts. For example, if we had a dataset which referenced positive gene activity of interest for the bronchopulmonary segments, the model selection algorithm may insert the bronchial tree as well as the lungs. However, this visualization is misleading because it cannot be properly said that the lungs express that gene, since they contain many different types of cells, each with their own expression profiles. Browsing the data via semantic level of detail may mitigate this distortion, but the resulting visualization loses its effectiveness.

Provenance is unclear. Another shortcoming of directly selecting concepts and inserting their mereological children into the scene is that the logic behind the model selection is unclear to the viewer. Without such provenance, the value of the visualization is decreased significantly.

Trivial mereological saturation. In cases where a selected term is a higher level mereological concept, such as *Nervous System*, the selected child models which will populate the scene include every nerve in Biolucida’s catalog. While such a visualization conveys the right impression of the data on the viewer, it is very unclear at first which structure(s) contain appropriate source records.

The scene may lack context. In the case where both concept and model represent very low-level members of a mereological relation, the scene can be limited to small and possibly obscure structures. While it may be clear which structures have attributable data to those who have anatomical training, the greater context of the scene is lost.

7.5.5 An alternative and superior data-driven strategy: data propagation and subsequent scene population

The aforementioned method of constructing visualizations by selecting single corresponding native concepts which will be associated with the appropriate data, followed by the insertion of those concepts' mereological relatives so that the scene incorporates proper 3D content, has limitations which became quickly apparent following the alpha testing of Biolucida's early prototypes, described above. An alternative to this simplified strategy is to implement modules which are responsible for propagating the data along select ontological relations after the initial assignment of records from the source data to the "seed" concepts, followed by model selection and insertion. The process of visualization construction that Biolucida implements in its most recent deployment uses data propagation to establish a larger set of concepts with which to seed the scene, followed by the insertion of their more atomic mereological children of which they are composed.

7.6 Data propagation system architecture and execution process

Like many of its other systems, Biolucida uses a modular and extensible approach to implement its Data Propagation abilities. The data propagation is carried out by a plugin object called a Propagation Package that contains individual classes which are each responsible for propagating a single data type along a single relation in the Biolucida catalog.

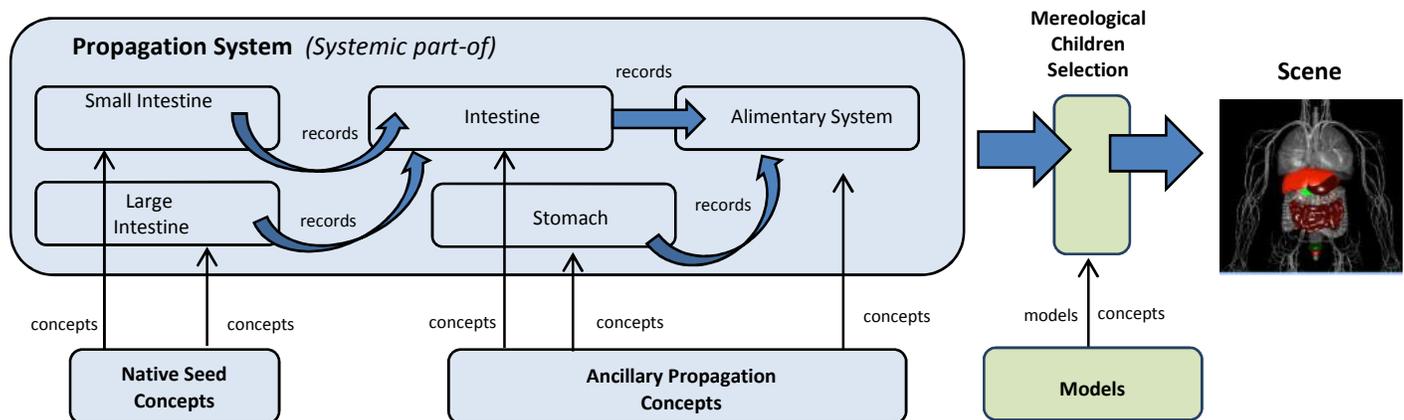


Figure 7d. Summarized graphical depiction of the data propagation process.

Figure 7d shows a simplified view of the propagation process, where seed concepts have been selected already and imported into the 3D Representation. Records attributed to the seed concepts are propagated forward along a given relation, in this case *systemic part of*. Table 7b shows the changes in the dataset which can result from this record propagation process. In this case, new records are created which attribute themselves to the next concept encountered in the relation. There are also new fields introduced which assist in establishing provenance. More

specifically, the *propagation steps* field shows how many “jumps” the data has made to land in this place, while the *propagation source* field notes the origin record ID, and the *transformed* column designates whether the record is the result of a transformation operation. A transformation, discussed later, is a process by which a record’s values are used as input to produce an entirely new field and value to be inserted into the dataset.

| Record ID | Gene Name | Expression | P-Value | Tissue Source | Native Concept | Propagation Steps | Propagation Source | Transformed |
|-----------|-----------|------------|---------|--------------------------------------|--------------------|-------------------|--------------------|-------------|
| 1 | ESR1 | True | 0.03 | Cerebrum | Cerebrum | 0 | -1 | False |
| 2 | VHL | False | 0.05 | Lungs | Set of Lungs | 0 | -1 | False |
| 3 | ESR1 | False | 0.04 | Left Fibrous Trigone of Mitral Valve | Mitral Valve | 0 | -1 | False |
| 4 | INSR | True | 0.07 | Blood | Arterial Tree | 0 | -1 | False |
| 5 | HBB | True | 0.02 | | | 0 | -1 | False |
| 6 | VHL | False | 0.05 | Lungs | Respiratory System | 1 | 2 | False |
| 7 | ESR1 | False | 0.04 | Left Fibrous Trigone of Mitral Valve | Respiratory System | 1 | 3 | False |

Table 7b. Results of data propagation performed on source data from table 7a. In this case the data is propagated along the part-of relation.

Data propagation in this sense was described previously by Halper [45], but mainly focused on this type of process in databases. Halper, et al, suggest methods for propagation focused exclusively on *part-of* relations. However, the *part-of* relation takes on more of the character of an inheritance relation within the world of database design, where propagation of attributes from child to parent and vice versa is more formalized and affords us a more constrained analysis of the rules governing such propagation. The propagation style noted in table 7b follows a mereologically-oriented strategy, which is the least logic-intensive. Due to their expansive scope, domains such as biomedicine require a more expressive means of describing relations. Therefore, Biolucida supports other methods of propagation, both hypothetical (table 7c), and those that have been implemented by the Biolucida prototype, described in chapter nine.

When propagating data in a given field over a non-mereological relation, the process may require a propagation module using modified logic, as the rules for propagating data will undoubtedly change when dealing with novel combinations of data and relations. Relations used for data propagation can vary according to the types of data involved. Some possibilities are listed in table 7c.

| Data Description | Relation | Rationale |
|---|------------------|--|
| Research citations about a given anatomical structure | Systemic part-of | Biomedical research often focuses on elucidating insights about <i>function</i> rather than structure and composition. Therefore citation records would best be propagated along a relation that organizes its members as functional |

| | | |
|--|--------------------|---|
| | | units. |
| Patient records showing locations of penetration wounds | Regional part of | A visualization which seeks to show most common areas to receive stab and gunshot wounds would be concerned mostly with <i>where</i> the wounds were located. Using a relation which focuses on spatial areas of the human body would create a more meaningful visualization for answering such <i>where</i> questions. |
| Toxicity levels of certain bodily fluids | Contains | This type of visualization would deal mostly with concepts such as “blood”, “mucous”, “spinal fluid”, etc., which are concepts that do not fit neatly into a part-of relation. Rather an is-contained-by relation could be used to visualize the structures in the body where these fluids are generally found, turning the resulting 3D models into abstractions of the attributed concepts. |
| Patient records show muscles and muscle groups affected by a neurological disorder | Innervation source | A visualization which shows this data might present itself quite well using traditional approaches. However, it may be more informative in a cause-and-effect manner to use the suggested relation as it would illuminate the nerves which are undoubtedly involved in the disorder, adding additional knowledge to the visualization. |

Table 7c. Listing of data types which could produce informative visualizations using a variety of relations.

7.7 Semantic Level of Detail’s role in data-driven visualizations

We’ve discussed how the ubiquitous levels of granularity in biomedicine are represented within Biolucida’s visualizations as Semantic Level of Detail (SLOD). Since SLOD can be used effectively to manipulate and visually interrogate the anatomical concepts of a given relation, it can also be used to browse data that has been propagated and therefore exists at multiple SLOD levels in the visualization. In fact, the resulting visualization from a propagated dataset would be of limited value if this were not the case. However, such propagation among many levels of concepts brings about an interesting question of relevancy: can (or should) data be allowed to move along a relation indefinitely and should its utility at a given level of propagation be taken into account? It is difficult to discuss possible answers to this question without giving some examples. First, it is prudent to introduce the concept of data *transformation*.

7.8 Moving from propagation to inference: data transformation

Suppose we are creating a visualization from the dataset originally driven by the data in tables 7a and 7b. If propagation continues, then we will eventually produce records attributed to the “respiratory system” and the entire human body itself. While it may be logically valid to say that the GMV5 gene is expressed (somewhere) within the body because one of its propagation relation children contains a record with a positive expression for GMV5, it is not highly informative at such a high level. However, if new data were introduced as a result of the mitral valve expressing the GMV5 gene when the body node was reached, such as a phenotype or a physiological characterization, the utility of records at that level would increase markedly. This is especially true when one takes

into account that different users may focus on different SLOD levels within the same Representation. Each of the users, depending on their domain of expertise, may be interested in different types of data.

Biolucida contains modules which reside within Data Propagators and are responsible for using existing data to introduce new fields into the dataset, and to continue the propagation process using these new data. These modules, called Data Transformers, are event driven, and cause the spawning of new fields and new Data Propagators when specific nodes and data conditions become existent within a given data set.

| Column ID | Gene Name | Expression | P-Value | Tissue Source | Native Concept | Physiological Function | Propagation Steps | Propagation Source | Transformed |
|-----------|-----------|------------|---------|--------------------------------------|--------------------|------------------------|-------------------|--------------------|-------------|
| 4 | INSR | True | 0.07 | Blood | Arterial Tree | | 0 | -1 | False |
| 5 | HBB | True | 0.02 | | | Null | 0 | -1 | False |
| 6 | VHL | False | 0.05 | Lungs | Respiratory System | Null | 1 | 2 | False |
| 7 | ESR1 | False | 0.04 | Left Fibrous Trigone of Mitral Valve | Respiratory System | Decreased blood flow | 1 | 3 | True |

Table 7d. Data transformation results during data propagation, spawning physiological function data from expression information in specific bodily structures.

| Column ID | Gene Name | Expression | P-Value | Tissue Source | Native Concept | Physiological Function | Propagation Steps | Condition | Propagation Source | Transformed |
|-----------|-----------|------------|---------|--------------------------------------|--------------------|------------------------|-------------------|-----------------|--------------------|-------------|
| 6 | INSR | False | 0.05 | Lungs | Respiratory System | Null | 1 | Null | 2 | False |
| 7 | ESR1 | False | 0.04 | Left Fibrous Trigone of Mitral Valve | Respiratory System | Decreased blood flow | 1 | Null | 3 | True |
| 8 | Null | Null | Null | Null | Body | | 2 | Chronic Fatigue | 7 | True |

Table 7e. Further data transformation results following initial transformations, producing “conditions” from the descriptions of physiological functioning

Such capabilities which transform data may use inference, signifying that Biolucida can serve not only as a visualization system within specific application domains, but also as a system which implements the basic concepts of translational medicine. Translational medicine is a discipline which studies the flow of information between biomedical disciplines, from molecular studies to clinical care provisioning and further on to policy formation within public health. In the process of bench-to-bedside knowledge propagation, information will serve researchers with much different responsibilities, knowledgebases, and roles. It makes sound logical sense that a system which is

designed to meet the needs of the greater Biomedical research community, such as Biolumina, should be concerned with the relevancy of data to each stakeholder as it crosses cellular, tissue, and system boundaries. Such is the primary justification of data transformation.

CHAPTER 8

Determining Visualization Characteristics: Visualization Schemas

8.1 Introduction

In previous chapters, we've described how the Biolucida system creates virtual anatomical scenes, publishes distributable content, and processes data to be associated with concepts/entities for purposes of visualization. Here, we describe the method by which visualizations are created, or more specifically, how Entities with associated data are processed in a manner that alters a number of visual and behavioral attributes to create informative Representations. A visualization process pipeline has been implemented within the Biolucida server system which is responsible for evaluating the data attached to an Entity against customizable rules/conditions which change the appearance, position, behavior, ancillary added content, and even the selection of the proper 3D model itself. This process allows visualizations to be easily customized by not only developers, but also by users at runtime.

8.2 An overview of Visualization Schemas

In following with Biolucida's modular approach to the implementation of many of its functions, the visualization process pipeline is designed to separate the components which are responsible for the determination of key variables of a visualization from the less artistic processes of selecting the entities and assigning the data before the scene is built. The difference between building a basic virtual anatomical scene and a data-driven visualization is that each entity is evaluated by a non-null Visualization Schema Set, composed of individual customized Schemas. A Visualization Schema implements the functionality necessary to interrogate an Entity's associated data, and make updates to the Entity's attributes before its insertion into the visualization scene.

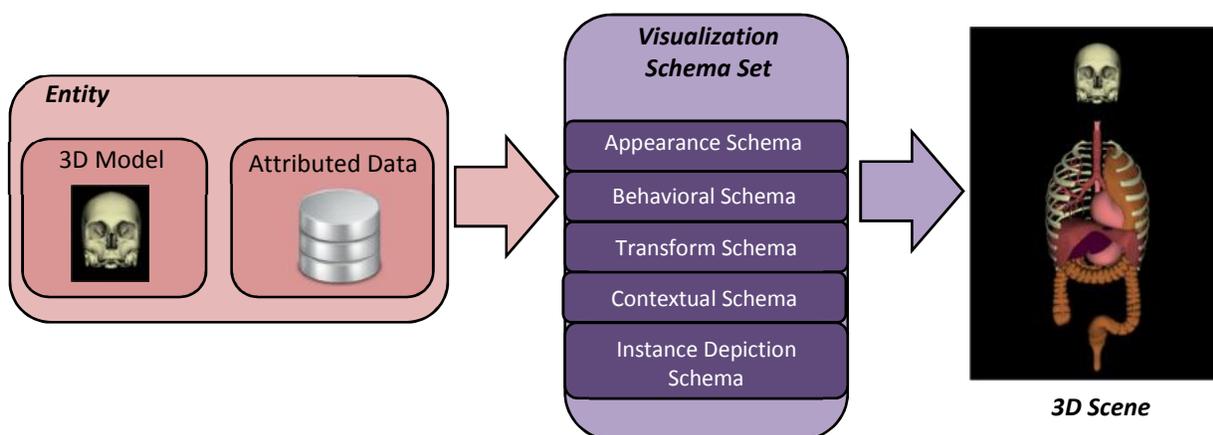


Figure 8a. Depiction of a Visualization Schema Set

Biolumida defines interfaces and abstract classes for five types of Visualization Schemas, representing the visual and functional attributes which define the look and feel of an atlas-driven visualization.

- **Appearance Schema** - The most commonly used visualization schema is the Appearance Schema, which is responsible for affecting the diffuse color, emissive color, specular color, transparency, shininess, ambient intensity, and texture of a given entity.
- **Behavior Schema** – The Behavior Schema is responsible for applying changes to the dynamic behavior of an Entity, meaning that it adds animations which are started by events such as the initialization of the scene, the user approaching the entity within a critical distance, the selection of the Entity by the user, or the Entity transitioning into/out of view.
- **Transform Schema** – Not all visualizations will use the default transform settings of an Entity added to the scene. The Transform Schema is responsible for applying changes to the position, rotation and scale of an Entity.
- **Contextual Schema** – Some Entities may require that other Entities and ancillary content be added to the scene to either give proper context to the Entity or to be used as additional visual metaphors representing the underlying data. For example, a data-driven visualization which produces a scene populated only by a few small veins and arteries may require that additional organs and bones be added to properly orient the viewer. Another situation which requires the actions of the contextual schema is one where the Entities are given floating 3D “icons” as representations of their attributed data, much like symbols in a map.
- **Instance Depiction Schema** – The instance depiction schema is responsible for the modification of the 3D model representing an Entity in the scene. These modifications can include warping the surface mesh of the model beyond the capabilities of the transform schema, or swapping the model out completely for a selected variant. These actions may be required in situations where the data reflect changes in morphology, such as alterations of 3D entities to produce a visualization showing potential facial abnormalities in children based on genomic information.

8.3 The visualization schema process pipeline

In order to initiate a change in a given Entity’s attributes, Visualization Schemas must compare the attributed data of an Entity against a set of conditions which serve as triggers for applying any such changes. These conditions are contained within a data structure called a Value Lookup Table (VLT), held inside each schema instance. The VLT is composed of an ordinal list of condition/value pairs. The contents of the Entity’s attributed data are interrogated, and if a condition evaluates to *true* then the condition’s corresponding value is applied to the Entity. For instance, the appearance schema contains Material objects, which define an Entity’s visual attributes. If any of the appearance schema’s VLT conditions evaluate to *true*, then their associated Materials are applied to the entity.

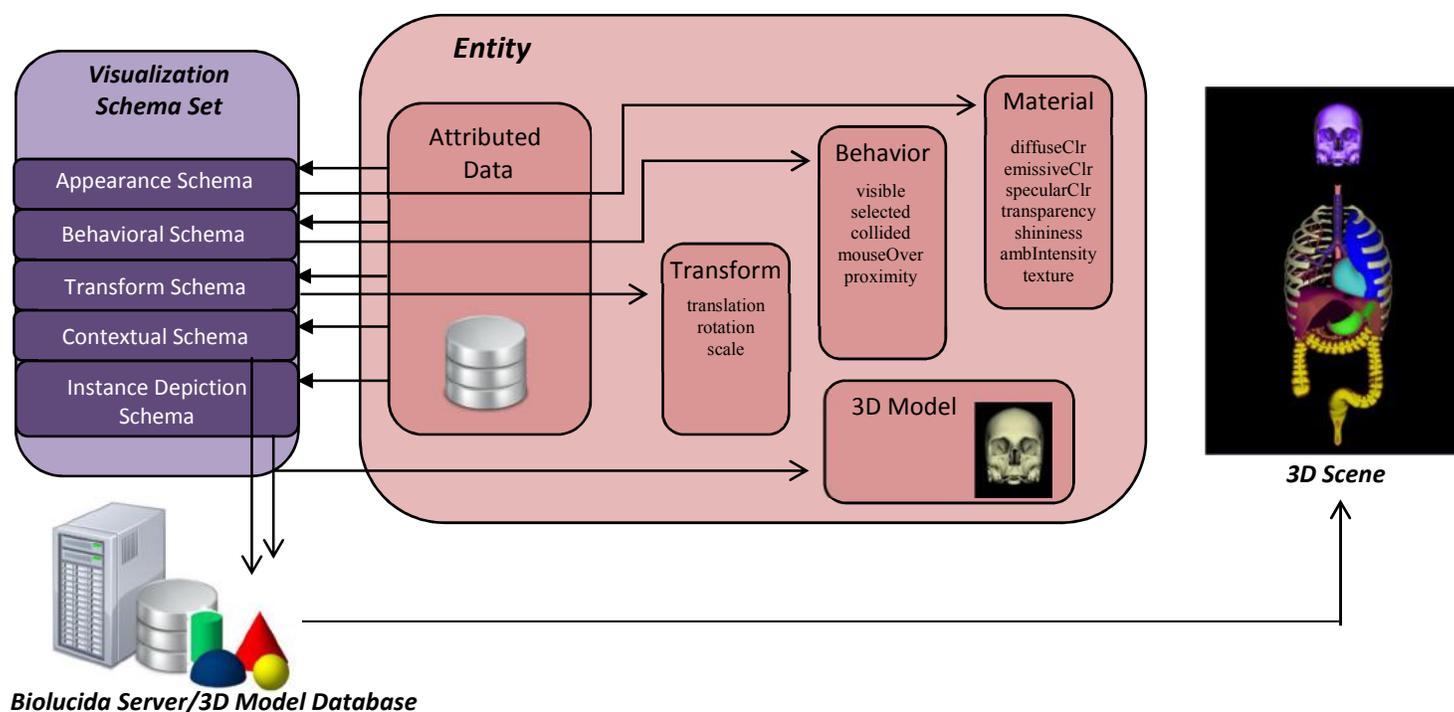


Figure 8b. Schematic of the process of a visualization schema set acting upon an Entity and upon the scene itself.

These conditions, or Value Lookup Entries (VLEs), contain the definition of the field which is to be interrogated (such as the p-Value field for expression in a microarray record) so that its index within the attached records can be determined, along with an operator, and a comparison value. Since the VLEs have been initialized with the definition of the field of interest, they contain knowledge of its data type: textual, numerical, or temporal. The field's format affects which operators are available for the VLE to use:

- Numerical: greater than, less than, equals
- Temporal: before after, equals, equals with tolerance
- Textual: contains, starts with, ends with, equals

8.3.1 Relationships between VLTs, VLEs, and values

Each type of Visualization Schema contains a different class of value associated with its conditions. As mentioned above, the Appearance Schema contains Material objects, affecting an Entity's appearance. The other schemas contain much different value types: the Behavior Schema contains Behavior objects, the Transform Schema contains Transforms, the Contextual Schema contains a delimited list of other required concept IDs, and the Instance Depiction Schema contains customizable mesh-alteration classes which perform more complex operations with methods defined by a standard Java interface.

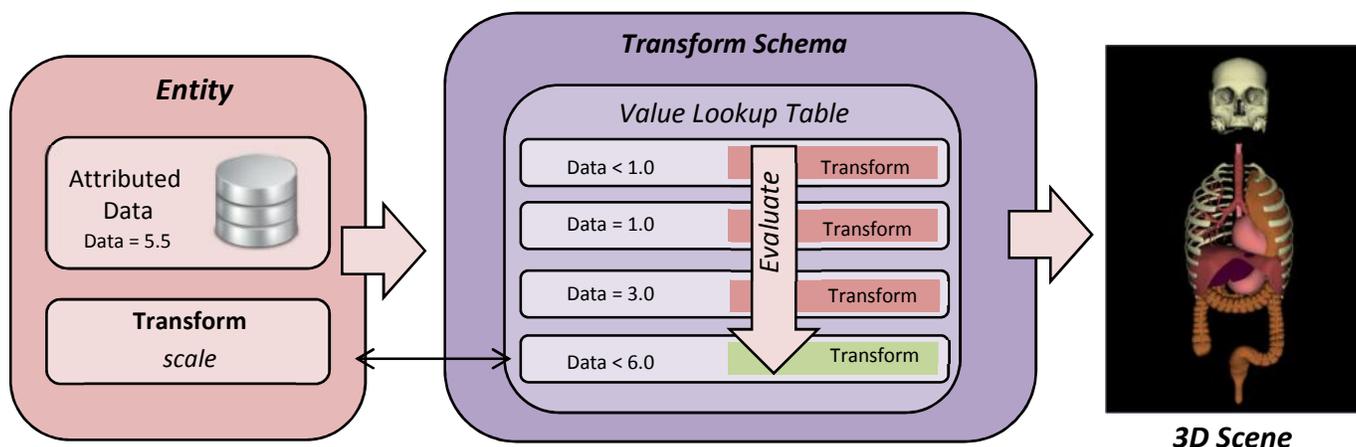


Figure 8c. Description of the VLT evaluation process within a visualization schema

The aforementioned value types allow a schema to perform a wholesale substitution of an original object (i.e. Material) for a new value object which can be composed of many different attributes, as is the case with Materials, Behaviors, and Transforms. However, these whole-object substitutions may not meet the needs of users and developers who seek to change only specific values within these objects, such as the scale with a Transform and the diffuse color within a Material. Moreover, one may desire to establish more of a continuous influence between the source data and the affected value. We may design a visualization where the red-channel value of a microarray record should become the red value of a Material's diffuse color (adjusted a bit to remain within the 0-1.0 range, of course). This capability would allow one to avoid constructing discreet objects for every possible value in the source data. VLEs which allow these operations are referred to as parametric VLEs.

8.4 Record resolution

An assumption which Biolucida makes regarding the relation between data and VLTs is that there is a single value to be evaluated against the conditions defined within the VLT. While this may be the case for the source data after it has been first attributed to an Entity (though, possibly not if there are multiple sources and/or Data Handlers), data propagation will lead to a potential accumulation of multiple records at higher levels of an ontology due to child nodes inheriting records from multiple parents. In order to simplify the use of the VLT in the visualization pipeline, the developer or user may specify a *record resolution* operation, which is responsible for producing the *single* value which will be evaluated. Currently, Biolucida offers these resolution operations: mean, median, minimum, maximum, most common, least common, and amalgamation. The amalgamation operation ensures that the VLE will evaluate all of the values, and if any evaluate to *true*, then the VLE is triggered to act upon the Entity. This type of operation, however, is incompatible with parametric VLEs.

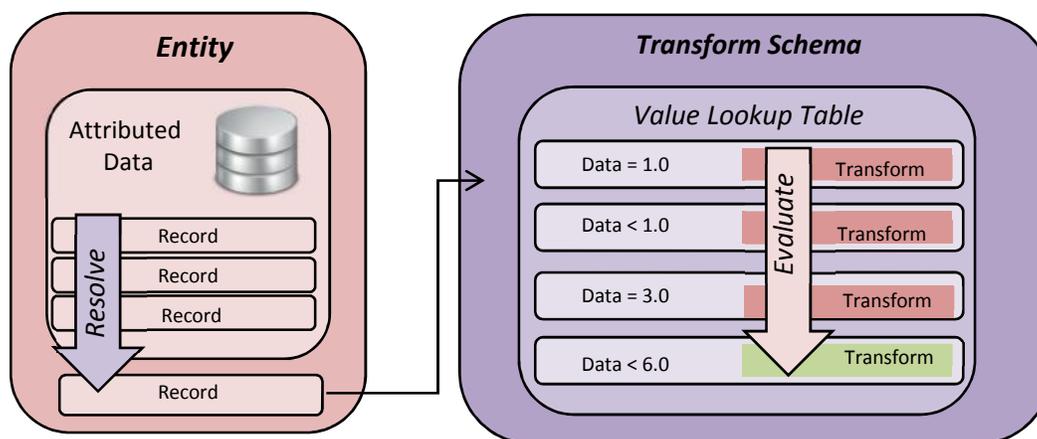


Figure 8d. Depiction of the record resolution operation

8.5 Customization of VLEs and VLTs

Like most of the systems which play a primary role in the look and feel of a visualization, the VLT systems are fully customizable. From a developer's perspective, the VLT system can be extended to include VLEs with enhanced processing capabilities which can allow the conditional evaluation of all records attributed to an Entity, interrogation of multiple fields, and maintenance of state between the evaluation of one Entity to the next. The developer is also able to extend the predefined VLT interfaces to create novel types of VLTs which might affect Entity attributes defined in future releases of Biolucida, such as a time vector associated with Entities in a scene.

Users may control the variables of the visualization through a series of core GUI panels which are designed to allow editing of the VLTs and their constituent VLE/value pairs. Users may select the field of interest within the incoming dataset, specify the resolution operation to use, and manually edit the contents of the VLTs embedded within the Appearance, Behavior, Transform, Contextual, and Instance Depiction Schemas. However, the developer of a particular representation may choose to constrain the user's ability to edit these settings, and instead embed a predefined set of VLT objects within the visualization schemas.

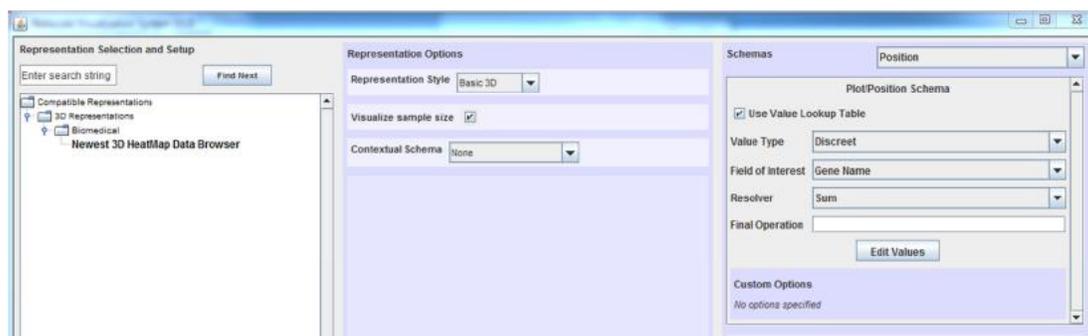


Figure 8e. User interface allowing the customization of VLT contents

8.6 Extending the functionality of the VLT system beyond simple data

As each Entity is processed prior to its insertion into the scene, the VLT classes have access to many of Biolucida's scene and data query mechanisms. Developers who extend the VLT system have the ability, as each Entity's attributed data is interrogated, to query the current contents of the scene, survey the concepts in the Biolucida catalog and their ontological relationships, and even query external data sources via the Data Handler interfaces. Such extensibility could possibly enable enhanced inference capabilities as the Biolucida system grows.

8.7 Persistence for visualization schemas

Biolucida has been designed not only as a way to produce informative visualizations across the biomedical research domain, but also as a means of producing and evaluating 3D visualization conventions. While two-dimensional data visualization has become a relatively mature discipline with well-known conventions and regular innovation through new representations, 3D data visualization is a much newer and less tested discipline.

The lack of conventions in the 3D visualization world may have much to do with the effort required to produce a 3D visualization scene. Studying visualization heuristics entails creating a large number of visualizations with the intent of comparing their effectiveness in describing the datasets they represent. With the aforementioned high knowledge requirement of working with low-level toolkits such as VTK, and the lack of higher-level frameworks, a researcher faces a daunting task when it comes to the production of any appreciable number of visualizations for such a study.

A number of features within Biolucida are designed to allow it to serve as a rapid scene generation system in order to produce a large number of visualizations for evaluation purposes. In this manner, Biolucida functions as a visualization research tool in addition to a Biomedical data representation and communication system. To further minimize the effort required in producing visualizations, the Biolucida system has been imbued with the ability to save and load visualization schema definitions, so that a researcher only needs to create a visualization definition for evaluation purposes once. This capability also allows the reuse of visualization parameters not only between sessions, but also between users and between different Representations, if allowed by the developer.

Chapter 9

Results and Evaluation

9.1 Introduction

In the previous chapters, we have described the software architecture of Biolucida and how the components within that architecture function together to carry out some of the core tasks in the process of building interactive 3D Representations, processing external datasets, and applying visual metaphors to the contents of a scene. In this chapter we describe how these core modules have been extended to implement four distinct visualization systems (as plugins) and address the needs presented in the use cases found in chapter three. We will also discuss the results of several evaluation exercises which were performed to determine the effectiveness and appeal of each plugin or application.

The use-cases described in chapter three were developed through review of literature describing prototype biomedical visualization systems to derive a set of prototypical scenarios where visualization tools are used. For review purposes, the use cases are summarized below.

1. **Anatomy education material production.** An anatomy instructor uses Biolucida to produce an interactive scene describing the structure and function of the cardiovascular system.
2. **Informed consent attainment.** A surgeon uses Biolucida to create and deliver portable animations showing the sequence of events in three separate surgical procedures to remove a tumor from the brain
3. **Molecular data visualization.** A researcher must produce a visualization of the expression location of a particular cell-surface receptor within the human body.
4. **Patient data visualization.** Developers desire to deploy a system which allows physicians to review a wide array of patient data within a single interactive visualization.

The development of add-on Biolucida modules to meet the needs of the above use-cases was undertaken as a proof-of-concept to explore how developers would extend the functions of the core system to create domain-specific applications. These use-cases were chosen not only because they are representative of the manner in which visualization systems tend to be applied in biomedicine, but also because together they demand the implementation of a very diverse set of features of which only the most capable of interactive visualization systems could implement.

9.2 Custom Biolucida client systems

Implementing the systems which provide the functionality described by the aforementioned use-cases required the creation of two custom clients and several custom Representations and data-centric plugins, just as was intended using Biolucida's current architecture.

9.2.1 The Anatomy Scene Authoring Client

The ASA client is the most basic implementation of a Biolucida client. The ASA client is focused on the needs of the educational and communication disciplines within biomedicine, supporting the efforts of educators and professionals who have an interest in developing interactive 3D scenes and animations.

Initialization operations. Upon logging into the Biolucida Server, the client sends a series of commands which set the conditions necessary to begin constructing a scene. First, the client requests the creation of a blank dataset at the server (a requirement for all Representations), and the resulting dataset ID is returned by the server which is cached by the client. Second, the client requests the creation of an ASA-related Representation based on the blank dataset ID. After the creation of the new Representation instance, its unique ID is sent back to the client, along with the X3D content of the blank scene for this type of Representation. The blank scene contains not only the default control and information nodes to allow for some basic functions such as determining the camera's position and orientation, it also contains a LayerSet node capable of hosting the models which compose a heads-up-display (HUD) control panel. The HUD contains a gyrosopic compass in the form of a human skeleton. The compass mirrors the orientation of the anatomical objects within the scene to prevent the viewer from becoming disoriented.

9.2.1.2 The Interactive Scene Generator (ISG) Representation – A full-featured intelligent scene generator and animation system

This Representation includes the basic features of the Biolucida core system responsible for building an anatomical scene, accessing its contents, and moving the camera. The ISG Representation does not implement any custom modules. It simply uses the modules found within the core client architecture.

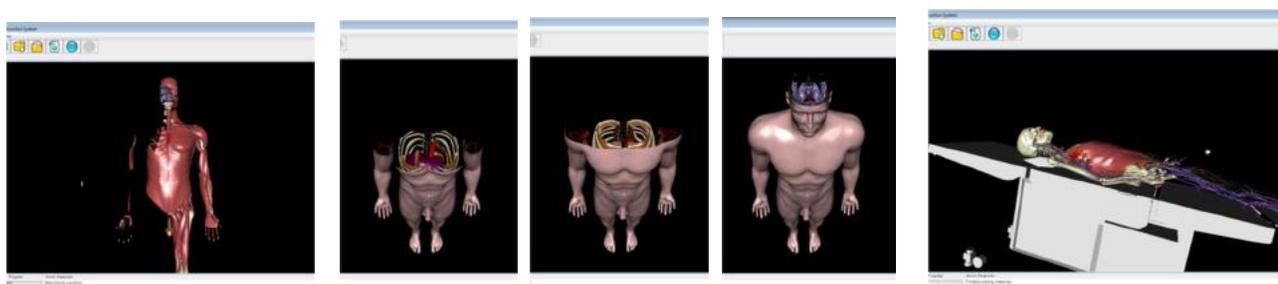


Figure 9a. The ASA Client, showing a basic scene, plane clipping, and contextual content

Implementation of the abstract 3D representation. For purposes of explaining the unique features of each Representation, it is best to describe each implementation in terms of the required areas of customization described in chapter 5. The ISG Representation implements mostly core functionality, drawing from the standard classes and controls made available to plugin developers, with slight modifications where appropriate.

User interface. The ISG implements a collection of default buttons, menu items, panels, and control assemblies within its user interface. These objects are included in the core Biolucida Client and require no alterations to provide their prescribed functionality. These objects include buttons to log in/out, save/load scenes, create scenes, as well as control panels and graphical information displays. In addition to the inclusion of these core GUI elements, the ISG includes a few custom displays which provide the user with additional features, such as an Entity appearance editor, and an Entity selection display.

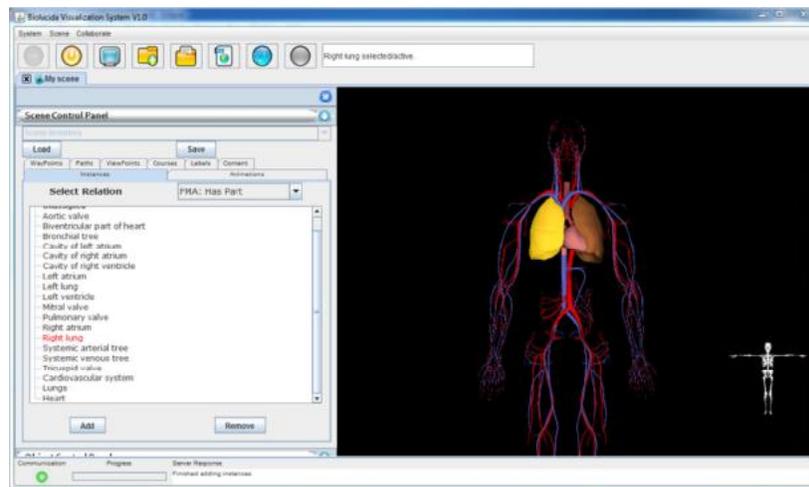


Figure 9b. The ISG Representations displaying the cardiovascular system.

Event logic. The ISG Representation implements three custom event processing “controllers”, which are responsible for camera manipulation, Entity selection processing, and scene editing.

- Camera View Controller (CVC). The CVC is responsible for manipulating the camera into proper position and orientation for canonical views of selected Entities. Responding to presses on the numerical keys 1-6, the controller moves the camera to the front, rear, top, bottom, right, and left views of the currently selected Entity. Proper distance between the camera and the centroid of the selected Entity is calculated using the Entity’s bounding box data, stored on the Biolucida Server.
- Pin Placement Controller (PPC). The PPC is responsible for placing labels, which show an Entity’s concept name, on the Entities within the scene. The labels “float” in space, occupying positions beyond the periphery of the bounding box of an amalgamation of all models contained within the Biolucida catalog. Each label is connected to its respective anatomical structure by a single line, or pin, running from the centroid of the 3D model to the base of the label. Placement of each label is established using the bounding box of all possible models describing the human body. The label is placed slightly beyond the periphery of the bounding box using a calculation which sets the amount of offset left/right and up/down based on where the centroid of the model is located compared with the maximum extent of the x and y coordinates of the body’s bounding box.

- Entity Placement Controller (EPC). The EPC allows the user to move and rotate Entities within the scene when an alteration of the position and orientation from the default value is desired. The EPC is activated using the keys “g” and “r” to grab (move) and rotate an Entity, respectively. When activated, this controller registers itself as a listener of mouse motion events, followed by the establishment of a connection with the set_translation and set_rotation fields of the Entity’s (scene) transform node. The controller updates the Entity’s position and rotation by translating the motion of the mouse, attenuated for the camera’s distance from the Entity, into successive updates to the transform’s translation and rotation fields. Changes in position and rotation can be constrained by a single axis by pressing the x, y, and z keys when the EPC is active. Axis proxies in the form of primitive shapes are added to the scene, centered at the Entity’s centroid, as visual cues to assist the user.
- Entity Identification Controller (EIC). The EIC receives Entity selection events from the Entity interaction event manager, extracts the concept name from the event object, and updates the text content of the Entity Selection Display, allowing the user to see the name of the Entity that they have selected.

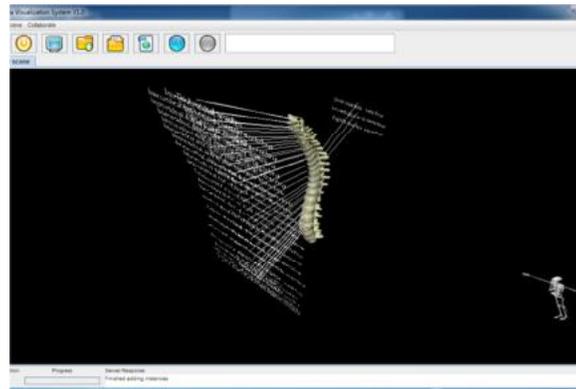


Figure 9c. Biolucida scene showing pin placement

The default Semantic Level of Detail controller. The core Biolucida client includes a default Semantic LOD controller which is responsible for responding to keystrokes that cause the focus of the client, in terms of the currently selected Entity, to move up or down along the chosen relationship representing Semantic Level of Detail. This capability allows the user to manipulate Entities in groups of varying level of granularity to achieve a desired look and feel of the scene.

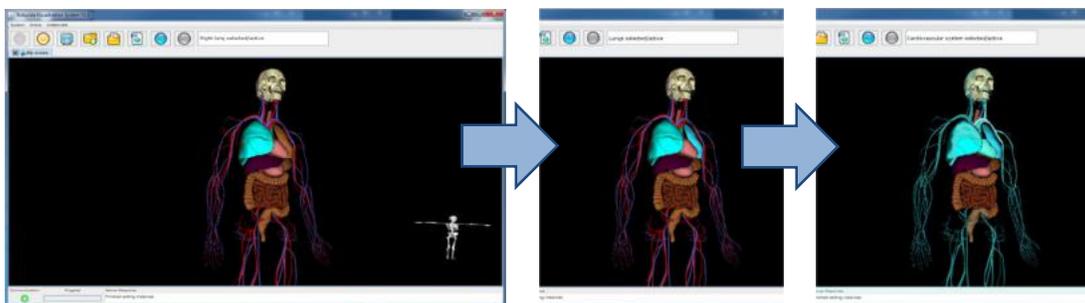


Figure 9d. Transitions between semantic levels of detail in the ASA client. The screen captures show movement from Right Lung, to Lungs, to Cardiovascular System

Evaluation – Task-based user evaluation to test usability and suitability for users who lack 3D software experience. While the ISG may be functional from an alpha testing perspective, it requires evaluation by users to establish a target audience’s potential satisfaction with its use. Biolucida was designed to be used by researchers, clinicians, and educators who have little training in 3D design and animation. The software design strategy behind the ISG’s development is to create an interactive scene builder which is easy to learn and requires no background in the 3D design software packages which have saturated their respective markets for the past 10 years. Solutions such as Inventor, Maya, 3DS Max, Solidworks, and Catia, which are found in the engineering, manufacturing, and architectural fields, require a great deal of training to utilize effectively.

The use of professional 3D design systems requires specialized software skills and equipment [93], the cost of which (both in time and funding) is often beyond the level of investment to which physicians and researchers will commit. Expert personnel tend to be available only to project teams at large institutions and/or to groups who are well-funded. Even in cases where such resources are available, commercial systems tend to be designed with the financial potential of the market, rather than innovation, in mind. Licensing restrictions, limited access to APIs and libraries, and a superfluous catalog of capabilities designed to meet the needs of those with the deepest pockets are among the many issues with using a commercial system for research purposes. On the other hand, one can always turn to the variety of open systems which the academic and progressive software development worlds have produced. However, these systems can require a larger time investment since their documentation and training materials are often inadequate or hard to come by, and learning by trial-and-error is difficult because these systems are highly complex because of their large feature sets, such as in the case of Blender.

The goal of producing a system which has a low barrier to entry and can be operated by users with little or no graphics training, calls for a usability assessment by two groups which differed markedly in their level of training in 3D design systems. Two groups of ten users each (randomly chosen from 20 candidates) were selected for this study. The first group was composed of subjects who were actively employed in a technical design profession which requires the use of 3D software. The second group was composed of professionals from the allied health field, specifically massage practitioners. The second group was selected not only because of a confirmed lack of

exposure to 3D design software, but also because a familiarity with anatomy provides each user the domain knowledge to utilize the software to complete the assigned tasks building anatomical scenes.

The subjects were each given a brief orientation in the use of the software, and subsequently asked to perform a series of tasks designed to test the breadth of the ISG's capabilities. The subject's screen activity was recorded and each user was asked to "think aloud" while executing each task, providing a cursory account of the subject's cognition during their activities. After the completion of all requested tasks, the subjects were asked to fill out a System Usability Survey [15] form, containing ten Likert-Scale items design to assess a subject's satisfaction with the software system. A copy of the task list and the questionnaire can be found in Appendix D, and the results of the evaluation can be found in Appendix E.

The user evaluations revealed that the Biolucida system offers a great deal of promise as a means of delivering 3D communication and analysis applications to novice 3D users. Analysis of the SUS item scores (Appendix E) revealed that the Biolucida prototype had a mean usability rating of 71.6, when compiling the scores of all users. While this score is only slightly above the average score of 70 for a large sample of SUS evaluations [5], it is promising for a prototype of Biolucida's complexity and novelty. There were slight differences between the two survey groups. However, these differences were not statistically significant.

A more qualitative assessment of notes taken during the evaluation, taken from the *think aloud* activities revealed several interesting insights. It seems difficulties arose mostly from the users' understanding of the concept terms used to identify models to be placed in the scene and to be manipulated in the required animations. These difficulties were more prevalent than those arising from any unfamiliarity with Biolucida's user interface design. Another prevalent difficulty which all users encountered was with the various methods of navigation. Users were unfamiliar with terms such as "examine" and "tilt", in reference to manipulation of a 3D camera.

9.2.1.3 The Procedural Animation (PA) Representation – A full-featured intelligent scene generator and animation system

The second use case described in chapter 3 refers to a surgeon constructing several animations of surgical procedures, and creating a distributable artifact that can be viewed without the use of the original authoring software. To meet the needs described in this use case, the PA Representation was created. The PA Representation is capable of constructing a scene using the same functionality as the ISG Representation, with three additional capabilities. The PA Representation includes an animation construction module which allows the user to create dynamic content within a scene, constructed by stringing together the atomic actions outlined in Chapter 6. The PA Representation also allows an author to import external 3D content into the scene beyond that described within Biolucida's concept catalog, such as surgical instruments and equipment. Lastly, the PA Representation has the ability to "publish" a scene, complete with all of its constituent animations, into a standard VRML/X3D file which can be read by any browser meeting the appropriate Web3D conformance levels.

Event logic. The PA Representation contains the same event processing controllers that are found within the ISG Representation, plus an additional controller to handle the insertion of an “instrument” within the scene, designed to represent a surgical tool or other implement which the author can use as a pointing or “picking” device, much like a mouse cursor. The design intent here is to provide the user with an illustrative prop with which they can better explain the procedure. The controller is activated by a single key command, at which point it inserts a selected model, representing the illustrative instrument, into the scene. The controller then registers itself as a listener of mouse motion events and moves the instrument in response to the user’s movements.

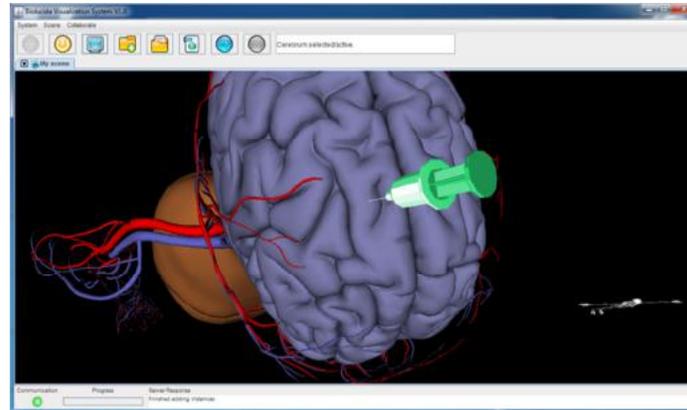


Figure 9e. The PA Representation using a syringe as the instrument pointer.

User interface. The PA Representation includes a few additional user interface panels in its ancillary options area, all implemented within Biolucida’s core set of user interface modules. The animation editor manages the animations, sequences, and actions which compose the dynamic content of a scene. There are also a series of inventory panels which are responsible for the collection of viewpoints, waypoints, paths, and courses within the scene. These objects serve as points and collections of points which can define motion for the camera or Entities over the course of an animation.

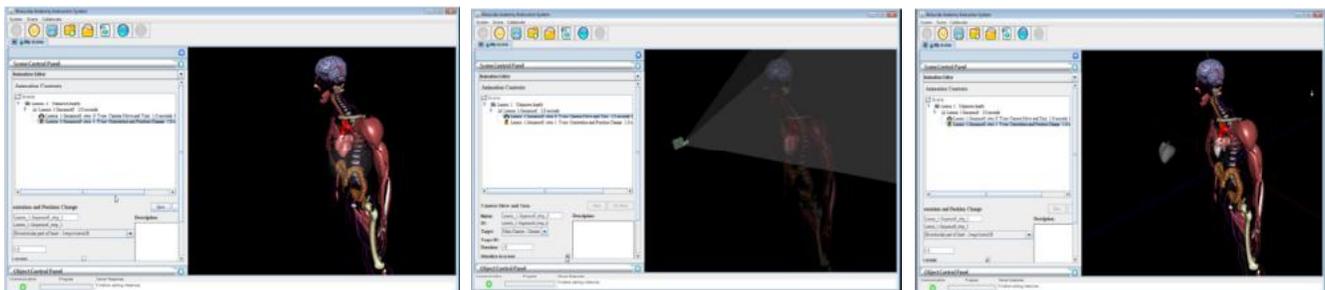


Figure 9f. PA user interfaces allowing the construction of animations by managing discreet actions.

File output. The export of a VRML/X3D file is made relatively simple due to the fact that the same content factory used to translate Entities into the appropriate syntax for rendering within the Xj3D browser window is used to write

the entire scene to a file. However, there is a modification made to the scene before such a final translation occurs. Entities transmitted between the server and the client generally rely on *inline* shapes, where the geometry is referenced by URL, but the exported VRML/X3D file must be self-contained to be an effective communication token. Therefore, all geometry information is first inserted into the Shape object of each Entity before the translation is completed and the resulting file is downloaded. If animations are present, then an interactive panel is inserted into the HUD display of the scene containing selection controls and a “start” button to begin the animation.

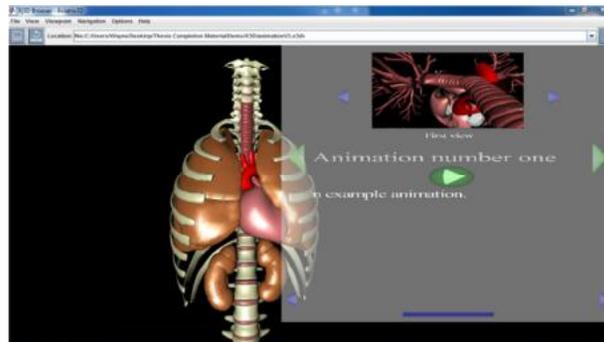


Figure 9g. Output X3D File with semi-transparent animation selection panel.

Evaluation – Heuristic evaluation to establish the usability of more complex interfaces. The PA Representation uses much of the same user interaction techniques implemented by the ISV Representation. Therefore, a user acceptance study would seem repetitive. It has been found that not only are expert reviews of software interfaces effective, but also that the free-form discussion they elicit can offer insights which fail to arise from tightly controlled task-focused user studies, especially during exploratory phases of development [130]. Five experts will typically find about 75% of the usability shortcomings in a given system [112]. Heuristic Evaluation [98], originally developed by Jakob Nielsen, is designed to evaluate the design of interfaces based on a set of general interface heuristics which are well known by software usability. In keeping with these guidelines, five usability experts were recruited to perform a heuristic evaluation on Biolucida’s animation authoring interfaces, in hopes of identifying areas where the interfaces could be troublesome or confusing. Because of the complex nature of the animation construction user interface system, it seemed most informative to focus an evaluation on the ease of use of those particular panels, especially since their design departs from standard 3D animation software conventions.

Many commercial and Open Source 3D animation software packages use a keyframe model of animation, where the author makes cumulative changes to the entire 3D scene, specifying the point at which the changes are to be completed along a timeline. The animation software then interpolates between the original and the altered scene, creating dynamic content which embodies the transitions between the states of the content of the keyframes. Biolucida uses a similar concept, but attempts to simplify animation authoring by limiting the changes to the discrete actions which affect only singular targets.

The heuristic evaluation exercise utilized a team of five individuals currently employed as usability experts at a local industrial design firm which receives a significant portion of its yearly revenue from contracts involving the design of user interfaces to be implemented in a variety of software systems and consumer electronic devices. The participants were given a brief demonstration covering the features of Biolucida which allow a user to create custom scenes and to publish animations. After the demonstration, a discussion ensued in which the following were highlighted as areas of improvement in Biolucida's user interface design.

1. **Lack of consistency in control “widget” appearances.** One item mentioned several times by the expert panel focused on the consistency of the look and feel of the standard controls in the various panels and menu bars throughout the system. Some buttons have only text, while others have an inconsistent iconography applied to them.
2. **No task-oriented linear flow reflected in control panel organization.** The expert panel revealed that there exists no grouping of UI elements in Biolucida which support the main categories of tasks which the user performs, such as “populate scene”, “customize scene”, and “publish”. It was suggested that users will expect control objects which are logically related to be physically grouped together, and that these groups should exhibit a linear flow to accomplish the task for which the system was designed. For instance, there exists no “add model” within the scene inventory panel, even though the two elements are intimately related.
3. **High level UI elements are too generic to be important.** Biolucida's main toolbar currently contains controls responsible for signing in/out, saving/loading a scene, etc.. The panel felt that these functions have little relation to the main tasks performed by the user, and should therefore be demoted while other UI elements should occupy this top-level toolbar.
4. **Generic terms do not yield an understanding of tasks performed.** In an effort to help maintain Biolucida's generalizability, UI elements were designed with descriptive text which does not commit the system's semantics to any one application. For example, in order to add an anatomical model to a scene, the user selects a button entitled, “Add Scene Instance”. The panel thought these and many other terms should be re-written to be more domain-specific and therefore, more intuitive.
5. **No “expert” mode.** While many of the recommendations of the group were based on a novice user's expectations, it was also mentioned that expert users will have different preferences and should therefore be accommodated by employing a dual-mode design. Such an “expert” mode may rely more on key strokes and scripting commands rather than UI elements, reserving more screen area for the scene versus the controls.
6. **Navigation should employ buttons.** The expert panel commented that novice users will most likely fail to understand the navigational terms used in the Biolucida interface, such as “fly”, “pan”, and “examine”. It was also mentioned that iconography should be employed to promote understanding.

Interestingly, it was not the animation construction interfaces which drew the experts' attention, but these more general concerns. Overall, the expert panel was pleased with the novel nature and functionality of the system,

and encouraging discussion revealed that Biolucida currently appears to be, from a look-and-feel perspective, not far from the quality of many early-stage commercial systems which the firm has encountered.

9.2.2 The DataGen Client (DGC)

When Biolucida is used to create data-driven visualizations, the development focus moves from including interfaces and enhanced user-interaction event logic to the implementation of Data Handlers, Data Attributors, Propagators, and Visualization Schemas. To support the functionality required to meet the needs of the researcher desiring to visualize the site of expression for a particular cell surface receptor, the SDV representation and a collection of data-related plugin objects were created.

9.2.2.1 Server-side components

The DGC is designed to issue commands which initiate a number of server-side processes that are responsible for building datasets, attributing records to anatomical concepts, and instantiating visualization schemas. The user controls this process through a set of user interfaces which are presented prior to the visualization of a 3D scene, unlike in the workflow established in the ASA client.

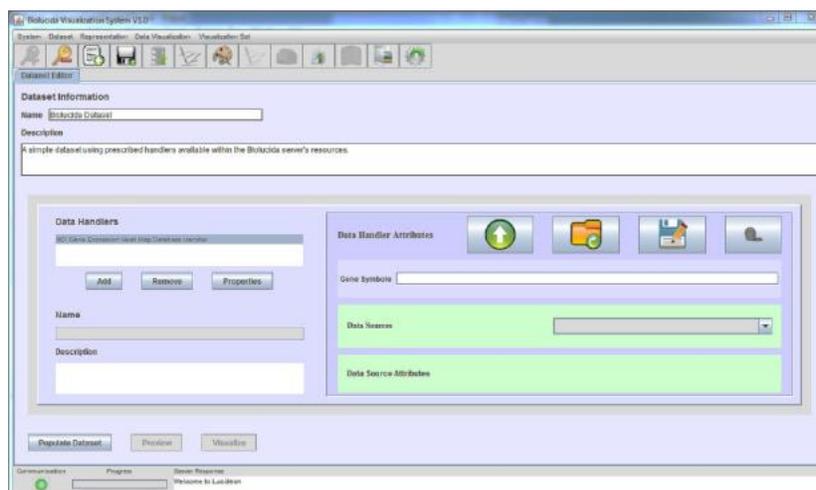


Figure 9h. The Datagen Client interface which allows an author to set the parameters for the extraction of a dataset which forms the basis for a subsequently produced visualization.

Data Handlers. Three data handlers were developed to acquire and process gene expression data which would subsequently drive the visualization process. Expression data can exist in a number of formats, and reside in different places. In order to demonstrate Biolucida's breadth of features which it offers the developer, the data handlers were responsible for processing of three unique data sources, accessed via local file parsing, HTTP connection, and by means of an API.

HuGE Index Data Handler. The HugeIndex project [48] contains the expression profiles for thousands of genes across twenty anatomical structures, and more are currently being added. Expression data was manually acquired using the HuGEIndex.org website’s query interface and unified into a single .csv file for subsequent parsing by the data handler. The HuGE Index dataset contains an arbitrary score reflecting the level of expression of each gene ranging from 0 to roughly 10,000. The expression data can be used to compare samples and genes, since the same consistent procedure is used to translate expression levels to this “score” in every case.

Array Express Data Handler. As opposed to other gene expression data sources, ArrayExpress [107] implements a much more rigid set of standards for its experimental datasets. MAGE-ML [89], or Microarray Gene Expression Markup language which was originated by the MGED working group [146], is a formal standard describing microarray results. It is the product of fusing a gene expression-specific schema, or document type definition, with MAGE-OM, an object model describing microarray experimental data. Databases utilizing this representation method require less file-parsing work from their users, as Java classes for the extraction of information are openly available. Several MAGE-ML files which describe gene expression in normal human tissues were downloaded to the data directory of the Biolucida server for subsequent parsing.

NCI Gene Expression Handler. The NCI Gene Expression data handler is designed to handle a dataset composed of a network-accessible database containing the data described by NCI’s normal tissue gene expression database. The dataset contains the expression profiles of 18,927 genes, taken from 158 tissue samples, and therefore contain references to anatomical concepts, used for data attribution. The data were downloaded from the NCI website and inserted into a csv file which was subsequently processed by a simple set of Java methods which inserted the data into the awaiting tables, hosted on a PostgreSQL database.

| Dataset | Gene Expression Data Type | Anatomical Structure Reference |
|------------------------|--|---|
| HuGE Index DB | Arbitrary score from 1 to 10,000 | Common tissue terms such as “stomach”, and “brain” |
| Array Express MAGE-ML | Expression flag of “up” or “down” with a p-Value | Unrestricted term set which can be of arbitrary origin, decided by researcher; can be omitted |
| NCI Gene Expression DB | Heatmap score of -2.0 to 2.0, higher score means higher expression level | A constrained set of twenty five terms, much like the HuGE Index anatomical references |

Table 9a. Descriptions of datasources used for the Biolucida prototype of the DGC client

9.2.2.2 Compatible data visualization Representations

Data visualization Representations offer limited authoring capabilities and focus more on simply allowing the user to explore the scene and investigate its contents after it has been delivered to the client. Each Representation is described by its RepresentationDescription class, identifying the necessary fields which are required to be in the

source dataset in order for the Representation to have access to the data it needs to fully execute its visualization schemas. If a Representation is compatible with the currently loaded dataset, then it is offered to the user as a potential selection for visualizing the dataset after it has been parsed and loaded into the system.

9.2.2.2.1 The Simple DataViz (SDV) Representation

The SDV Representation has the most limited set of implemented core features of any representation. Selecting Entities and requesting the expansion of the ancillary options panels will produce no results. Therefore, the SDV presents a scene to the user, which is designed only for exploration. Though the user still has the ability to switch navigation modes, the scene defaults to the “examine” mode. The SDV representation was produced to be used as a means of testing the effectiveness of visualization schemas, where the scene, and the appearance, spatial characteristics, and behavior of its contents are to serve as sufficient representations of the underlying data, without the help of Biolucida’s other interactive displays.

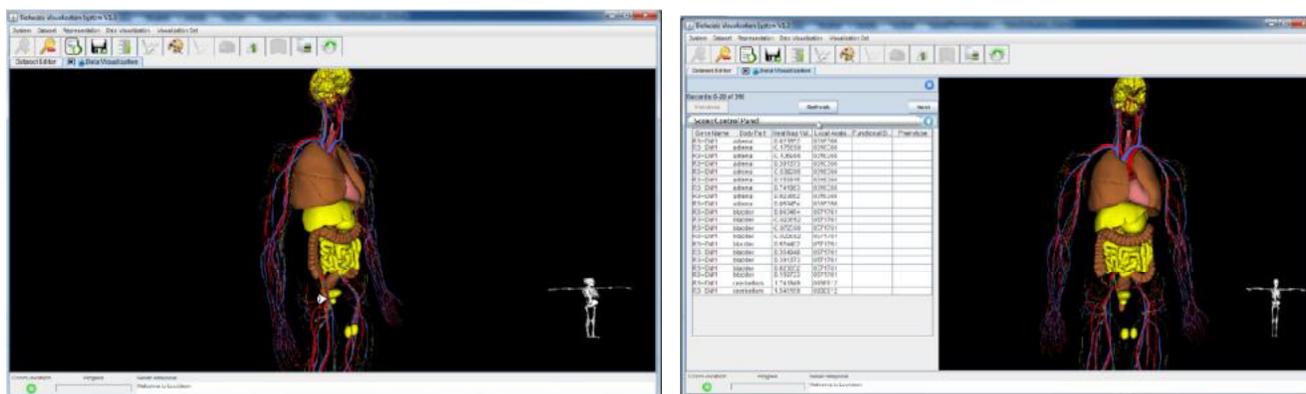


Figure 9i. The SDV and DBV Representations showing positive gene expression for a given queried gene name.

9.2.2.2.2 The Data Browsing Visualization (DBV) Representation

The DBV Representation implements more functionality than the SDV Representation in that it offers the user a data browsing panel within the ancillary option panel array. This UI feature allows the user to select an Entity, and view the source data attributed to that Entity, as well as review all records within the dataset. The table display applies text formatting to differentiate propagated and transformed values from those derived from the source data.

Evaluation – Focus group discussion of the effectiveness of the data visualization Representations, and of the visualization schemas. A focus group was selected from members of the Structural Informatics Group, a University of Washington informatics research team which develops applications which manage, query, and visualize information in an effort to meet the needs of the Biomedical research community. A demonstration was given using random datasets and the representations described above, showing Biolucida’s visualization workflow and the use of the interactive visualizations which are produced as a result.

Discussion of Biolucida’s capabilities yielded generally positive feedback, and several key areas were highlighted as potential points of improvement slated for future development. These suggestions included:

1. **Better navigation controls.** The focus group felt that the navigation controls within Biolucida were unintuitive and that the system was missing certain camera controls which would be most helpful for novice users. These controls include a “fit to scene” command, where a user could establish a position and orientation where the view frustum included all models within the scene, and a “straighten up” command which would position the camera upright, yet looking in the basic direction of the original view.
2. **Leveraging external content via URL-based objects.** Even though Biolucida has the capability of including a wide variety of content in its current prototype state, it should allow the referencing of external information sources via URL. Such sources can point to narrative information which is relevant to the imported data and to the anatomical models within the scene, or such information may leverage the full functionality of web services, semantic technologies, and other information sources offered by the Web.
3. **Simpler client initialization process.** There was consensus among the group that a simpler method of starting the client must be made available to the user. Launch methods such as JNLP or alternative deployments using browser-based technologies seemed preferable to the current standalone Java prototype.

After general feedback on the Biolucida system was collected, the focus group was shown a series of visualizations using different Visualization Schemas in an effort to explore effective means of presenting 3D data visualizations. The schemas used differed slightly in the manner in which gene expression data were illustrated. A random data generator, implemented as a custom Data Handler, was used to populate a dataset containing the same fields as the NCI gene expression dataset. The visualization schemas were designed to interrogate the gene expression flag (true or false) of each record, and when appropriate, use the p-value of the expression finding for values. In each case, anatomical structures which contained no data or a p-Value of higher than 0.10 were colored black.

- **Color-based visualization schemas.** Color-based visualization schemas were implemented to apply a red shade to the appropriate color field when the expression was found to be negative, and green when the expression was found to be positive. The p-Value was used to modulate the intensity of the color.
 - **Diffuse schema.** Appropriate color was applied to the Entity Material’s diffuse color field
 - **Diffuse with transparency schema.** Appropriate color was applied to the Entity Material’s diffuse color field, while the p-Value was used to modulate the transparency.
 - **Emissive schema.** Appropriate color was applied to the Entity Material’s emissive color field
 - **Specular schema.** Appropriate color was applied to the Entity Material’s specular color field.

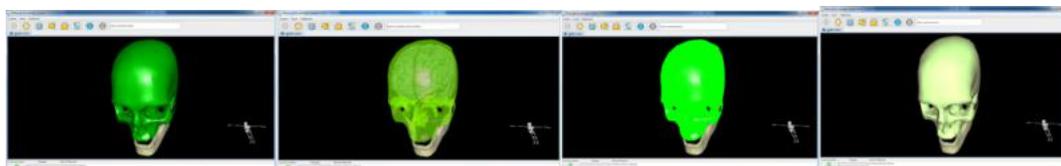


Figure 9j. Screen captures of four of the schemas described above, depicting positive gene expression. From left to right: diffuse color, diffuse color with transparency, emissive color, and specular color.

After viewing the aforementioned visualization schemas, the focus group found diffuse color to be the most informative retinal attribute used as a visual metaphor for data representation. The other appearance attributes exhibited shortcomings when used in visualizations, which the focus group highlighted in their discussion:

- Transparency proved to be a useful visual metaphor when applied to models with relatively simple shapes, such as the femur. However, interpretation of its meaning was difficult when applied to more complex shapes, where a single model may exhibit undulations which require the viewer to peer through more than three layers of transparency within the same entity, as in the case of the small and large intestine models.
- Specular and emissive colors were less useful than diffuse colors because specular highlights appeared very weak, while emissive colors clearly overpowered other visual attributes
- Color gradients were useful only when composed of appropriately spaced discreet levels, since focus group users had trouble identifying differences in levels of red and green when the displayed color ranges were divided into more than four levels. This comparison was especially difficult for focus group members when performed in sparsely populated scenes and applied to entities separated by larger distances in space.

Unfortunately, the following additional schemas were prepared for viewing and subsequent discussion by the focus group, but time limitations did not permit their inclusion. These visualization schemas, therefore, will be most likely used in future work:

- **Behavior-based visualization schemas.** Entities were given behaviors which represented whether the gene expression was found to be positive or negative. Entities with positive values were given a pulsating behavior, and entities with negative expression values were given a rotating behavior. P-value was used to control the speed of the behavior.
- **Hybrid behavior/color/transparency schema.** Entities were given the same coloration as in the case of the diffuse schema, but the positives which displayed a p-Value of below 0.03 were given a pulsating behavior, and transparency was linked to the number of records attributed to the Entity.

9.2.2.2.3 Display of heterogeneous media in a single environment: The Medical Record Visualization (MRV)

The MRV is a Representation which brings together all of the elements of the previously presented data visualizations. It also incorporates more data types into the scene and provides some additional functionality to meet the needs of the clinician described in the fourth, and last, use case scenario. The MRV is composed of a data handler which imports patient data to be visualized using a reduced set of atlas models, and a random medical image

data selector which supplies a range of image types to supplement the content of the data handler, thus mimicking the breadth of content found in a real medical record that might be reviewed by a clinician.

The Practice Fusion Data Handler. In November of 2010, Practice Fusion (an electronic medical record provider) released a dataset composed of de-identified demographics, diagnoses, prescriptions, allergy data, and chart information for use by researchers in a number of medical fields. As a publicly available dataset, it can be downloaded from Microsoft's data market website. The data handler is responsible for randomly choosing a particular patient id, and extracting the necessary information described in table 9b associated with that patient.

| Field | Data Type | Description |
|--------------|-----------|---|
| ID | Text | The random id of the patient |
| Age | Integer | The age of the patient, calculated from their birth year to the time when the visualization is produced |
| Gender | Text | The gender of the patient |
| Systolic | Float | Systolic blood pressure reading |
| Diastolic | Float | Diastolic blood pressure reading |
| BMI | Float | Body mass index of the patient |
| Temperature | Float | Body temperature of the patient, in degrees Fahrenheit |
| ICD-9 | Text | The ICD-9 code for the physician's diagnosis |
| Diagnosis | Text | The physician's diagnosis, in free text form |
| Prescription | Text | The prescription assigned, if any |

Table 9b. Data extracted by the Practice Fusion Data Handler

Randomized Medical Image (RMI) Handler. A patient's medical record will undoubtedly contain a number of types of media besides the text-based data described in table 9b. These data can include voxel-based DICOM images such as CT and MRI scans, auditory data, movies such as ultrasound footage, and flat images such as X-ray films. A visualization system capable of presenting such information to a user must support the management and rendering of all of these types of data. Fortunately, the Xj3D browser embedded within Biolucida offers such support.



Figure 9k. The MRV Representation.

The RMI Handler simulates an association between a set of medical images and the Practice Fusion dataset by randomly selecting a DICOM image to display alongside the atlas visualization of the patient data. This style of presentation serves mainly to demonstrate the capability of display radiological image data within the same scene as the atlas-base visualization. However, there are a variety of ways given the capabilities of the SAI and Biolucida's event handling logic, of displaying 3D medical imagery within a visualization. This is especially true when one considers that voxel data can be translated to surface-based models by a number of established algorithms and third-party utilities. Figure 91 displays 3D medical image data display strategies within previous implementations of Biolucida Representations.

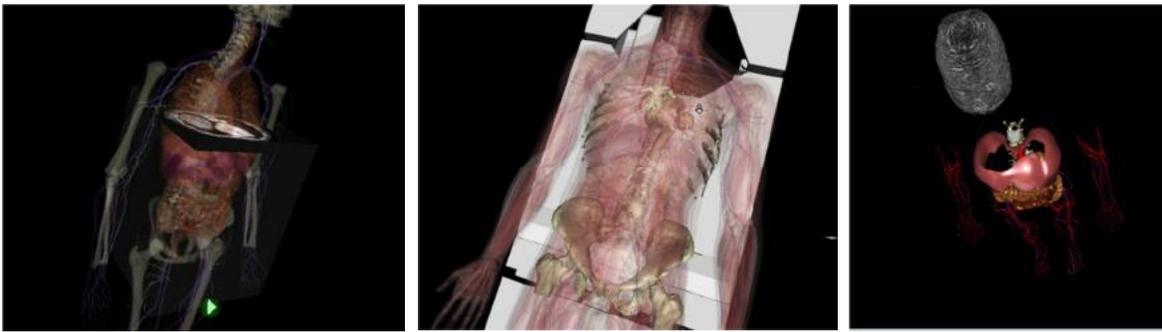


Figure 91. Multiple ways of displaying DICOM images within Biolucida: as serial slices displayed within the volume (left), as models from a 3rd party process through isosurface generation (center), and as an image stack with transparency (right).

Chapter 10

Discussion and Future Work

10.1 Revisiting biolucida's purpose

Biolucida was designed with the intent of serving as a visualization system to meet the broad needs of the Biomedical research community. It was constructed to provide a toolset for producing interactive content used to enhance the communication and comprehension of experimental datasets, clinical skills and procedures, anatomical knowledge, and many other types of information.

The intention behind the creation of such a broadly targeted technology is that it should provide both domain experts and external developers with a toolkit that would form the foundations for an innumerable number of derivative applications. These core systems provide necessary common functionality to these derivative visualization applications, such as: a communication framework, user interface tools, rendering systems, scene manipulation utilities, data processing pipelines, and visualization workflow management. With these core assets, developers or domain experts need invest only a small fraction of the amount of time it would take to deploy entirely new applications.

10.2 A review of the prototype applications developed as plugins to the core Biolucida system

The plugin modules described in chapter 9 demonstrate Biolucida's extensibility as a visualization application platform. Each derivative application was evaluated in a different manner to determine its effectiveness and to yield different insights which will inform future development efforts. Additionally, the applications provided opportunities to pursue related research, such as evaluating the effectiveness of visualization methods in 3D scenes.

10.2.1 Discussion of the evaluation results

The three evaluation exercises designed to examine different aspects of Biolucida (user-based, heuristic, and focus group) produced not only common insights which will inform the direction of future work, but also helped to further describe the strengths inherent in and challenges faced by the current prototype that is offered as a broad-use 3D biomedical 3D visualization system.

All three evaluation exercises highlighted 3D navigation as an area of desired improvement in the current prototype. This finding is not surprising given that proper implementation of navigation techniques is a well-documented challenge in 3D systems. However, novel methods for representing the modes of navigation, whether through iconography or animation, must be properly investigated if this challenge is to be overcome. It will be more revealing to evaluate Biolucida using other display technologies (such as head-mounted displays, and CAVE systems) to ascertain whether users have the same challenges when using more 3D-oriented equipment.

The user evaluation was designed not only as a validation of the basic interfaces provided within the core client, but also as a comparison of two user groups – one with substantial familiarity with 3D design software, and one with only knowledge of the relevant domain content, or more specifically, anatomy. The two user groups were chosen to be representative of two types of stake holders. The allied health subjects represented the graphics naïve researchers, educators, and clinicians who would use Biolucida to produce interactive content to help to communicate their own concepts and ideas. The 3D designers represented animators and 3D modeling professionals who might be brought into a project to produce these artifacts. The logic behind comparing the evaluation results of the two groups is that it might provide insights about whether or not graphics naïve domain experts can perform 3D design and animation just as well as a 3D content professional, if given the same amount of training on a system like Biolucida.

While the mean usability score among the allied health practitioners was higher (74) than the 3D designers (69.3), the difference was not statistically significant and therefore it suggests that both groups were equally satisfied with their experience in using the prototype system. This finding gives us a faint picture of the level of comfort each group experienced while using the software, but it does not speak to their levels of competence in executing the required tasks. The time of completion of the task list is a much better measure of such aptitude. A comparison of the two user groups yielded a slight difference in completion times of the set of requested tasks, an average of 27.5 minutes for the 3D designers and 24.2 minutes for the massage practitioners. Like the SUS scores, these differences were not statistically significant, so one could say that each group performed the set of tasks with similar speed.

The above findings suggest that users who lack familiarity with 3D software are likely to be able to operate the Biolucida prototype just as well as if they had been exposed to similar software systems already available. This finding is bolstered by a more in-depth analysis of the difficulties encountered by the users as they operated the software. Reviewing the narrative elements produced by the *think aloud* activities revealed that a majority of the points at which the users paused or encountered difficulties arose from an unfamiliarity with the anatomical concepts used to describe the models in Biolucida's inventory. The allied health practitioners had fewer troubles with such semantics than the 3D software experts. This suggests that a familiarity with the application domain may be more important than a familiarity with the operation of the visualization software system itself. This fact can form an argument against the compulsory procurement of expert software services when 3D communication and analysis is required.

A larger question to ask here is whether Biolucida and its plugins can be implemented and used effectively by researchers, educators, and clinicians. Given these users' limited time and resources, the learning curve of the Biolucida system must not be steep, and the time to develop plugins must be minimal. Otherwise, we will have created no viable alternative to the substantial investment of becoming familiar with the very complex commercial packages used for producing animations, nor will we have reduced the tendency of developers to engineer solutions to biomedicine's visualization problems from the ground up.

Biolucida was designed to approach a very daunting task – to serve as a generalizable 3D visualization system for the Biomedical field. This challenge produced a wide variety of use-cases and their related requirements. However, in an effort to gain an understanding of whether the Biolucida’s feature set is broad enough to support the application needs of future informaticists, the focus group described in chapter nine was asked to list the types of applications to which they would most likely apply Biolucida’s capabilities. It was thought that this type of exercise may yield a forward-looking view of tomorrow’s development community needs, versus the literature search described in chapter 3 which serves as a historical account of what biomedicine has required of its 3D software tools. The following types of applications were described by the members of the focus group.

1. **Medical education.** Anatomy instruction was an obvious domain of application for the Biolucida system. The group focused on the value, however, of customizing the content found in the catalog of 3D models, as no system will ever offer a totally complete catalog of models for all possible situations.
2. **Radiological aids.** The group discussed the value of using a 3D system to serve as an atlas aiding radiologists and their students in achieving proper orientation and to help identify anatomical structures.
3. **Spatial reasoning.** Visualizations such as those involving mapping activation areas of the brain were discussed. It was mentioned by one participant that an advantage which Biolucida offers in this area of application, again, is its ability to use any model set required – such as different “average” brain models for use in comparing data from multiple patients. A key feature discussed, which Biolucida has not yet implemented, is modeling time in its Representations. Such a feature would allow Biolucida to be used as a developmental biology data visualization system, showing spatially distributed phenomena over progressive stages of development.
4. **“Personalized medicine” display.** The focus group described several applications designed to visualize patient medical record data. Such applications would digest chart information and medical history and present visualizations to either the patient or the provider for enhanced comprehension or communication.
5. **Illustrating ontology concepts.** Ontologies, such as the FMA, would benefit from having a visualization engine which can instantiate concepts included in a view or in the results of a query as a 3D scene.

Analysis of the features required by the aforementioned applications reveals that Biolucida supports most of these functional needs. However, Biolucida does not currently offer Representations which include a dimension of time. This will most likely be remedied in the next release of the software, and only requires that concepts be identified by both terminology and by time, such as is the case with organ models derived from developmental biology atlases.

The provisioning of features alone will not guarantee the adoption of a software framework by developers. There are characteristics of libraries, frameworks, and extensible applications which determine the amount of effort, likelihood of success, and potential risk in deploying a software solution. In order to shed some light on what factors may influence the adoption of Biolucida within the software engineering community, the focus group was also asked to discuss factors which placed a library or application framework in their favor when developing applications which leveraged such resources. The group listed the following factors.

1. **Proper licensing.** While not an architectural feature of any software system, the manner in which it is offered for use by a community, university, or company is an important political factor which will inevitably affect whether or not it is used in a project. For reasons of flexibility and indemnity, non-viral Open Source licenses such as LGPL are preferred.
2. **Comprehensive documentation.** Documentation which describes not only the architecture of the software, but also details its methods and provides adequate examples was cited as a critical component of a usable software library.
3. **Robust operation.** Software which has been adequately tested and its operation validated, instills the confidence necessary for its inclusion in mission-critical systems.
4. **Wide adoption.** Software which has already been included in a variety of systems carries with it a sense of reliability and appeal.
5. **Large user community.** Software which not only is in use in a number of applications authored by a large group of developers, but is also supported by an active population of contributors, is more likely to continue its viability.
6. **Modular architecture.** Libraries which have a modular architecture allow developers to include portions of their functionality rather than requiring the packaging of potentially unnecessary code and resources leading to unnecessarily large deployments.
7. **Properly managed dependencies.** Libraries which contain other dependencies must provide proper documentation and straight forward methods for managing such resources through development utilities such as Maven.

While many of the software attributes discussed by the focus are outside the development scope of this project, such as licensing strategy and the fostering of a community, they highlight the importance of efforts that lay beyond effective software design and implementation. Clearly, Biolucida will require significant logistical support to become widely used should it be released as a production system.

10.3 Forward-looking research topics and design developed from further analysis of the atlas motif from the perspective of GIS

Biolucida has the capability of creating visualizations populated with arbitrary concepts and their associated 3D models. Given this unrestricted and flexible design, Biolucida is capable of constructing many types of visualizations beyond those presented in this work. Biolucida's visualization capabilities allow for the production of graphs, plots, and abstract 3D illustrations to represent its underlying datasets. However, we chose to give atlas-based visualizations special focus in this work because of their effectiveness, expressivity, common-appeal, and relevance to much of the informatics work carried out in biomedicine today.

We must keep in mind that biomedical atlas visualizations are relatively new and it is therefore difficult to establish conventions and guidelines for these types of representations given the limited history of work in this discipline. On

the other hand, GIS has focused exclusively on this type of visualization for over fifty years [127]. By analyzing the commonalities between GIS and Biomedical atlases, we can gain a better understanding of forward-looking design guidelines, conventions that inform proper implementation, and important research topics for the future.

10.3.1 Placing biomedical atlas visualizations within the context of GIS

The term “atlas” has been used within biomedicine to describe a range of works from a collection of anatomical illustrations to an abstract set of images showing locations of genetic loci on a chromosome. Yet, mentioning the term *atlas* beyond the domain of the natural sciences conjures up ideas originating from cartography, where compendiums of maps have been assembled for more than 700 years to serve as a convenient repository of information to allow navigators and historians to perform calculations or communicate ideas more effectively. Maps have been one of the most commonly used communication artifacts since the beginnings of human history. It is no mistake that many of the natural sciences, Biomedicine included, have adopted the motif to use as a form of visualization. Maps are basically representations of real places and the things located within them, with varying levels of abstraction. This representation is combined with external data to create informative visualizations. There are a number of advantages that maps, or atlas-based visualizations, grant to both the author and the reader.

1. Many visualizations are already based on illustrating real objects, therefore atlas-based visualizations can leverage a large collection of existing content and achieve instant comprehension.

Though geographic maps can represent physical features of the earth in a number of different ways, interpretation of these features usually requires minimal guidance since these features exist in a spatial representation modeled after our own reality. It is an advantage whenever a visualization can capitalize on existing mental models which govern the identification of its own symbology. The visualization of real objects is common in a number of scientific disciplines, especially in biomedicine. The fusion of medical image modalities and the display of *in situ* hybridization studies are a few examples of visualizing data as representations in terms of real anatomical structures. Since these methods have been practiced in biomedicine for decades, both methods of content production and a familiarity with these modalities support the development and use of atlas-based visualizations. Attaching data to the modeled structures, then, leverages this familiarity to create a comprehensible artifact, establishing a “triple linkage” between the reader’s world, the subject which is illustrate, and information applicable to that subject [151].

2. Map content, when used as a substrate for visualization, need not be application-based and can therefore be repurposed many ways.

“Maps make present the cumulative labor of the past” [152], providing painstakingly acquired geographical data which represents both the earth’s physical features and abstract concepts such as political boundaries. GIS practitioners can use these provided data to build novel visualizations without the need of reproducing such spatial foundations for themselves. Map data is used for a large number of types of visualizations regardless of what information is to be displayed or how it is to be used. For instance, maps of the ocean floor have enough common features, whether they are used for submarine navigation or for scientific exploration, that they can share the same topographic source.

Biomedical map data is no different. The framework of the human body and model organisms can, and should, be used as a common visualization framework for a variety of data.

3. **Use of atlases for visualization provides a common frame of reference for the integration of data and for collaborative analyses.** Authors have made the argument that atlases are good examples of early heterogeneous data integration [95], not only because of their interrelated visualizations, but also because of the variety of data that can be attributed to the areas that they represent. Because of their use of real-world concepts such as location and time, maps provide an effective structure for data integration because their content is directly referenced across a wide variety of information resources. For instance, public health data of disease outbreaks may contain the county or town of the residences of individual cases, which is easily referenced in a map. Other datasets may also contain references to place and time, and upon combining such information, one has effectively leveraged a map as a framework for data integration. Universally referenced concepts such as place and time (Biomedicine uses the concept of anatomical location to serve as “place” in a map) also provide researchers with a common frame of reference on which different analyses can be organized to support integration and collaboration. For example, a map of an area can visually combine a collection of data points to create a weather simulation, which can be subsequently combined with electrical grid data to show predictions of geographically distributed energy usage. Creating both simulations may represent a daunting task for a single researcher, but the map provides two specialists with a means of combining their insights to produce new data. While GIS has taken full advantage of such a common framework, biomedicine has only scratched the surface with projects like EMAGE.

The following sections elaborate on the above statements, consider additional questions within each topic, and discussing in detail some valid areas of investigation which can serve as areas of future work.

10.3.2 The use of a range of content in atlas-based visualization: the real, the approximate, and the high utility of the iconographic

The notion of comparing Biolumina’s visualizations (and the atlas-based visualizations of other systems) to geographic maps and atlases inevitably spawns a discussion of accuracy of representation. After all, the popular perception of maps is that their primary responsibility is to accurately depict the spatial dimensions and physical features of the earth’s surface. This belief highlights a hidden assumption that one should use the most accurate representations possible when creating a map.

The perception of accuracy trumping all other attributes of a map extends to content derived from CT and MRI scans. It is true that 3D medical images represent accuracy, reality, and a single truth. Biolumina, however, uses a surface-based modeling approach for the production of its visualizations for purposes of content availability and optimization of graphic display. One might say that to use visualizations which feature anatomical structures built in 3D modeling software seems fallible, and that one should build visualizations on voxel-based content such as in

the case of the EMAGE and Allen Brain Atlas systems. This view, however, is an exceedingly simple view of the role of maps as visualizations, and of the purposes of those visualizations.

10.3.2.1 Abstractions can be more effective than exact depictions within visualizations

When viewed within the context of both biomedicine and GIS, atlas visualization should not always be designed as a representation of real experience. On the positive side, a visualization can be *better* than reality in many ways. Any well designed artifact that conveys information to a user limits the amount of information to that which is necessary for the task at hand. Maps serve as great examples. A roadmap may represent the surface of the earth, but it differs greatly from a satellite photo in that it omits certain entities, such as topographic features, and includes non-real data elements such as the names of streets or symbols representing various services. Obviously, a lost driver would prefer such an abstraction to a satellite photo. Biolumida's visualizations have the expressive power to *represent* reality, yet offer a level of customization that allows for creating an appropriate abstraction [8].

10.3.2.2 Both geographical maps and biomedical images are subject to error

Though geographical maps focus on accuracy, they are still subject to error. Cartography has always labored to achieve representations which most accurately depict real features of our world, but mapping a spherical structure onto a flat surface leads to unavoidable approximations. Projections of spherical surfaces onto flat paper have concerned cartographers throughout history, and have represented a substantial barrier to accurate representation and analysis of size, distance, and other attributes. Various methods to mitigate the interpretive inaccuracies have been developed, such as the Mercator and the Secant projection. There are, however, tradeoffs with every method. Selection of such a method is dictated by a priority list of what is important to preserve, and this list is determined by the purpose for which the map is used.

Atlas-based biomedical visualization is no different from cartography with respect to potential errors in the accuracy of its anatomical representations. General inaccuracies can arise directly within the scanning and 3D tiling process, where Hemmy referred to CT and MRI scan measurements containing errors with twice the magnitude of caliper measurements on the same samples [52]. Errors can also arise when attempting to fuse multiple image modalities or register scans from multiple subjects. Errors within systems such as EMAGE, where an author is responsible for physically "painting" the experimental data, can arise simply through human error – a source that is difficult to identify. One might ask, then, is the resource-intensive use of voxel-based data worthwhile, when the perception of accuracy is not necessarily substantiated? Furthermore, do visualizations which use such modalities give the viewer a false sense of precision? These are valid research topics to investigate, but beyond the scope of this work.

10.3.2.3 Atlas-based visualizations as subjective content, hypotheses, and social constructs

Maps are generally regarded as representations of the world, when in fact they represent social constructions, complete with bias, opinion, and fallibility. Maps do not always show an objective reality and often exhibit inherent biases. For instance, everyone knows where various residences are, but a map is a socially constructed

view of the township they are considered part of, which may be in dispute [96]. By integrating data with the spatial features of maps, the viewer connects abstract, contested, or proposed phenomena to other aspects of reality which are not questionable, such as the illustration of a disputed fence line superimposed on a satellite photo of the land it divides. “Maps serve interests because they selectively represent the information acquired in the past. They are not wholly reconstructing the past, but are effective abstractions of it.” [151].

The false sense of precision that accompanies illustrations using seemingly accurate voxel-based scans and superimpositions are one example in which a viewer can be misled by what is portrayed in a biomedical visualization. Visualizations can certainly contain biases and hypotheses combined with factual representations. For example, the fusion of fMRI images from many subjects to an atlas describing a “common” human brain contains the hypotheses of a registration algorithm which requires intimate knowledge of its strengths and potential for errors to fully understand the significance of the resulting images.

10.3.2.4 A discussion of information and evidence in relation to accurate and iconographic visualizations

The GIS community has engaged in ample discussion regarding the classification of knowledge in terms of a hierarchy. Norman, et al [87] has been heavily cited as the first research to apply the classifications of data, information, evidence, knowledge, and wisdom. While it may seem arbitrary to discuss such a classification scheme here, it does provide some interesting perspectives on atlas-based visualizations. The classification scheme highlights the philosophical difference between overlay and attributed visualizations, and more specifically, surface-based versus voxel-based data.

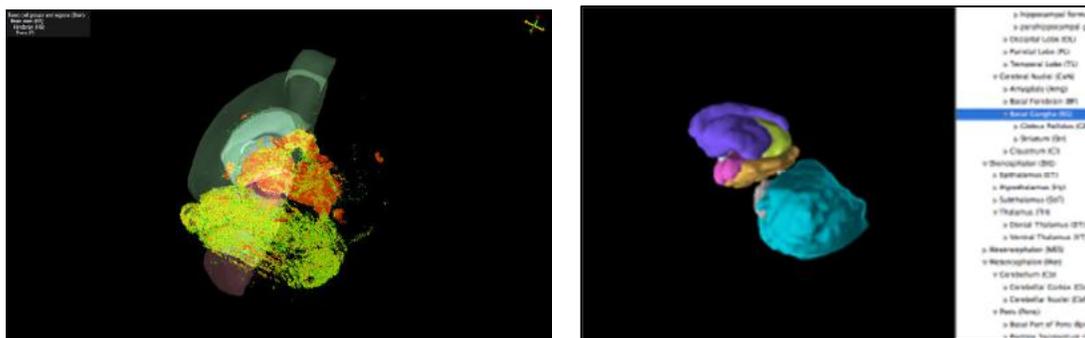


Figure 10a. Overlay and Semantically Attributed visualizations in Brain Explorer 2.

The difference between visualizations which rely on an overlay approach, such as the ISH images combined with an outline map of a model organism’s anatomy, and Bioluminescence’s visualizations which use an attributed approach to associate data with concepts/models in an object-based sense, mimics the differences between information and evidence, as discussed by Norman et al. Norman claims that data is raw information, such as a temperature at a given time and place. Data becomes information when the data is given context, such as when a temperature/place/time is viewed within the context of the current weather within Washington State. Evidence is viewed as information which is collected, evaluated, and organized into a comprehensible compilation. This

suggests that there is human intervention or automated processes involved in the transition between information and evidence. For example, the aforementioned ISH visualization can be thought of as information, since it involves a single source of data (the ISH image) combined with a contextual artifact (the atlas). On the other hand, when curators invest effort in interpreting ISH images, denoting the site of expression for particular genes and storing that information into a database, it becomes evidence. While Biolucida is capable of visualizing *information*, its primary purpose is to display *evidence* to create its attributed visualizations.

Because Biolucida primarily uses a surface-model based atlas as a visualization framework, the wholesale association between records in a dataset and the concepts to which they refer creates visualizations which display *evidence*, or *information* already collated and associated with the rendered scene. Users are free to browse the dataset to investigate the provenance of the associations, but the end result presented to them is the product of information processed by either Biolucida's data-attribution modules or by *evidence* production processes at the source of the dataset. In either case, these associations convert Biolucida's atlas models from spatial reference-frames for the analysis of raw data, into symbols, or iconography. While the anatomical structures used in the scene still have spatial significance, the fidelity with which a visualization must represent them is significantly lower than in a rendering such as an ISH visualization. The anatomical models used in such an iconographical need only be recognizable to the viewer and show fairly accurate spatial representation for the granularity that the visualization represents. Using this type of visualization carries with it two advantages: iconography is (1) more representative of the type of visualized information, its accuracy, and provenance, and (2) more generalizable than *information-analysis* visualizations.

10.3.2.5 Domain-specific applications are still relevant for the analysis of *information* to produce *evidence*

Our earlier analysis of the visualization applications created to date within biomedicine revealed a large number of systems that were very domain specific, and not constructed in a manner to allow much extensibility beyond the specific tasks for which they were designed. However, it is clear that many of these systems were designed as analysis applications for evaluating *information* to produce *evidence*, a role which Biolucida is not designed to fill at the moment. Rather than consider these systems as full data visualization platforms, we feel that they are best regarded as data analysis applications, and as such allow users to populate databases with *evidence*, serving as resources for higher-level visualization applications such as Biolucida. As data-analysis software, there will always be a place for applications like these within the tool set of any given domain.

10.4 Future work: research endeavors informed by the concepts of distributed cognition

The idea of working together is a much more powerful concept than the phrase suggests. The efficient collaboration of many minds and artifacts can produce a more powerful system than those same minds working independently. The human mind has not changed in its capabilities for thousands of years, yet the evolving social structures through which we work have cast the illusion the humankind is in fact becoming more intelligent. In reality, our distributed cognitive systems have enabled us to reduce the scope of an individual's information space, while increasing the

complexity of the system as a whole. Higher levels of cognition are achieved via a distributed cognitive “workaround”, which produces a higher level of cognition without the necessity of augmenting our individual processing capabilities. Scientific research is a prototypical example of distributed cognition, because it is based on distributing cognitive effort over time, over people, and over artifacts.

Edwin Hutchins pioneered the concept of distributed cognition in his seminal work *Cognition in the Wild* [59], a cognitive anthropological account of pilotage aboard a navy vessel. Hutchins’ ethnographic study revealed several components of a distributed cognitive system which operates efficiently. First, efficient means of communication must be present. If there is no way to effectively communicate information, distribution of cognitive activity becomes impossible. Second, a clear understanding of where certain information can be obtained must exist among all parties involved. Comprehension of the inputs and outputs is required for the efficient operation of such a network. Third, the volume and complexity of the information that any one node handles cannot saturate its capacity for processing. Distributed cognition most often occurs in situations where one person cannot possibly perform all cognitive tasks required to accomplish whatever goal the situation dictates.

Karin Knorr Cetina described an environment similar to Hutchins’ ethnographic study, but in a scientific setting in her book *Epistemic Cultures*. Cetina studied the processes and culture of scientific research in two very different domains: high energy physics (HEP), and molecular biology. Cetina describes, without previous knowledge of Hutchins’ theory, distributed cognition between the members of research teams performing large-scale experiments using the CERN accelerator in Geneva, Switzerland. The scale of the projects undertaken at CERN, like many within HEP, are beyond the intellectual expertise of a single researcher. Therefore, distributed cognition is an integral part of HEP research culture. Biomedicine, however, presents an entirely different situation.

Cetina noted a much different structure within the typical molecular biology laboratory which seems more akin to a collection of isolated small projects. In molecular biology, Cetina found that a single researcher tends to “own” a project or experiment. While much of clinical medicine is based on teamwork and distributed cognition, a majority of “wet bench” biologists function in their own operational silos. Most laboratories are run by a primary investigator, supervising a team which is composed of individuals who are responsible for their own day-to-day research protocols, which are isolated endeavors from their peers’. Though researchers may communicate through intimate and national-scale meetings, or via publication, the phenomenon of distributed cognition as seen in other disciplines is often almost absent.

Only recently has biomedicine endeavored to undertake projects of similar scale to those seen at CERN. The Human Genome Project and the Human Brain Project are examples of large endeavors where scale and integration of results may push for processes, tools, and practices which enable distributed cognition. To overcome this tendency in biomedical research so that we can make our community more productive, we must design tools to facilitate the adoption of distributed cognition. In support of its promotion across the biomedical sciences, future endeavors to improve Biolumina’s functionality will be guided by Hutchins’ three requirements: (1) effective communication, (2) awareness of information sources, and (3) reduction in cognitive load.

10.4.1 Effective communication: collaborative visualization through shared worlds

Hutchins' first requirement of an efficient distributed cognitive system is that members within a group are able to communicate effectively, whether by means of direct communication, or via some artifact. As we our discussed earlier, the field of biomedical research is fast becoming a digital world, populated with many types of research tools – software applications of all scopes and sizes which save their users time and effort. However, tools that exist in a vacuum are not necessarily as effective as they would be if they were networked applications, enabling the kind of communication to which Hutchins is referring.

Biolucida is designed to allow its Representations, or scenes, to maintain state on the server. This means that multiple clients can connect to the same Representation and experience a common visualization. The system currently allows such concurrent connections to take place, but “peer” clients may only connect to the scene and interact with its constituent models in the state they were in at the time of their initial connection. There currently exists no means of transmitting updates on the server originating from the primary author, or possibly from connected collaborators, back to all of the currently connected clients. It is a priority that this functionality be implemented to encourage collaborative analysis of visualizations, and to conduct research on the effects of viewing a visualization as a shared virtual world.

10.4.2 Supporting information resource awareness: plugin resource centralization

It is a powerful feature that Biolucida has implemented which allows for a variety of plugin objects to extend its capabilities, enabling a number of derivative applications, visualizations, and data sources to be created. However, it is unclear how administrators of the Biolucida servers will be able to find these plugins, verify their authenticity, and develop an understanding of their functionality. It is our intention to research ways in which other popular frameworks, toolkits, and standalone applications manage and inform users of the ecosystem of plugins that extends their capabilities. After gaining an understanding of which methods are most successful, a prototype management system will be created, tested, and deployed to Biolucida's future user community.

10.4.3 Reduction in cognitive load by specialization: Linked server domains

While Biolucida's atlas-based approach is an effective method of providing extensible integrative visualization platform for the biomedical sciences, it must employ strategies discussed in chapter 7 for mitigating the perpetual issue of a lack of content and concepts which any incoming data may reference. In short, no one can implement an atlas that models the entire world and every organism found in it. Moreover, no group or project team could model an organism at every possible level of granularity.

Enabling effective distributed cognition entails each “participant” taking a role no larger than they can handle. Therefore, if Biolucida server teams are to begin representing a vast sea of biomedical knowledge, they best do so using a policy of specialization, where each server is responsible for representing a manageable subdomain of an organism. For example, rather than have a single server manage a concept catalog which describes the entire human

body, perhaps there could exist a collective group of systems, each of which manage an organ system, or region of the body. With each server having partial knowledge of an entire organism, systems of communication and referencing would have to be implemented where each server may develop an awareness of the conceptual domains of other servers, and may also reference the content located on those systems.

Specialization can mean much more than the scope of concepts which a Biolucida server seeks to manage, for it is within this idea that there is an opportunity to explore biological semantic level of detail in a whole new way. Consider the possibilities of a set of servers managing concepts at the cellular, tissue, anatomical structure, and whole organism level. If a means of seamlessly moving between visualizations dealing with differing levels of granularity modeled on different systems were properly implemented, a most interesting set of interactive visualizations could be created to support more than just research, education, and communication within Biomedicine. It is through the implementation of such *conceptual level of detail* that a bench-to-bedside flow of knowledge may manifest itself in the form of visualizations, to be used by the translational medicine research community.

WORKS CITED

1. ADAM Software. "Adam Scholar Series." CD-ROM, 1995.
2. Ahlberg, C., and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In SIGCHI '94 Conference Proceedings, pages 313–317, Boston, Massachusetts, April 1994.
3. Alvarez-Bolado G, Eichele G. Analysing the developing brain transcriptome with the GenePaint platform. *J Physiol.* 2006 Sep 1;575(Pt 2):347-52.
4. Baldock RA, Bard JB, Burger A, Burton N, Christiansen J, Feng G, Hill B, Houghton D, Kaufman M, Rao J, Sharpe J, Ross A, Stevenson P, Venkataraman S, Waterhouse A, Yang Y, Davidson DR. EMAP and EMAGE: a framework for understanding spatially organized data. *Neuroinformatics.* 2003;1(4):309-25.
5. Bangor, Aaron, Kortun, Philip, and Miller, James, 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4 (3) May 2009, 114-123.
6. Barillot C, Lemoine D, Le Briquer L, Lachmann F, Gibaud B. Data fusion in medical imaging: merging multimodal and multipatient images, identification of structures and 3D display aspects. *Eur J Radiol.* 1993 Jun;17(1):22-7.
7. Bartling WC, Schleyer TK: An application of Geospatial Information System (GIS) technology to anatomic dental charting. *AMIA Annu Symp Proc* 2003:786.
8. Bartram, D. J., Comprehending Spatial Information: The Relative Efficiency of Different Methods of Presenting Information About Bus Routes. *Journal of Applied Psychology*, 65, pp. 103-110.
9. Belmamoune M, Verbeek FJ. Data integration for spatio-temporal patterns of gene expression of zebrafish development: the GEMS database. *J Integr Bioinform.* 2008 Aug 25;5
10. Biomed Central Group. Biomed Central. Biomed Central Databases. N.D. Available at: <http://databases.biomedcentral.com/browsecatalog>. Accessed Jan 22, 2012. 2010.
11. Blythe, D. The Direct3D 10 system. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2006 Volume 25 Issue 3*, July 2006
12. Bolliger SA, Thali MJ, Ross S, Buck U, Naether S, Vock P. Virtual autopsy using imaging: bridging radiologic and forensic sciences. A review of the Virtopsy and similar projects. *Eur Radiol.* 2008 Feb;18(2):273-82.
13. Braunstein, M.L. (1986) Perception of rotation in depth: the psychophysical evidence. In N.I.Badler & J.K. Tsotsos (Eds.), *Motion: representation and perception* (pp. 242-247). Elsevier Science.
14. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FC, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A, Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, and Vingron M. Minimum information about a microarray experiment (MIAME): toward standards for microarray data. *Nat Genet* 29: 365–371, 2001

15. Brooke, J. (1996) SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester & A. L. McClelland (eds.) Usability Evaluation in Industry. London: Taylor and Francis.
16. Brutzman, Don and Leonard Daly. X3D: 3D Graphics for Web Authors. Morgan Kaufmann Publishers, Waltham, MA. 2007.
17. Bug WJ, Gupta A, Gustafson C, Martone ME, Nissanov J, Wong WW, Zaslavsky I (2006) Integrating Brain Atlas Tools for Broad-field Morphometric Analysis and Semantic Data Integration. Society for Neuroscience Conference, Atlanta, Oct 16, 2006
18. Blender Group. Blender. Blender Home. N.D. Available at: <http://www.blender.org>. Accessed January 10, 2011.
19. Chhadia A, Dech F, Ai Z, Silverstein JC Autocolorization of Three-Dimensional Radiological Data. Stud Health Technol Inform 2001; 81:90-96
20. Chien-Ping Lu. K3 Moore's Law in the Era of GPU Computing . International Symposium on VLSI Design Automation and Test (VLSI-DAT). 2010. April. P. 5
21. Clark, J.H. Hierarchical Geometric Models for Visible Surface Algorithms, Comm. ACM, vol. 19, no. 10, pp. 547-554, Oct. 1976.
22. Codd AM, Choudhury B. Virtual reality anatomy: is it comparable with traditional methods in the teaching of human forearm musculoskeletal anatomy? Anat Sci Educ. 2011 May;4(3):119-25.
23. Collada Group. Collada. Collada File Format. N.D. Available at: <http://www.collada.org>. Accessed Dec 22, 2010.
24. Cromby JJ, Standen PJ, Brown DJ. The potentials of virtual environments in the education and training of people with learning disabilities. J Intellect Disabil Res. 1996 Dec;40 (Pt 6):489-501.
25. Dang C, Sodt A, Lau C, Youngstrom B, Ng L, Kuan L, Pathak S, Jones A, Hawrylycz M (2007) The Allen Brain Atlas: Delivering neuroscience to the Web on a genome wide scale. Data Integration in the Life Sciences in Lecture Notes in Computer Science 4544:17-26.
26. Davies, Kevin. First Base: Questions of Science and Progress. Bio-IT World. Sept 09 2005
27. de Boer BA, Ruijter JM, Voorbraak FP, Moorman AF. More than a decade of developmental gene expression atlases: where are we now? Nucleic Acids Res. 2009 Dec;37(22):7349-59.
28. Derthick, M., Kolojechick, J., and Roth S.. An Interactive Visual Query Environment for Exploring Data. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '97), ACM Press, October 1997, pp 189-198.
29. Diczfalusy E, Zsigmond P, Dizdar N, Kullman A, Loyd D, Wårdell K. A model for simulation and patient-specific visualization of the tissue volume of influence during brain microdialysis. Med Biol Eng Comput. 2011 Dec;49(12):1459-69. Epub 2011 Nov 13.
30. Dilmanian FA, Qu Y, Liu S, Cool CD, Gilbert J, Hainfeld JF, Kruse CA, Larterra J, Lenihan D, Nawrocky MM, Pappas G, Sze CI, Yuasa T, Zhong N, Zhong Z, McDonald JW. X-ray microbeams: Tumor therapy and central nervous system research. Nucl Instrum Methods Phys Res A. 2005 Aug

- 11;548(1-2):30-37.
31. Divintini, M., Farchshian, B., Tuikka, T.. Internet-based groupware for user participation in product development. *ACM SIGCHI Bulletin*. Vol 32 Issue 1, p. 31 - 35 Jan 2000
 32. Drupal Group. Drupal. Drupal, Get Involved. N.D. Available at: <http://drupal.org/getting-involved-guide>. Accessed Jun 6, 2011.
 33. Edgar, R., Domrachev, M. and Lash, A.E.. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 2002, Vol. 30, No. 1 207-210
 34. Eick, S.G., J. L. Steffen, and E. E. Sumner Jr. Seesoft™ —a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, November 1992.
 35. Eick, S.G. T. L. Graves, A. F. Karr, A. Mockus, and P. Schuster. Visualizing software change. Submitted for publication in *IEEE Transactions on Software Engineering*, 2000. Available at <http://home.visualinsights.com/Research/WhitePapers/vizchanges.pdf>.
 36. Feng G, Burton N, Hill B, Davidson D, Kerwin J, Scott M, Lindsay S, Baldock R. JAtlasView: a Java atlas-viewer for browsing biomedical 3D images and atlases. *BMC Bioinformatics*. 2005 Mar 9;6:47.
 37. Feng D, Marshburn D, Jen D, Weinberg RJ, Taylor RM 2nd, Burette A.. Stepping into the third dimension. *J Neurosci*. 2007 Nov 21;27(47):12757-60.
 38. Flynn CJ, Sharma T, Ruffins SW, Guerra SL, Crowley JC, Etensohn CA. High-resolution, three-dimensional mapping of gene expression using GeneExpressMap (GEM). *Dev Biol*. 2011 Sep 15;357(2):532-40. Epub 2011 Jun 29.
 39. Garb J.L., J, Ganai, S, Skinner ,R, Boyd, C, Wait, R.B., . Using GIS for spatial analysis of rectal lesions in the human body. *International Journal of Health Geographics* 2007, 6:11
 40. Gobbetti, E., Scateni, R.. *Virtual Reality: Past, Present, and Future*. *Virtual Environments in Clinical Psychology and Neuroscience*. Riva, G., Wiederhold, B., Molinari, E. (eds).1998. Ios Press: Amsterdam, Netherlands
 41. Goebel, Rainer. *BrainVoyager — Past, present, future*. *Neuroimage*.2012.01.083
 42. Gold Standard Multimedia, Inc. “Human Anatomy.” CD-ROM, 1996.
 43. Google, Inc. Google. Google Earth. November 1, 2007. Available at: <http://earth.google.com>. Accessed June 20, 2009.
 44. Gray, Henry. *Anatomy of the Human Body*. Lea & Febiger. Philadelphia. 1918.
 45. Halper, M., Geller, J., and Perl, Y.. Value propagation in object-oriented database part hierarchies. In B. Bhargava, T. Finin, and Y. Yesha, editors, *Proc. of the 2nd Int. Conf. on Information and Knowledge Management, CIKM'93*, pages 606-614, Washington DC, USA,. ACM Press. 1993.
 46. Handels H, Hacker S. A framework for representation and visualization of 3D shape variability of organs in an interactive anatomical atlas. *Methods Inf Med*. 2009;48(3):272-81.
 47. Harrell WE Jr, Hatcher DC, Bolt RL. In search of anatomic truth: 3-dimensional digital modeling and the future of orthodontics. *Am J Orthod Dentofacial Orthop*. 2002 Sep;122(3):325-30.
 48. Haverty, PM., Weng, Z., Best, N., Auerbach, K., Hsiao, L., Jensen, R., Gullans, SR. *Huge Index: A*

Database of High Density Oligonucleotide Array Data from Normal Human Tissues. *Nucleic Acids Res.* 2002 January 1; 30(1): 214–217.

49. Heinonen T, Visala K, Blomqvist M, Eskola H, Frey H. 3D visualization library for multimodal medical images. *Comput Med Imaging Graph.* 1998 Jul-Aug;22(4):267-73.
50. Hemmy DC, Zonneveld FW, Lobregt S, Fukuta K. A decade of clinical three-dimensional imaging: a review. Part I. Historical development. *Invest Radiol.* 1994 Apr;29(4):489-96.
51. Hemmy DC, Zonneveld FW, Lobregt S, Fukuta K. A decade of clinical three-dimensional imaging: a review. Part II: Clinical applications. *Invest Radiol.* 1994 May;29(5):574-89.
52. Hemmy DC, Zonneveld FW, Lobregt S, Fukuta K. A decade of clinical three-dimensional imaging: a review. Part III. Image analysis and interaction, display options, and physical models. *Invest Radiol.* 1994 Jul;29(7):716-25.
53. Hermann M. [3-dimensional computer animation--a new medium for supporting patient education before surgery. Acceptance and assessment of patients based on a prospective randomized study-- picture versus text]. *Chirurg.* 2002 May;73(5):500-7.
54. Hillen, B.. "Temporal Bone and Posterior Cranial Fossa." CD-ROM, 1995.
55. Hisley KC, Anderson LD, Smith SE, Kavic SM, Tracy JK. Coupled physical and digital cadaver dissection followed by a visual test protocol provides insights into the nature of anatomical knowledge and its evaluation. *Anat Sci Educ.* 2008 Jan;1(1):27-40.
56. Höhne, K.H., Andreas Petersik, Bernhard Pflesser, Andreas Pommert, Kay Priesmeyer, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede, Markus Urban, Hans Frederking, Mike Lowndes, John Morris: VOXEL-MAN 3D Navigator: Brain and Skull. Regional, Functional and Radiological Anatomy. Springer-Verlag Electronic Media, Heidelberg, 2001. (2 CD-ROMs, ISBN 978-3-540-14910-1).
57. Howe, J. The Rise of Crowdsourcing. *Wired.* Issue 14.06 - June 2006
58. Hudson Alan, Justin Couch , Stephen N. Matsuba, The Xj3D browser: community-based 3D software development, ACM SIGGRAPH 2002 conference abstracts and applications, July 21-26, 2002, San Antonio, Texas
59. Edwin Hutchins, *Cognition in the Wild.* 1996. Bradford Press, Cambridge, MA.
60. Hyrtl, Joseph. *Lehrbuch der Anatomie des Menschen.* 1851 Wilhelm Braumüller, Leipzig.
61. International Atomic Energy Agency, *The Role of PET/CT in Radiation Treatment Planning for Cancer Patient Treatment.* Vienna, Austria: IAEA, 2008.
62. IBM Visible medical record project. 2007. Sept. 2009
<<http://www.zurich.ibm.com/news/07/asme.html>>.
63. IBM Inc., Open DX. IBM Open DX, N.D. Available at: <http://www.ibm.com/dx>. Access June 30, 2011.
64. Itti, L., and Koch, C.. A model of saliency-based visual attention for rapid scene analysis. *IEEE Pattern Analysis and Machine Intelligence,* 20(11), 1998.

65. Jang, S., Jyung, R. & Black, J. (2007). How Direct Interaction in a Virtual Reality Program Aids in Developing an Internal Representation of a Complex 3-D Structure. In C. Montgomerie & J. Seale (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* (pp. 4214-4218). Chesapeake, VA
66. JOGRE Project. JOGRE. The JOGRE Project. N.D. Available at: <http://jogre.sourceforge.net/> Accessed. September 2011.
67. John NW, McCloy RF. Navigating and visualizing three-dimensional data sets. *Br J Radiol.* 2004;77 Spec No 2:S108-13.
68. John NW, Lim IS. Cybermedicine tools for communication and learning. *J Vis Commun Med.* 2007 Mar;30(1):4-9.
- 69 Johnson, C.R. "Top Scientific Visualization Research Problems," In *IEEE Computer Graphics and Visualization: Visualization Viewpoints*, pp. 2--6. July/August, 2004.
70. Johson, S. *The Ghost Map*. Riverhead Books – The Penguin Group. New York, NY. 2006.
71. Kaiser, M.K., & Proffitt, D.R. Perceptual issues in scientific visualization. Three-dimensional visualization and display technologies. *Proceedings of SPIE Volume: 1083*, 11 September 1989. 205-211.
72. Kamel Boulos MN, Roudsari AV, Carson ER. HealthCyberMap: Mapping the Health Cyberspace Using Hypermedia GIS and Clinical Codes. In: *Proceedings of the First European Conference for Geographic Information Sciences in Public Health*, Sheffield, UK, September 2001.
73. Kang HS, Kim BH, Ryu JW, Hong SH, Chung HW, Cho SY, Kim YH, Hwang SI, Jeong DK, Shin YG. The visible man: three-dimensional interactive musculoskeletal anatomic atlas of the lower extremity. *Radiographics.* 2000 Jan-Feb;20(1):279-86.
74. Kazic T, Coe E, Polacco M, Shyu CR. Whither biological database research?. *OMICS.* 2003 Spring;7(1):61-5
75. Keller, P. and Keller, M. *Visual clues: Practical Data Visualization*. Los Alamitos, CA, IEEE Computer Society Press, Los Alamitos, CA. 1993.
76. Kerlow, Isaac Victor . *The art of 3D computer animation and effects*. John Wiley and Sons, Hoboken, NJ. 2004.
77. Kerwin J, Yang Y, Merchan P, Sarma S, Thompson J, Wang X, Sandoval J, Puelles L, Baldock R, Lindsay S. The HUDSEN Atlas: a three-dimensional (3D) spatial framework for studying gene expression in the developing human brain. *J Anat.* 2010 Oct;217(4):289-99.
78. Kilmon CA, Brown L, Ghosh S, Mikitiuk A. Immersive virtual reality simulations in nursing education. *Nurs Educ Perspect.* 2010 Sep-Oct;31(5):314-7.
79. Koerner JG. The virtues of the virtual world. Enhancing the technology/knowledge professional interface for life-long learning. *Nurs Adm Q.* 2003 Jan-Mar;27(1):9-17.
80. Koutsofios, E. D. A. Keim, and S. North. Visualization of large-scale telecommunications data. *IEEE Computer Graphics and Applications*, pages 33–35, May 1999.

81. Khronos Group, WebGL Working Group. Kronos. WebGL. Available at: <http://www.khronos.org/webgl/>. Accessed Jan 20, 2012.
82. Lacroix, Zoe. Biological Data Integration: Wrapping Data and Tools. *IEEE Transactions on Information Technolgy in Biomedicien*, Vol 6, No 2, 2002.
83. Lau C, Ng L, Thompson C, Pathak S, Kuan L, Jones A, Hawrylycz M. Exploration and visualization of gene expression with neuroanatomy in the adult mouse brain. *BMC Bioinformatics*. 2008 Mar 18;9:153.
84. Lee JT, Munch KR, Carlis JV, Pardo JV. Internet image viewer (iiV). *BMC Med Imaging*. 2008 May 29;8:10.
85. Lindsay S, Sarma S, Martínez-de-la-Torre M, Kerwin J, Scott M, Luis Ferran J, Baldock R, Puelles L. Anatomical and gene expression mapping of the ventral pallium in a three-dimensional model of developing human brain. *Neuroscience*. 2005;136(3):625-32.
86. Livneh Y, Mizrahi A. A time for atlases and atlases for time. *Front Syst Neurosci*. 2010 Feb 22;3:17.
87. Longley, Paul. Michael F. Goodchild, David J. Maguire, David W. Rhind. *Geographic Information Systems and Science*. Wiley and Sons, Hoboken, NJ.. 2005.
88. LWJGL Project. LWJGL. LWJGL Home. N.D. Available at: <http://www.lwjgl.org>. Accessed December 12, 2010.
89. MAGE Working Group. MGED. MAGE-ML. April 2007. Available at: <http://www.mged.org/Workgroups/MAGE/mage.html>. Accessed April 12 2007.
90. Mantovani F, Castelnovo G, Gaggioli A, Riva G. Virtual reality training for health-care professionals. *Cyberpsychol Behav*. 2003 Aug;6(4):389-95.
91. Martone ME, Gupta A, Ellisman MH (2004) E-neuroscience: challenges and triumphs in integrating distributed data from molecules to brains. *Nature Neuroscience* 7:467-472.
92. Mathers, L., and Chase, R. A. "C.I.a.s.s.-clinical anatomy study system." Mosby, 1994. CD-ROM.
93. McGhee J. 3-D visualization and animation technologies in anatomical imaging. *J Anat*. 2010 Feb;216(2):264-70.
94. Microsoft (2006-06) (PDF), [MS-XAML] – v1.0, Xaml Object Mapping Specification 2006, Copyright © 2008 Microsoft Corporation. Release: June 2008.
95. Monmonier, Mark. *Mapping it out: expository cartography for the humanities and the social sciences*. University of Chicago Press, Chicago, IL. 1993.
96. Monmonier, Mark. *How to Lie with Maps*. University of Chicago Press, Chicago, IL. 1996.
97. Mustafa, N., Krishnan, S., Varadhan, G., Venkatasubramanian, S.. "Dynamic Simplification and Visualization of Large Maps," *International Journal of Geographical Information Science*, 2002
98. Nielsen, J., and Mack, R. L. (Eds.). *Usability Inspection Methods*. John Wiley & Sons, Hobken, NJ. 1994.
99. Nieman BJ, Wong MD, Henkelman RM. Genes into geometry: imaging for mouse development in 3D. *Curr Opin Genet Dev*. 2011 Oct;21(5):638-46.

100. Norman, D. A. Cognitive artifacts. In J. M. Carroll (Ed.), *Designing Interaction* (pp.17- 38). Cambridge University Press. New York, NY. 1991.
101. Garg, A., Norman, G. R., Eva, K., Spero, L., & Sharan, S. (2002). Is there any virtue in virtual reality? The minor role of multiple orientations in learning anatomy from computers. *Academic Medicine*, 77, S97-S99.
102. Nowinski WL, Thirunavuukarasuu A, Volkau I, Marchenko Y, Aminah B, Puspitasari F, Runge VM. A three-dimensional interactive atlas of cerebral arterial variants. *Neuroinformatics*. 2009 Dec;7(4):255-64.
103. Nowinski WL, Chua BC, Qian GY, Nowinska NG. The human brain in 1700 pieces: Design and development of a three-dimensional, interactive and reference atlas. *J Neurosci Methods*. 2012 Feb 15;204(1):44-60.
104. Ogunyemi OI. TraumaSCAN: Assessing Penetrating Trauma with Abductive and Geometric Reasoning. Ph.D. Thesis. Philadelphia (PA): University of Pennsylvania, 1999.
105. Ogunyemi O, Clarke JR, Ash N, Webber B. Combining Geometric and Probabilistic Reasoning for Computer-Based Penetrating Trauma Assessment. *Journal of the American Medical Informatics Association*. 2002 May-Jun;9(3):273-82.
106. Paivio, A. The empirical case for dual coding. In J.C. Yuille (Eds.) *Imagery, memory, and cognition* (pp. 307-332). Hillsdale, NJ. 1983.
107. Parkinson, H. , Sarkans, U. , Shojatalab, M., Abeygunawardena, N., Contrino, S., Coulson, R., Farne, A., Garcia G, Holloway, E., Kapushesky, M., Lilja, P., Mukherjee, G., Oezcimen, A., Rayner, T., Rocca-Serra, P., Sharma, A., Sansone, S., Brazma, A.. ArrayExpress—a public repository for microarray gene expression data at the EBINucl. *Acids Res.*, 2005, 33: D553-D555.
108. Parthenon Publishing Group. “The Anatomy Project.” CD-ROM, 1996.
109. Pennisi, E.. "How will big pictures emerge from a sea of biological data?" *Science* 309 (5731), 75-102. 2005.
110. Petersson H, Sinkvist D, Wang C, Smedby O. Web-based interactive 3D visualization as a tool for improved anatomy learning. *Anat Sci Educ*. 2009 Mar-Apr;2(2):61-8.
111. Pieper S., Halle M., Kikinis R. 3D SLICER Proceedings of the 1st IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2004; 632-635.
112. Preece, J., Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, John Wiley and Sons, Hoboken, NJ. 2007.
113. Richardson L, Venkataraman S, Stevenson P, Yang Y, Burton N, Rao J, Fisher M, Baldock RA, Davidson DR, Christiansen JH. EMAGE mouse embryo spatial gene expression database: 2010 update. *Nucleic Acids Res*. 2010 Jan;38(Database issue):D703-9.
114. Riechardt, T. It's Sink or Swim as a Tidal Wave of Data Approaches. *Nature* Vol 399, June 1999. 517–520.
115. Robb RA.3-D visualization in biomedical applications. *Annu Rev Biomed Eng*. 1999;1:377-99.

116. Roosendaal, T. Hess, R. Blender Foundation. *The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender*. 2007. No Starch Press, San Francisco, CA.
117. Rosse, C., & Mejino, J. L., Jr. (2003). A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J Biomed Inform*, 36(6), 478-500.
118. Roth NM, Kiani MF: A "geographic information systems" based technique for the study of microvascular networks. *Ann Biomed Eng* 1999, 27:42-47.
119. Sarkar IN. Biomedical informatics and translational medicine. *J Transl Med*. 2010 Feb 26;8:22.
120. Sarma S, Kerwin J, Puelles L, Scott M, Strachan T, Feng G, Sharpe J, Davidson D, Baldock R, Lindsay S. 3D modelling, gene expression mapping and post-mapping image analysis in the developing human brain. *Brain Res Bull*. 2005 Sep 15;66(4-6):449-53.
121. Schubert R, Bomans M, Höhne KH, Pommert A, Riemer M, Schiemann T, Tiede U, Lierse W. A new method for representing the human anatomy. *Comput Med Imaging Graph*. 1993 Jul-Oct;17(4-5):243-9.
122. Schubert R, Höhne KH, Pommert A, Riemer M, Schiemann T, Tiede U, Lierse W. A new method for practicing exploration, dissection, and simulation with a complete computerized three-dimensional model of the brain and skull. *Acta Anat (Basel)*. 1994;150(1):69-74.
123. Shahidi, R. Surface rendering versus volume rendering in medical imaging: techniques and applications. *VIS '96 Proceedings of the 7th conference on Visualization '96* pp. 439-440
124. Shreiner, D., M. Woo, J. Neider, T. Davis. *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. Addison-Wesley, Boston, MA. 2005.
125. Silva, CT, and Freire J. Software Infrastructure for exploratory visualization and data analysis: Past, present, and future 2008 *J. Phys.: Conf. Ser.* 125 012100
126. Slicer Dicer Team. Slicer Dicer. Slicer Dicer Volume Data Visualization Software, 3D, 3-D Now shipping v5. Available at: <http://www.slicerdicer.com/>. Accessed January 14, 2012.
127. Star, Jeffrey, and John Estes. 1990. *Geographic Information Systems: An Introduction*. Prentice-Hall, Englewood Cliffs, NJ.
128. Swayne, D. F., D. Cook, and A. Buja. XGOBI: Interactive dynamic data visualization in the X window system. *Journal of Computational and Graphical Statistics*, 7(1), June 1998.
129. Theiler, Max. *The House Mouse: Atlas of Embryonic Development*. New York: Springer-Verlag, 1989.
130. Tory, M. and Torsten Möller, "Evaluating Visualizations: Do Expert Reviews Work?", *IEEE Computer Graphics and Applications*, vol. 25, no. 5, Sept./Oct. 2005, pp.8-11.
131. Trelease RB. Anatomical informatics: Millennial perspectives on a newer frontier. *Anat Rec*. 2002 Oct 15;269(5):224-35.
132. Trelease RB, Rosset A. Transforming clinical imaging data for virtual reality learning objects. *Anat Sci Educ*. 2008 Mar-Apr;1(2):50-5.
133. Tufte, E.R.. *The Visual Display of Quantitative Information*. Graphics Press, second edition, May

2001. ISBN 0-96-139214-2.
134. Udupa, J.K.; Hung, H.-M. Surface versus volume rendering: a comparative assessment. Proceedings of the First Conference on Visualization in Biomedical Computing. 22-25 May 1990
 135. Unity, Inc. Unity 3D. Unity3D Game Engine. N.D. Available at: <http://unity3d.com/support/documentation/>. Accessed January 4, 2011.
 136. Upson, C., Faulhaber, T., Kamins D., Laidlaw D., Schlegel D., Vroom J., Gurwitz R., van Dam A. The Application Visualization System: A Computational Environment for Scientific Visualization. IEEE Computer Graphics and Applications. Volume 9 Issue 4, July 1989. 30-42.
 137. Velleman, P.F.. The DataDesk Handbook. Odesta Corporation, 1988.
 138. Venkataraman S, Stevenson P, Yang Y, Richardson L, Burton N, Perry TP, Smith P, Baldock RA, Davidson DR, Christiansen JH. EMAGE--Edinburgh Mouse Atlas of Gene Expression: 2008 update. Nucleic Acids Res. 2008 Jan;36(Database issue):D860-5.
 139. Visual Insights, Inc.. ADVIZOR. Visual Insights ADVIZOR. Available at: <http://www.visualinsights.com/advizor>. Accessed January 13, 2010.
 140. VTK Development Team, Kitware, Inc.. VTK. Visualization Toolkit (VTK). N.D.. Available at: <http://www.vtk.org>. Accessed Dec 12, 2010.
 141. VRML Consortium, "The Virtual Reality Modelling Language" ISO/IEC 14772-1:1997
 142. Wang X, Lindsay S, Baldock R. From spatial-data to 3D models of the developing human brain. Methods. 2010 Feb;50(2):96-104.
 143. Ward, M O. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In Visualization '94 Conference Proceedings, pages 326-333, Washington, DC, October 1994.
 144. Web 3D Consortium, "X3D: eXtensible 3D" ISO/IEC 19775:2004.
 145. Weber GH, Rübél O, Huang MY, DePace AH, Fowlkes CC, Keränen SV, Luengo Hendriks CL, Hagen H, Knowles DW, Malik J, Biggin MD, Hamann B. Visual exploration of three-dimensional gene expression using physical views and linked abstract views. IEEE/ACM Trans Comput Biol Bioinform. 2009 Apr-Jun;6(2):296-309.
 146. Whetzel PL, Parkinson H, Causton HC, Fan L, Fostel J, Frago G, Game L, Heiskanen M, Morrison N, Rocca-Serra P, Sansone SA, Taylor C, White J, Stoeckert CJ Jr. The MGED Ontology: a resource for semantics-based description of microarray experiments. Bioinformatics. 2006 Apr 1;22(7):866-73.
 147. Wiecha J, Heyden R, Sternthal E, Meriardi M. Learning in a virtual world: experience with using second life for medical education. J Med Internet Res. 2010 Jan 23;12(1):e1.
 148. Wills, G. J., Nicheworks – interactive visualization of very large graphs. In Graph Drawing'97 Conference Proceedings, New York. Springer-Verlag.
 149. Wong, B. A. and Rosse, C. and Brinkley, J. F. (1999) Semi-Automatic Scene Generation Using the Digital Anatomist Foundational Model. In Proceedings, American Medical Informatics Association Fall Symposium, pages 637-641.
 150. Woo, M. Neider, J. Davis, T., Shreiner, D. OpenGL Programming Guide: The Official Guide to

- Learning OpenGL, Version 1.2. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 1999.
151. Wood, Denis. The Power of Maps. Guilford Press, New York, NY. 1992.
 152. Wood, D. Kaiser, W.. Seeing through Maps. 2001. Amherst Press. Amherst, MA.
 153. Zaslavsky H, He J, Tran ME, Martone AG: Integrating Brain Data Spatially: Spatial Data Infrastructure and Atlas Environment for Online Federation and Analysis of Brain Images: Aug/ Sept 2004; Zaragosa, Spain. Biological Data Management Workshop (BIDM 2004) in conjunction with 15th International Workshop on Database and Expert Systems Applications (DEXA'04) 2004.
 154. Zygote, Inc. Zygote.com. Zygote Model Set. N.D. Available at: <http://www.zygote.com/> Accessed January 5, 2010.

APPENDIX A

3D Atlas Visualization Systems

The following list is composed of a systems which present 3D biomedical atlas content for a variety of purposes, including education, simulation, medical image display and review, data visualization, as well as other applications.

1. Adamovich SV, August K, Merians A, Tunik E. A virtual reality-based system integrated with fmri to study neural mechanisms of action observation-execution: a proof of concept study. *Restor Neurol Neurosci.* 2009;27(3):209-23.
2. Adams CM, Wilson TD. Virtual cerebral ventricular system: an MR-based three-dimensional computer model. *Anat Sci Educ.* 2011 Nov-Dec;4(6):340-7. doi: 10.1002/ase.256. Epub 2011 Oct 4.
3. Alverson DC, Saiki SM Jr, Jacobs J, Saland L, Keep MF, Norenberg J, Baker R, Nakatsu C, Kalishman S, Lindberg M, Wax D, Mowafi M, Summers KL, Holten JR, Greenfield JA, Aalseth E, Nickles D, Sherstyuk A, Haines K, Caudell TP. Distributed interactive virtual environments for collaborative experiential learning and training independent of distance over Internet2. *Stud Health Technol Inform.* 2004;98:7-12.
4. Armstrong JD, Kaiser K, Müller A, Fischbach KF, Merchant N, Strausfeld NJ. Flybrain, an on-line atlas and database of the *Drosophila* nervous system. *Neuron.* 1995 Jul;15(1):17-20.
5. Aschwanden C, Sherstyuk A, Burgess L, Montgomery K. A surgical and fine-motor skills trainer for everyone? Touch and force-feedback in a virtual reality environment for surgical training. *Stud Health Technol Inform.* 2006;119:19-21.
6. Auer LM, Auer DP. Virtual endoscopy for planning and simulation of minimally invasive neurosurgery. *Neurosurgery.* 1998 Sep;43(3):529-37; discussion 537-48.
7. Belfore LA 2nd, Crouch JR, Shen Y, Girtelschmid S, Baydogan E. A software framework for surgical simulation virtual environments. *Stud Health Technol Inform.* 2006;119:46-8.
8. Bellon E, Wauters J, Fernández-Bayó J, Feron M, Verstreken K, Van Cleynenbreugel J, Van den Bosch B, Desmaret M, Marchal G, Suetens P. Using WWW and JAVA for image access and interactive viewing in an integrated PACS. *Med Inform (Lond).* 1997 Oct-Dec;22(4):291-300.
9. Bernardo A, Preul MC, Zabramski JM, Spetzler RF. A three-dimensional interactive virtual dissection model to simulate transpetrous surgical avenues. *Neurosurgery.* 2003 Mar;52(3):499-505; discussion 504-5.
10. Bezgin G, Reid AT, Schubert D, Kötter R. Matching spatial with ontological brain regions using Java tools for visualization, database access, and integrated data analysis. *Neuroinformatics.* 2009 Spring;7(1):7-22.
11. Blackwell M, Nikou C, DiGioia AM, Kanade T. An image overlay system for medical data visualization. *Med Image Anal.* 2000 Mar;4(1):67-72.
12. Brewer DN, Wilson TD, Eagleson R, de Ribaupierre S. Evaluation of Neuroanatomical Training Using a 3D Visual Reality Model. *Stud Health Technol Inform.* 2012;173:85-91.
13. Brown N, Natsupakpong S, Johannsen S, Manjila S, Cai Q, Liberatore V, Cohen AR, Cavusoglu MC. Virtual environment-based training simulator for endoscopic third ventriculostomy. *Stud Health Technol Inform.* 2006;119:73-5.
14. Brown PM, Hamilton NM, Denison AR. A novel 3D stereoscopic anatomy tutorial. *Clin Teach.* 2012 Feb;9(1):50-3.
15. Burke D, Zhou X, Rotty V, Konchada V, Shen Y, Konety B, Sweet R. Web-accessible interactive software of 3D anatomy representing pathophysiological conditions to enhance the patient-consent process for procedures. *Stud Health Technol Inform.* 2011;163:96-8.

16. Carson JP, Ju T, Lu HC, Thaller C, Xu M, Pallas SL, Crair MC, Warren J, Chiu W, Eichele G. A digital atlas to characterize the mouse brain transcriptome. *PLoS Comput Biol*. 2005 Sep;1(4):e4
17. Carvalho E, Marcos A, Santos MY, Espregueira-Mendes J. A prototype for cartographic human body analysis. *IEEE Comput Graph Appl*. 2008 Sep-Oct;28(5):16-21.
18. Cavaşođlu MC, Göktekin TG, Tendick F. GiPSi: a framework for open source/open architecture software development for organ-level surgical simulation. *IEEE Trans Inf Technol Biomed*. 2006 Apr;10(2):312-22.
19. Cemenasco AF, Bianchi CC, Tornincasa S, Bianchi SD. The WEBD project: a research of new methodologies for a distant-learning 3D system prototype. *Dentomaxillofac Radiol*. 2004 Nov;33(6):403-8.
20. Chami G, Phillips R, Ward JW, Bielby MS, Mohsen AM. 3D visualization and open planning platform in virtual fluoroscopy. *Stud Health Technol Inform*. 2007;125:64-6.
21. Chen G, Zhang SX, Tan LW, Liu GJ, Li K, Dong JH. The study of three-dimensional reconstruction of Chinese adult liver. *Surg Radiol Anat*. 2009 Jul;31(6):453-60.
22. Chen J, Yoon I, Bethel W. Interactive, internet delivery of visualization via structured prerendered multiresolution imagery. *IEEE Trans Vis Comput Graph*. 2008 Mar-Apr;14(2):302-12.
23. Chen KC, Glicksman JT, Haase P, Johnson M, Wilson T, Fung K. Introduction of a novel teaching paradigm for head and neck anatomy. *J Otolaryngol Head Neck Surg*. 2010 Aug;39(4):349-55.
24. Chung MS, Kim SY. Three-dimensional image and virtual dissection program of the brain made of Korean cadaver. *Yonsei Med J*. 2000 Jun;41(3):299-303.
25. Clapworthy G, Krokos M, Vasilonikolidakis N. An Internet service for manipulating 3D models of human organs reconstructed from computer tomography and magnetic resonance imaging. *Int J Med Inform*. 1997 Nov;47(1-2):121-4.
26. Coleman J, Savchenko A, Goetsch A, Wang K, Bono P, Littlefield R, Macedonia C. TeleInViVo: a collaborative volume visualization application. *Stud Health Technol Inform*. 1997;39:115-24.
27. Cosić D. An open medical imaging workstation architecture for platform-independent 3-D medical image processing and visualization. *IEEE Trans Inf Technol Biomed*. 1997 Dec;1(4):279-83.
28. Courteille O, Felländer-Tsai L, Hedman L, Kjellin A, Enochsson L, Lindgren G, Fors U. Mixed virtual reality simulation--taking endoscopic simulation one step further. *Stud Health Technol Inform*. 2011;163:144-6.
29. Crossingham JL, Jenkinson J, Woolridge N, Gallinger S, Tait GA, Moulton CA. Interpreting three-dimensional structures from two-dimensional images: a web-based interactive 3D teaching model of surgical liver anatomy. *HPB (Oxford)*. 2009 Sep;11(6):523-8.
30. De Paolis LT, De Mauro A, Raczkowsky J, Aloisio G. Virtual model of the human brain for neurosurgical simulation. *Stud Health Technol Inform*. 2009;150:811-5.
31. De Paolis LT, Pulimeno M, Aloisio G. A virtual interface for interactions with 3D models of the human body. *Stud Health Technol Inform*. 2009;142:68-70.
32. Dech F, Ai Z, Silverstein JC. Manipulation of volumetric patient data in a distributed virtual reality environment. *Stud Health Technol Inform*. 2001;81:119-25.
33. Delaurier A, Burton N, Bennett M, Baldock R, Davidson D, Mohun TJ, Logan MP. The Mouse Limb Anatomy Atlas: an interactive 3D tool for studying embryonic limb patterning. *BMC Dev Biol*. 2008 Sep 15;8:83.
34. Dev P, Heinrichs WL, Youngblood P. CliniSpace: a multiperson 3D online immersive training environment accessible through a browser. *Stud Health Technol Inform*. 2011;163:173-9.
35. Ding A, Zhang D, Xu XG. Training software using virtual-reality technology and pre-calculated effective dose data. *Health Phys*. 2009 May;96(5):594-601.
36. Dirnberger J, Giretzlehner M, Ruhmer M, Haller H, Rodemund C. Modelling human burn injuries in a three-dimensional virtual environment. *Stud Health Technol Inform*. 2003;94:52-8.

37. Dobson HD, Pearl RK, Orsay CP, Rasmussen M, Evenhouse R, Ai Z, Blew G, Dech F, Edison MI, Silverstein JC, Abcarian H. Virtual reality: new method of teaching anorectal and pelvic floor anatomy. *Dis Colon Rectum*. 2003 Mar;46(3):349-52.
38. Donnelly L, Patten D, White P, Finn G. Virtual human dissector as a learning tool for studying cross-sectional anatomy. *Med Teach*. 2009 Jun;31(6):553-5.
39. Doyle MD, Ang CS, Martin DC, Noe A. The visible embryo project: embedded program objects for knowledge access, creation and management through the World Wide Web. *Comput Med Imaging Graph*. 1996 Nov-Dec;20(6):423-31.
40. Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics*. 2009 Feb 3;10:48.
41. Ehrhardt J, Handels H, Wegner T, Strathmann B, Plötz W, Pöppel SJ. An anatomical atlas to support the virtual planning of hip operations. *Stud Health Technol Inform*. 2000;77:1226-30.
42. Eno K, Sundsten JW, Brinkley JF. A multimedia Anatomy Browser incorporating a knowledge base and 3D images. *Proc Annu Symp Comput Appl Med Care*. 1991:727-31.
43. Estevez ME, Lindgren KA, Bergethon PR. A novel three-dimensional tool for teaching human neuroanatomy. *Anat Sci Educ*. 2010 Nov-Dec;3(6):309-17. doi: 10.1002/ase.186. Epub 2010 Oct 11.
44. Feng G, Burton N, Hill B, Davidson D, Kerwin J, Scott M, Lindsay S, Baldock R. JAtlasView: a Java atlas-viewer for browsing biomedical 3D images and atlases. *BMC Bioinformatics*. 2005 Mar 9;6:47.
45. Flores RL, Deluccia N, Grayson BH, Olikier A, McCarthy JG. Creating a virtual surgical atlas of craniofacial procedures: Part I. Three-dimensional digital models of craniofacial deformities. *Plast Reconstr Surg*. 2010 Dec;126(6):2084-92.
46. Foo JL, Lobe T, Winer E. A virtual reality environment for patient data visualization and endoscopic surgical planning. *J Laparoendosc Adv Surg Tech A*. 2009 Apr;19 Suppl 1:S211-7.
47. Foo JL, Martinez-Escobar M, Peloquin C, Lobe T, Winer E. A collaborative interaction and visualization multi-modal environment for surgical planning. *Stud Health Technol Inform*. 2009;142:97-102.
48. Foo JL, Miyano G, Lobe T, Winer E. A framework for interactive examination of automatic segmented tumors in a virtual environment. *Stud Health Technol Inform*. 2008;132:120-2.
49. Freudenberg J, Schiemann T, Tiede U, Höhne KH. Simulation of cardiac excitation patterns in a three-dimensional anatomical heart atlas. *Comput Biol Med*. 2000 Jul;30(4):191-205.
50. Freysinger W, Truppe MJ, Gunkel AR, Thumfart WF. A full 3D-navigation system in a suitcase. *Comput Aided Surg*. 2001;6(2):85-93.
51. Fukuda I, Osanai S, Shirota M, Inamura T, Yanaoka H, Minakawa M, Fukui K. Computer-simulated fluid dynamics of arterial perfusion in extracorporeal circulation: From reality to virtual simulation. *Int J Artif Organs*. 2009 Jun;32(6):362-70.
52. Gacto P, Barrera F, Sicilia-Castro D, Miralles F, Collell M, Leal S, De La Higuera J, Parra C, Gómez-Cía T. A three-dimensional virtual reality model for limb reconstruction in burned patients. *Burns*. 2009 Nov;35(7):1042-6. Epub 2008 Dec 13
53. Garg AX, Norman GR, Eva KW, Spero L, Sharan S. Is there any real virtue of virtual reality?: the minor role of multiple orientations in learning anatomy from computers. *Acad Med*. 2002 Oct;77(10 Suppl):S97-9.
54. Gehrman S, Höhne KH, Linhart W, Pflesser B, Pommert A, Riemer M, Tiede U, Windolf J, Schumacher U, Rueger JM. A novel interactive anatomic atlas of the hand. *Clin Anat*. 2006 Apr;19(3):258-66.
55. Genda E. Making of virtual human body--virtual human body as communication interface. *J Physiol Anthropol Appl Human Sci*. 2005 Jan;24(1):123-7.
56. Gerth VE, Vize PD. A Java tool for dynamic web-based 3D visualization of anatomy and overlapping gene or protein expression patterns. *Bioinformatics*. 2005 Apr 1;21(7):1278-9.
57. Glombitza G, Evers H, Hassfeld S, Engelmann U, Meinzer HP. Virtual surgery in a (tele-)radiology framework. *IEEE Trans Inf Technol Biomed*. 1999 Sep;3(3):186-96.

58. Góes VP, Machado LS, Cabral MC, Leitão RB, Lopes RD, Zuffo MK. Interactive stereoscopic full-color direct volume visualization for virtual reality applications in medicine. *Stud Health Technol Inform.* 2001;81:161-7.
59. Golland P, Kikinis R, Halle M, Umans C, Grimson WE, Shenton ME, Richolt JA. AnatomyBrowser: A novel approach to visualization and integration of medical information. *Comput Aided Surg.* 1999;4(3):129-43.
60. Grottke O, Ntoubas A, Ullrich S, Liao W, Fried E, Prescher A, Deserno TM, Kuhlen T, Rossaint R. Virtual reality-based simulator for training in regional anaesthesia. *Br J Anaesth.* 2009 Oct;103(4):594-600. Epub 2009 Aug 25.
61. Grunewald M, Heckemann RA, Wagner M, Bautz WA, Greess H. ELERA: a WWW application for evaluating and developing radiologic skills and knowledge. *Acad Radiol.* 2004 Dec;11(12):1381-8.
62. Guo YL, Heng PA, Zhang SX, Liu ZJ, Tan LW, Li QY, Qiu MG, Li K, Fan HQ, Wang YS, Tang ZS. Thin sectional anatomy, three-dimensional reconstruction and visualization of the heart from the Chinese Visible Human. *Surg Radiol Anat.* 2005 Apr;27(2):113-8.
63. Gustafson C, Bug WJ, Nissanov J. NeuroTerrain--a client-server system for browsing 3D biomedical image data sets. *BMC Bioinformatics.* 2007 Feb 5;8:40.
64. Handels H, Hacker S. A framework for representation and visualization of 3D shape variability of organs in an interactive anatomical atlas. *Methods Inf Med.* 2009;48(3):272-81. Epub 2009 Mar 31.
65. Heng PA, Xie Y, Wang X, Chui YP, Wong TT. Virtual acupuncture human based on chinese visible human dataset. *Stud Health Technol Inform.* 2006;119:194-7.
66. Heng PA, Zhang SX, Xie YM, Wong TT, Chui YP, Cheng CY. Photorealistic virtual anatomy based on Chinese Visible Human data. *Clin Anat.* 2006 Apr;19(3):232-9.
67. Hernes TA, Ommedal S, Lie T, Lindseth F, Langø T, Unsgaard G. Stereoscopic navigation-controlled display of preoperative MRI and intraoperative 3D ultrasound in planning and guidance of neurosurgery: new technology for minimally invasive image-guided surgery approaches. *Minim Invasive Neurosurg.* 2003 Jun;46(3):129-37.
68. Hippalgaonkar NR, Sider AD, Hamza-Lup FG, Santhanam AP, Jaganathan B, Imielinska C, Rolland JP. Generating classes of 3D virtual mandibles for AR-based medical simulation. *Simul Healthc.* 2008 Summer;3(2):103-10.
69. Hoffman H, Murray M. Anatomic VisualizeR: realizing the vision of a VR-based learning environment. *Stud Health Technol Inform.* 1999;62:134-40.
70. Hoffman HM, Murray M, Irwin AE, McCracken T. Developing a virtual reality-multimedia system for anatomy training. *Stud Health Technol Inform.* 1996;29:204-10.
71. Holubar SD, Hassinger JP, Dozois EJ, Camp JC, Farley DR, Fidler JL, Pawlina W, Robb RA. Virtual pelvic anatomy and surgery simulator: an innovative tool for teaching pelvic surgical anatomy. *Stud Health Technol Inform.* 2009;142:122-4.
72. Huang S, Baimouratov R, Xiao P, Ananthasubramaniam A, Nowinski WL. A medical imaging and visualization toolkit in Java. *J Digit Imaging.* 2006 Mar;19(1):17-29.
73. Hwang JW, Lee JM, Kim IY, Song IH, Lee YH, Kim S.. A PC-based high-quality and interactive virtual endoscopy navigating system using 3D texture based volume rendering. *Comput Methods Programs Biomed.* 2003 May;71(1):77-84.
74. Ichikawa K. A modeling environment with three-dimensional morphology, A-Cell-3D, and Ca²⁺ dynamics in a spine. *Neuroinformatics.* 2005;3(1):49-64.
75. Jiménez-Lozano N, Segura J, Macías JR, Vega J, Carazo JM. aGEM: an integrative system for analyzing spatial-temporal gene-expression information. *Bioinformatics.* 2009 Oct 1;25(19):2566-72. Epub 2009 Jul 9.
76. Kapakin S. The claustrum: three-dimensional reconstruction, photorealistic imaging, and stereotactic approach. *Folia Morphol (Warsz).* 2011 Nov;70(4):228-34.

77. Kaspar M, Dech F, Parsad NM, Silverstein JC. Web-based stereoscopic visualization for the global anatomy classroom. *Stud Health Technol Inform.* 2011;163:264-70.
78. Kaye JM, Primiano FP Jr, Metaxas DN. A three-dimensional virtual environment for modeling mechanical cardiopulmonary interactions. *Med Image Anal.* 1998 Jun;2(2):169-95.
79. Kerr J, Ratiu P, Sellberg M. Volume rendering of visible human data for an anatomical virtual environment. *Stud Health Technol Inform.* 1996;29:352-70.
80. Kesavadas T, Joshi D, Mayrose J, Chugh K. A virtual environment for esophageal intubation training. *Stud Health Technol Inform.* 2002;85:221-7.
81. Kim DO, Kang HS, Kim JH, Min BG. Virtual anatomy and movement of lower extremities using virtual reality modeling language. *J Digit Imaging.* 2000 May;13(2 Suppl 1):238-40.
82. Kim J, Choi C, De S, Srinivasan MA. Virtual surgery simulation for medical training using multi-resolution organ models. *Int J Med Robot.* 2007 Jun;3(2):149-58.
83. Kling-Petersen T, Pascher R, Rydmark M. 3D-brain 2.0--narrowing the gap between personal computers and high end workstations. *Stud Health Technol Inform.* 1998;50:234-9.
84. Kling-Petersen T, Rydmark M. The BRAIN project: an interactive learning tool using desktop virtual reality on personal computers. *Stud Health Technol Inform.* 1997;39:529-38.
85. Knudsen BE, Campbell G, Kennedy A, Amann J, Beiko DT, Watterson JD, Chew BH, Denstedt JD, Pautler SE. Design of functional simulation of renal cancer in virtual reality environments. *Urology.* 2005 Oct;66(4):732-5.
86. Kobayashi M, Nakajima T, Mori A, Tanaka D, Fujino T, Chiyokura H. Three-dimensional computer graphics for surgical procedure learning: Web three-dimensional application for cleft lip repair. *Cleft Palate Craniofac J.* 2006 May;43(3):266-71.
87. Kockro RA, Hwang PY. Virtual temporal bone: an interactive 3-dimensional learning aid for cranial base surgery. *Neurosurgery.* 2009 May;64(5 Suppl 2):216-29; discussion 229-30.
88. Kockro RA, Stadie A, Schwandt E, Reisch R, Charalampaki C, Ng I, Yeo TT, Hwang P, Serra L, Perneckzy A. A collaborative virtual reality environment for neurosurgical planning and training. *Neurosurgery.* 2007 Nov;61(5 Suppl 2):379-91; discussion 391.
89. Koehring A, Foo JL, Miyano G, Lobe T, Winer E. A framework for interactive visualization of digital medical images. *J Laparoendosc Adv Surg Tech A.* 2008 Oct;18(5):697-706.
90. Konchada V, Shen Y, Burke D, Argun OB, Weinhaus A, Erdman AG, Sweet RM. The Minnesota pelvic trainer: a hybrid VR/physical pelvis for providing virtual mentorship. *Stud Health Technol Inform.* 2011;163:280-2.
91. Koning AH, Rousian M, Verwoerd-Dikkeboom CM, Goedknecht L, Steegers EA, van der Spek PJ. V-scope: design and implementation of an immersive and desktop virtual reality volume visualization system. *Stud Health Technol Inform.* 2009;142:136-8.
92. Kooy HM, Cormack RA, Mathiowitz G, Tempany C, D'Amico AV. A software system for interventional magnetic resonance image-guided prostate brachytherapy. *Comput Aided Surg.* 2000;5(6):401-13.
93. Korocsec D, Holobar A, Divjak M, Zazula D. Building interactive virtual environments for simulated training in medicine using VRML and Java/JavaScript. *Comput Methods Programs Biomed.* 2005 Dec;80 Suppl 1:S61-70.
94. Krsek P, Spanel M, Svub M, Stancel V, Siler O, Sára V. Consultation virtual collaborative environment for 3D medicine. *Conf Proc IEEE Eng Med Biol Soc.* 2008;2008:775-8.
95. Kurylas AE, Rohlfing T, Kroczyk S, Jenett A, Homberg U. Standardized atlas of the brain of the desert locust, *Schistocerca gregaria*. *Cell Tissue Res.* 2008 Jul;333(1):125-45. Epub 2008 May 27.
96. Lamperti A, Sodicoff M. Computer-based neuroanatomy laboratory for medical students. *Anat Rec.* 1997 Nov;249(3):422-8.
97. Iapan I, Vranjes Z, Prgomet D, Lukinović J. Application of advanced virtual reality and 3D computer assisted technologies in tele-3D-computer assisted surgery in rhinology. *Coll Antropol.* 2008

- Mar;32(1):217-9.
98. Lau C, Ng L, Thompson C, Pathak S, Kuan L, Jones A, Hawrylycz M. Exploration and visualization of gene expression with neuroanatomy in the adult mouse brain. *BMC Bioinformatics*. 2008 Mar 18;9:153.
 99. Lee JS, Jani AB, Pelizzari CA, Haraf DJ, Vokes EE, Weichselbaum RR, Chen GT. Volumetric visualization of head and neck CT data for treatment planning. *Int J Radiat Oncol Biol Phys*. 1999 Jun 1;44(3):693-703.
 100. Lemole GM Jr, Banerjee PP, Luciano C, Neckrysh S, Charbel FT. Virtual reality in neurosurgical education: part-task ventriculostomy simulation with dynamic visual and haptic feedback. Lemole GM Jr, Banerjee PP, Luciano C, Neckrysh S, Charbel FT.
 101. Liem SL. [An interactive three-dimensional model of the human body]. *Ned Tijdschr Tandheelkd*. 2009 Jan;116(1):40-2. Dutch.
 102. Liu S, Liao W, Yu Q, Cheng X, Dai N, Zhang X. 51. [The development of a system for 3D reconstruction from DICOM data and collaborative visualization]. *Sheng Wu Yi Xue Gong Cheng Xue Za Zhi*. 2007 Oct;24(5):1152-6.
 103. Ljung P, Winskog C, Persson A, Lundström C, Ynnerman A. Full body virtual autopsies using a state-of-the-art volume rendering pipeline. *IEEE Trans Vis Comput Graph*. 2006 Sep-Oct;12(5):869-76.
 104. López-Cano M, Rodríguez-Navarro J, Rodríguez-Baeza A, Armengol-Carrasco M, Susín A. A real-time dynamic 3D model of the human inguinal region for surgical education. *Comput Biol Med*. 2007 Sep;37(9):1321-6. Epub 2007 Feb 12.
 105. Luo J, Ashton-Miller JA, DeLancey JO. A model patient: Female pelvic anatomy can be viewed in diverse 3-dimensional images with a new interactive tool. *Am J Obstet Gynecol*. 2011 Oct;205(4):391.e1-2.
 106. Ma M, Zheng H, Lallie H. Virtual reality and 3D animation in forensic visualization. *J Forensic Sci*. 2010 Sep;55(5):1227-31.
 107. Mackenzie-Graham AJ, Lee EF, Dinov ID, Yuan H, Jacobs RE, Toga AW. Multimodal, multidimensional models of mouse brain. *Epilepsia*. 2007;48 Suppl 4:75-81.
 108. Mahmoudi SE, Akhondi-Asl A, Rahmani R, Faghih-Roohi S, Taimouri V, Sabouri A, Soltanian-Zadeh H. Web-based interactive 2D/3D medical image processing and visualization software. *Comput Methods Programs Biomed*. 2010 May;98(2):172-82.
 109. Manyak MJ, Santangelo K, Hahn J, Kaufman R, Carleton T, Hua XC, Walsh RJ. Virtual reality surgical simulation for lower urinary tract endoscopy and procedures. *J Endourol*. 2002 Apr;16(3):185-90.
 110. Mastrangelo MJ Jr, Hoskins JD, Nicholls M, Munch LC, Johnston TD, Reddy KS, Ranjan D, Witzke WO, Park A. Using immersive VR as a tool for preoperative planning for minimally invasive donor nephrectomy. *Stud Health Technol Inform*. 2001;81:298-304.
 111. McEnery KW, Roth SM, Kelley LK, Hirsch KR, Menton DN, Kelly EA. A method for interactive medical instruction utilizing the World Wide Web. *Proc Annu Symp Comput Appl Med Care*. 1995:502-7.
 112. McKeough DM, Mattern-Baxter K, Barakatt E. Effectiveness of a computer-aided neuroanatomy program for entry-level physical therapy students: anatomy and clinical examination of the dorsal column-medial lemniscal system. *J Allied Health*. 2010 Fall;39(3):156-64.
 113. Michel MS, Knoll T, Köhrmann KU, Alken P. The URO Mentor: development and evaluation of a new computer-based interactive training system for virtual life-like simulation of diagnostic and therapeutic endourological procedures. *BJU Int*. 2002 Feb;89(3):174-7.
 114. Mitsuhashi N, Fujieda K, Tamura T, Kawamoto S, Takagi T, Okubo K. BodyParts3D: 3D structure database for anatomical concepts. *Nucleic Acids Res*. 2009 Jan;37(Database issue)
 115. Modha DS. A conceptual cortical surface atlas. *PLoS One*. 2009 Jun 2;4(6):e5693.
 116. Mori S, van Zijl P. Human white matter atlas. *Am J Psychiatry*. 2007 Jul;164(7):1005.
 117. Morris DP, Van Wijhe RG. Cholesteatoma in three dimensions: a teaching tool and an aid to improved

- pre-operative consent. *J Laryngol Otol.* 2010 Feb;124(2):126-31.
118. Mowery D, Clayton M, Hu J, Schleyer TK. Tooth Atlas 3D, version 6.3.0. *J Dent Educ.* 2010 Nov;74(11):1261-4
 119. Nakao M, Kuroda T, Komori M, Oyama H, Komeda M. Physics-based preoperative approach planning using hybrid virtual bodies. *Stud Health Technol Inform.* 2004;98:262-4.
 120. Narayan S, Sensharma D, Santori EM, Lee AA, Sabherwal A, Toga AW. Animated visualization of a high resolution color three dimensional digital computer model of the whole human head. *Int J Biomed Comput.* 1993 Jan;32(1):7-17.
 121. Natarajan K, Claridge E, Davis DN. The symbolic atlas of the brain: handling non-visual information associated with neuroanatomy. *Med Inform (Lond).* 1996 Jan-Mar;21(1):1-21
 122. Ng YR, Shiffman S, Brosnan TJ, Links JM, Beach LS, Judge NS, Xu Y, Kelkar UV, Reiss AL. BrainImageJ: a Java-based framework for interoperability in neuroscience, with specific application to neuroimaging. *J Am Med Inform Assoc.* 2001 Sep-Oct;8(5):431-42.
 123. Nguyen N, Wilson TD. A head in virtual reality: development of a dynamic head and neck model. *Anat Sci Educ.* 2009 Nov-Dec;2(6):294-301.
 124. Nieder GL, Nagy F, Wagner LA. Preserving and sharing examples of anatomical variation and developmental anomalies via photorealistic virtual reality. *Anat Rec B New Anat.* 2004 Jan;276(1):15-8.
 125. Nieder GL, Scott JN, Anderson MD. Using QuickTime virtual reality objects in computer-assisted instruction of gross anatomy: Yorick--the VR Skull. *Clin Anat.* 2000;13(4):287-93.
 126. Nowinski WL, Chua BC, Qian GY, Nowinska NG. The human brain in 1700 pieces: design and development of a three-dimensional, interactive and reference atlas. *J Neurosci Methods.* 2012 Feb 15;204(1):44-60.
 127. Nowinski WL, Thirunavuukarasuu A, Volkau I, Baimuratov R, Hu Q, Aziz A, Huang S. Informatics in Radiology (infoRAD): three-dimensional atlas of the brain anatomy and vasculature. *Radiographics.* 2005 Jan-Feb;25(1):263-71.
 128. Nowinski WL, Thirunavuukarasuu A, Volkau I, Marchenko Y, Aminah B, Gelas A, Huang S, Lee LC, Liu J, Ng TT, Nowinska NG, Qian GY, Puspitasari F, Runge VM. A new presentation and exploration of human cerebral vasculature correlated with surface and sectional neuroanatomy. *Anat Sci Educ.* 2009 Jan-Feb;2(1):24-33.
 129. Nowinski WL, Thirunavuukarasuu A, Volkau I, Marchenko Y, Aminah B, Puspitasari F, Runge VM. A three-dimensional interactive atlas of cerebral arterial variants. *Neuroinformatics.* 2009 Dec;7(4):255-64.
 130. Nowinski WL. Towards construction of an ideal stereotactic brain atlas. *Acta Neurochir (Wien).* 2008 Jan;150(1):1-13
 131. Numminen K, Sipilä O, Mäkisalo H. Preoperative hepatic 3D models: virtual liver resection using three-dimensional imaging technique. *Eur J Radiol.* 2005 Nov;56(2):179-84.
 132. Oyama H, Wakao F, Takahira Y. The clinical advantages of editable real-time volume rendering in a medical virtual environment: VolMed. *Stud Health Technol Inform.* 1998;50:341-5.
 133. Palombi O, Pihuit A, Cani MP. 3D Modeling of branching vessels from anatomical sketches: towards a new interactive teaching of anatomy: Interactive virtual blackboard. *Surg Radiol Anat.* 2011 Sep;33(7):631-6. Epub 2011 May 27.
 134. Panchaphongsaphak B, Burgkart R, Riener R. Three-dimensional touch interface for medical education. *IEEE Trans Inf Technol Biomed.* 2007 May;11(3):251-63.
 135. Parikh M, Rasmussen M, Brubaker L, Salomon C, Sakamoto K, Evenhouse R, Ai Z, Damaser MS. Three dimensional virtual reality model of the normal female pelvic floor. *Ann Biomed Eng.* 2004 Feb;32(2):292-6.
 136. Pearl RK, Evenhouse R, Rasmussen M, Dech F, Silverstein JC, Prokasy S, Panko WB. The Virtual Pelvic Floor, a tele-immersive educational environment. *Proc AMIA Symp.* 1999:345-8.
 137. Petersson H, Sinkvist D, Wang C, Smedby O. Web-based interactive 3D visualization as a tool for

- improved anatomy learning. *Anat Sci Educ.* 2009 Mar-Apr;2(2):61-8.
138. Pflessner B, Petersik A, Tiede U, Höhne KH, Leuwer R. Volume cutting for virtual petrous bone surgery. *Comput Aided Surg.* 2002;7(2):74-83.
 139. Qatarnah SM, Kiricuta IC, Brahme A, Tiede U, Lind BK. Three-dimensional atlas of lymph node topography based on the visible human data set. *Anat Rec B New Anat.* 2006 May;289(3):98-111.
 140. Reeves GM, Lambert S, Wells AP. Development of a highly detailed virtual model eye. *Ophthalmic Surg Lasers Imaging.* 2006 Mar-Apr;37(2):165-8.
 141. Reinig K, Lee C, Rubinstein D, Bagur M, Spitzer V. The United States military's thigh trauma simulator. *Clin Orthop Relat Res.* 2006 Jan;442:45-56.
 142. Robb RA, Hanson DP. A software system for interactive and quantitative visualization of multidimensional biomedical images. *Australas Phys Eng Sci Med.* 1991 Mar;14(1):9-30.
 143. Rudin JL. A.D.A.M. (Animated Dissection of Anatomy for Medicine) comprehensive--a computerized human anatomy program. *Compend Contin Educ Dent.* 1996 Apr;17(4):344-6, 348, 350
 144. Ruffins SW, Martin M, Keough L, Truong S, Fraser SE, Jacobs RE, Lansford R. Digital three-dimensional atlas of quail development using high-resolution MRI. *ScientificWorldJournal.* 2007 May 11;7:592-604.
 145. Ruthenbeck GS, Owen H, Reynolds KJ. A virtual reality throat examination simulation. *Stud Health Technol Inform.* 2008;132:433-5.
 146. Sadikot AF, Chakravarty MM, Bertrand G, Rymar VV, Al-Subaie F, Collins DL. Creation of Computerized 3D MRI-Integrated Atlases of the Human Basal Ganglia and Thalamus. *Front Syst Neurosci.* 2011;5:71. Epub 2011 Sep 6.
 147. Schleich JM, Dillenseger JL, Houyel L, Almange C, Anderson RH. A new dynamic 3D virtual methodology for teaching the mechanics of atrial septation as seen in the human heart. *Anat Sci Educ.* 2009 Mar-Apr;2(2):69-77.
 148. Schubert R, Bomans M, Höhne KH, Pommert A, Riemer M, Schiemann T, Tiede U, Lierse W. A new method for representing the human anatomy. *Comput Med Imaging Graph.* 1993 Jul-Oct;17(4-5):243-9.
 149. Schubert R, Höhne KH, Pommert A, Riemer M, Schiemann T, Tiede U, Lierse W. A new method for practicing exploration, dissection, and simulation with a complete computerized three-dimensional model of the brain and skull. *Acta Anat (Basel).* 1994;150(1):69-74.
 150. Seemann MD, Luboldt W, Haferkamp C, Bode A, Schäfer J, Allen CM, Dammann F, Plinkert P, Claussen CD. [Hybrid 3D visualization and virtual endoscopy in cochlear implants]. *Rofo.* 2000 Mar;172(3):238-43.
 151. Seipel S, Wagner IV, Koch S, Schneider W. A virtual reality environment for enhanced oral implantology. *Medinfo.* 1995;8 Pt 2:1710.
 152. Seitel M, Maier-Hein L, Rietdorf U, Nikoloff S, Seitel A, Franz A, Kenngott H, Karck M, De Simone R, Wolf I, Meinzer HP. Towards a mixed reality environment for preoperative planning of cardiac surgery. *Stud Health Technol Inform.* 2009;142:307-9.
 153. Seitel M, Maier-Hein L, Seitel A, Franz AM, Kenngott H, De Simone R, Wolf I, Meinzer HP. RepliExplore: coupling physical and virtual anatomy models. *Int J Comput Assist Radiol Surg.* 2009 Sep;4(5):417-24
 154. Semeraro F, Frisoli A, Bergamasco M, Cerchiari EL. Virtual reality enhanced mannequin (VREM) that is well received by resuscitation experts. *Resuscitation.* 2009 Apr;80(4):489-92. Epub 2009 Feb 8.
 155. Sergovich A, Johnson M, Wilson TD. Explorable three-dimensional digital model of the female pelvis, pelvic contents, and perineum for anatomical education. *Anat Sci Educ.* 2010 May-Jun;3(3):127-33.
 156. Silverstein JC, Amine M, Dech F, Jurek P, Pires I, Tsirlina V, Walsh C. Web-based viewer for systematic combination of anatomy and nomenclature. *Stud Health Technol Inform.* 2006;119:518-22.
 157. Silverstein JC, Dech F, Edison M, Jurek P, Helton WS, Espat NJ. Virtual reality: immersive hepatic surgery educational environment. *Surgery.* 2002 Aug;132(2):274-7.

158. Silverstein JC, Dech F, Kouchoukos PL. Enhancing radiological volumes with symbolic anatomy using image fusion and collaborative virtual reality. *Stud Health Technol Inform.* 2004;98:347-52.
159. Silverstein JC, Walsh C, Dech F, Olson E, Papka ME, Parsad N, Stevens R. Immersive virtual anatomy course using a cluster of volume visualization machines and passive stereo. *Stud Health Technol Inform.* 2007;125:439-44.
160. Smith DM, Aston SJ, Cutting CB, Olikier A, Weinzwieg J. Designing a virtual reality model for aesthetic surgery. *Plast Reconstr Surg.* 2005 Sep;116(3):893-7.
161. Smith DM, Olikier A, Carter CR, Kirov M, McCarthy JG, Cutting CB. A virtual reality atlas of craniofacial anatomy. *Plast Reconstr Surg.* 2007 Nov;120(6):1641-6.
162. Soh J, Turinsky AL, Trinh QM, Chang J, Sabhaney A, Dong X, Gordon PM, Janzen RP, Hau D, Xia J, Wishart DS, Sensen CW. Spatiotemporal integration of molecular and anatomical data in virtual reality using semantic mapping. *Int J Nanomedicine.* 2009;4:79-89. Epub 2009 Apr 1.
163. Sora MC, Jilavu R, Matusz P. Computer aided three-dimensional reconstruction and modeling of the pelvis, by using plastinated cross sections, as a powerful tool for morphological investigations. *Surg Radiol Anat.* 2011 Aug 23. [
164. Sorensen MS, Mosegaard J, Trier P. The visible ear simulator: a public PC application for GPU-accelerated haptic 3D simulation of ear surgery based on the visible ear data. *Otol Neurotol.* 2009 Jun;30(4):484-7.
165. Sørensen TS, Therkildsen SV, Makowski P, Knudsen JL, Pedersen EM. A new virtual reality approach for planning of cardiac interventions. *Artif Intell Med.* 2001 Jun;22(3):193-214.
166. Stefanich L, Cruz-Neira C. A virtual surgical simulator for the lower limbs. *Biomed Sci Instrum.* 1999;35:141-5.
167. Stephens B, Temkin B, Krummel TM, Heinrichs WL. Virtual body structures: a 3D structure development tool from visible human data. *Stud Health Technol Inform.* 2000;70:323-6.
168. Stredney D, Ebert DS, Svakhine N, Bryan J, Sessanna D, Wiet GJ. Emphatic, interactive volume rendering to support variance in user expertise. *Stud Health Technol Inform.* 2005;111:526-31.
169. Stredney D, Hittle B, Chen CM, Kerwin T, Bratasz A, Manivannan N, Powell K. Virtual simulation of mouse anatomy and procedural techniques. *Stud Health Technol Inform.* 2012;173:500-5.
170. Suwelack S, Maier S, Unterhinninghofen R, Dillmann R. Web-based interactive volume rendering. *Stud Health Technol Inform.* 2011;163:635-7.
171. Tanaka D, Kobayashi M, Chiyokura H, Nakajima T, Fujino T. Web-based educational tool for cleft lip repair using XVL. *Stud Health Technol Inform.* 2001;81:485-91.
172. Temkin B, Acosta E, Malvankar A, Vaidyanath S. An interactive three-dimensional virtual body structures system for anatomical training over the internet. *Clin Anat.* 2006 Apr;19(3):267-74.
173. Tikhonova A, Correa CD, Ma KL. Visualization by proxy: a novel framework for deferred interaction with volume data. *IEEE Trans Vis Comput Graph.* 2010 Nov-Dec;16(6):1551-9.
174. Toh MY, Falk RB, Main JS. Interactive brain atlas with the Visible Human Project data: development methods and techniques. *Radiographics.* 1996 Sep;16(5):1201-6.
175. Tomandl BF, Hastreiter P, Eberhardt KE, Rezk-Salama C, Naraghi R, Greess H, Nissen U, Huk WJ. Virtual labyrinthoscopy: visualization of the inner ear with interactive direct volume rendering. *Radiographics.* 2000 Mar-Apr;20(2):547-58.
176. Trelease RB, Rosset A. Transforming clinical imaging data for virtual reality learning objects. *Anat Sci Educ.* 2008 Mar-Apr;1(2):50-5.
177. Trelease RB. The virtual anatomy practical: a stereoscopic 3D interactive multimedia computer examination program. *Clin Anat.* 1998;11(2):89-94.
178. Trelease, RB. Toward virtual anatomy: a stereoscopic 3-D interactive multimedia computer program for cranial osteology. *Clin Anat.* 1996;9(4):269-72.
179. Turinsky AL, Fanea E, Trinh Q, Dong X, Stromer JN, Shu X, Wat S, Hallgrímsson B, Hill JW, Edwards

- C, Grosenick B, Yajima M, Sensen CW. Integration of genomic and medical data into a 3D atlas of human anatomy. *Stud Health Technol Inform.* 2008;132:526-31.
180. Turinsky AL, Fanea E, Trinh Q, Wat S, Hallgrímsson B, Dong X, Shu X, Stromer JN, Hill JW, Edwards C, Grosenick B, Yajima M, Sensen CW. CAVEman: Standardized anatomical context for biomedical data mapping. *Anat Sci Educ.* 2008 Jan;1(1):10-8.
 181. Umehara T, Matsuda T, Chiyokura H, Kobayashi M. Human body textbook with three-dimensional illustrations. *Stud Health Technol Inform.* 1996;29:694-702.
 182. Van Sint Jan S, Crudele M, Gashegu J, Feipel V, Poulet P, Salvia P, Hilal I, Sholukha V, Louryan S, Rooze M. Development of multimedia learning modules for teaching human anatomy: application to osteology and functional anatomy. *Anat Rec B New Anat.* 2003 May;272(1):98-106.
 183. Vartanian AJ, Holcomb J, Ai Z, Rasmussen M, Tardy ME, Thomas JR. The virtual nose: a 3-dimensional virtual reality model of the human nose. *Arch Facial Plast Surg.* 2004 Sep-Oct;6(5):328-33.
 184. Vazquez-Duchêne MD, Kraemer P, Saugnier N, Levy B, Mine S, Jeanmaire C, Freis O, Pauly G, Rathjens A. SkinExplorer(TM) : a new high-tech platform to interact inside the skin by immersive virtual 3D cutaneous environment. *Skin Res Technol.* 2012 Jan 24.
 185. Vergara VM, Panaiotis, Kingsley D, Alverson DC, Godsmith T, Xia S, Caudell TP. The use of virtual reality simulation of head trauma in a surgical boot camp. *Stud Health Technol Inform.* 2009;142:395-7.
 186. Verwoerd-Dikkeboom CM, Koning AH, Groenenberg IA, Smit BJ, Brezinka C, Van Der Spek PJ, Steegers EA. Using virtual reality for evaluation of fetal ambiguous genitalia. *Ultrasound Obstet Gynecol.* 2008 Sep;32(4):510-4.
 187. Ward BM, Charissis V, Rowley D, Anderson P, Brady L. An evaluation of prototype VR medical training environment: applied surgical anatomy training for malignant breast disease. *Stud Health Technol Inform.* 2008;132:550-5.
 188. Warrick PA, Funnell WR. A VRML-based anatomical visualization tool for medical education. *IEEE Trans Inf Technol Biomed.* 1998 Jun;2(2):55-61.
 189. Welch GF, Sonnenwald DH, Fuchs H, Cairns B, Mayer-Patel K, Söderholm HM, Yang R, State A, Towles H, Ilie A, Ampalam MK, Krishnan S, Noel V, Noland M, Manning JE. 3D medical collaboration technology to enhance emergency healthcare. *J Biomed Discov Collab.* 2009 Apr 19;4:4.
 190. Wiet GJ, Yagel R, Stredney D, Schmalbrock P, Sessanna DJ, Kurzion Y, Rosenberg L, Levin M, Martin K. A volumetric approach to virtual simulation of functional endoscopic sinus surgery. *Stud Health Technol Inform.* 1997;39:167-79.
 191. Wilkinson T. Primal pictures anatomy teaching resources: 3D anatomy software and 3D real-time viewer. *J Anat.* 2011 Oct 17
 192. Wong BA, Rosse C, Brinkley JF. Semi-automatic scene generation using the Digital Anatomist Foundational Model. *Proc AMIA Symp.* 1999:637-41.
 193. Xia J, Ip HH, Samman N, Wang D, Kot CS, Yeung RW, Tideman H. Computer-assisted three-dimensional surgical planning and simulation: 3D virtual osteotomy. *Int J Oral Maxillofac Surg.* 2000 Feb;29(1):11-7.
 194. Xiao M, Soh J, Meruvia-Pastor O, Osborn D, Lam N, Hallgrímsson B, Sensen CW. An efficient virtual dissection tool to create generic models for anatomical atlases. *Stud Health Technol Inform.* 2009;142:426-8.
 195. Yeung JC, Fung K, Wilson TD. Development of a computer-assisted cranial nerve simulation from the visible human dataset. *Anat Sci Educ.* 2011 Mar-Apr;4(2):92-7
 196. Yip GW, Rajendran K. SnapAnatomy, a computer-based interactive tool for independent learning of human anatomy. *J Vis Commun Med.* 2008 Jun;31(2):46-50.
 197. Yu L, Svetachov P, Isenberg P, Everts MH, Isenberg T. FI3D: direct-touch interaction for the exploration of 3D scientific visualization spaces. *IEEE Trans Vis Comput Graph.* 2010 Nov-Dec;16(6):1613-22.
 198. Yu ZY, Zheng SS, Chen LT, He XQ, Wang JJ. Dynamic concision for three-dimensional reconstruction

- of human organ built with virtual reality modelling language (VRML). *J Zhejiang Univ Sci B*. 2005 Jul;6(7):611-6.
- 199.** Zary N, Johnson G, Boberg J, Fors UG. Development, implementation and pilot evaluation of a Web-based Virtual Patient Case Simulation environment--Web-SP. *BMC Med Educ*. 2006 Feb 21;6:10.
- 200.** Zhang Q, Eagleson R, Peters TM. Dynamic real-time 4D cardiac MDCT image display using GPU-accelerated volume rendering. *Comput Med Imaging Graph*. 2009 Sep;33(6):461-76. Epub 2009 May 21.

APPENDIX B

Biolucida Client/Server Commands and Scene Access API

The following tables describe both commands which can be issued by a client and processed by the Biolucida server, as well as the API methods which can be called by a novel Representation to access and manipulate a 3D scene.

Client to Server Commands

| Command | Description | Parameters |
|---------------------------------|--|---|
| <i>Scene Authoring Commands</i> | | |
| addInstances | Adds instances to a Representation | The unique IDs of the concepts to be inserted, along with the Representation ID of which the instances are to be inserted |
| changeInstancePosition | Changes the position of an Entity or contextual object within a Representation | The instance ID, and the new position of the instance, in x,y,z format |
| changeInstanceColor | Sets the diffuse color of an Entity to a new value over a period of time | The instance ID, the desired value in RGB format, 0.0-1.0, and the period of time for the change to take place |
| setInstanceColor | Instantly sets the diffuse color of an Entity or contextual object to a new value | The instance ID, and the desired value in RGB format, 0.0-1.0 |
| changeInstanceRotation | Changes the rotation of an Entity or contextual object over a period of time | The instance ID, the new desired rotation value for the Entity, in x, y, z, radians format, and the period of time in which the change is to take place |
| setInstanceRotation | Instantly sets the rotation of an Entity to a new value | The instance ID, the new desired rotation value for the Entity, in x, y, z, radians format |
| changeInstanceTransparency | Sets the transparency of an Entity or contextual object to a new value, occurring over a period of time | The instance ID, the new transparency value, from 0.0 to 1.0, and the period of time in which the change is to take place |
| setInstanceTransparency | Instantly sets the transparency of an Entity or contextual object to a new value | The instance ID, and the new transparency value, from 0.0 to 1.0 |
| setInstanceMaterial | Sets the Material object of an Entity to a new value | The instance ID, and the instance ID, and the serialized the Material object |
| updateSchemaSet | | The Representation ID, and the serialized Schema Set object |
| uploadResource | Uploads a resource to be utilized in a Representation, such as a sound file, texture, movie, or contextual model | The file path of the resource to be uploaded, and the integer code for the type of resource |
| setGlobalInstanceScale | Sets the scale used for all Entities in a given scene/Representation | The Representation ID, and new scale value in x,y,z format |
| setInstanceScale | Sets the scale factor used for a given Entity | The Representation ID, instance ID, and the new scale value in x,y,z format |
| postContextualContent | Uploads a 3D model resource for use as | The Representation ID for which |

| | | |
|--|--|---|
| | textual content in the scene, downloadable by other viewers of the scene. | the contextual model is destined. All other data transfer is handled by a separate upload process. |
| <i>Dynamic Content Authoring Commands</i> | | |
| updateCameraAnimation-Metadata | Updates the name and description of the Lesson object or camera animation. | The Representation ID, serialized Item Profile describing a “lesson” or camera animation |
| setCameraAction | Sets the value of a given action to the desired object within an animation | The Representation ID, animation index, sequence index, action index, and the serialized action |
| updateCameraActionMetadata | Updates the name and description of the action found within a camera animation or Lesson object. | The Representation ID, animation index, sequence index, action index, and the serialized Item Profile object describing the action. |
| addCameraSequence | Inserts a new Sequence into a given Lesson or camera animation. | The Representation ID, animation index, and the serialized Sequence object to be inserted |
| updateCameraSequence-Metadata | Updates the name and description associated with a given Sequence object, found within a Lesson or camera animation. | The Representation ID, index of the animation, index of the Sequence, and the serialized Item Profile object. |
| addCameraAction | Adds an action to a given camera animation or Lesson | The Representation ID, index of the animation, index of the Sequence, and serialized Action |
| addCameraAnimation | Adds an entire Animation to the Representation | The Representation ID, and the serialized Animation object |
| removeCameraAnimation | Removes an Animation from the Representation | The Representation ID, and the index of the animation to be removed |
| removeCameraSequence | Removes a sequence from an Animation within the Representation | The Representation ID, the animation index, and the sequence index |
| removeCameraAction | Removes an action from an Animation within the Representation | The Representation ID, and the action index |
| moveCameraAction | Moves an action from one index within a sequence to another | The Representation ID, Animation index, Sequence index, Action index, and destination index |
| addWayPoint | Adds a Way Point to the Representation | The Representation ID, and serialized Way Point object |
| updateWayPoint | Updates a Way Point’s data | The Representation ID, Way Point ID, and serialized object with new data |
| addViewPoint | Adds a View Point to the Representation | The Representation ID, and the serialized View Point |
| updateViewPoint | Updates a View Point’s data | The Representation ID, View Point ID, and serialized object with new data |
| removeWayPoint | Removes a Way Point from the Representation | The Representation ID, and Way Point ID |
| removeViewPoint | Removes a View Point from the Representation | The Representation ID, and View Point ID |
| addCourse | Adds a Course to a Representation | The Representation ID, and serialized Course object |
| updateCourse | Updates a Course’s data | The Representation ID, Course ID, |

| | | |
|--|--|---|
| | | and serialized Course ID |
| removeCourse | Removes a Course from a Representaion | The Representation ID, and Course ID |
| addPath | Adds a Path to a Representation | The Representation ID, and serialized Path object |
| updatePath | Updates a Path's data | The Representation ID, Path ID, and serialized Path object |
| removePath | Removes a Path from a Representation | The Representation ID, and Path ID |
| addLabel | Adds a Label to a Representation | The Representation ID, and serialized Label object |
| updateLabel | Updates a Label object's data | The Representation ID, Label ID, and serialized Label object |
| removeLabel | Removes a Label object from a Representation | The Representation ID, and Label ID |
| <i>Representation Management Commands</i> | | |
| createRepresentation | Creates a Representation | The common name of the Representation, and Dataset ID to which it belongs |
| disposeActiveRepresentation | Dispose of the currently active Representation, and release the resources used by the Representation | The Representation ID |
| downloadRepresentation | Downloads a Representation, in its XML form | The Representation ID |
| saveRepresentation | Saves a Representation to the Biolucida database, for subsequent recall and use | The Representation ID |
| loadRepresentation | Loads a previously saved Representation | The Representation's database ID |
| createRepresentationVia-Description | Instantiates a Representation | The Dataset ID to which the Representation belongs, and the serialized RepresentationDescription object |
| getSavedRepresentationsCatalog | Get a list of the saved Representations in the Biolucida database | None |
| deleteRepresentation | Removes a saved Representation from the Biolucida database | The Representation database ID |
| <i>Custom Content Use and Management</i> | | |
| getUserContentCatalog | Gets a catalog composed a Animations, Sequences, and other author-generated content | The user's ID |
| getSavedAnimation | Retrieves a previously saved Animation from the Biolucida database | The Animaiton database ID |
| saveAnimation | Saves an Animation to the Biolucida database | The serialized Animation object |
| getSavedWayPoint | Retrieves a saved Way Point from the Biolucida database | The Way Point database ID |
| saveWayPoint | Saves a Way Point to the Biolucida database | The serialized Way Point object |
| getSavedViewPoint | Retrieves a saved View Point from the Biolucida database | The View Point database ID |
| getSavedCourse | Retrieves a saved Course from the Biolucida database | The Course database ID |
| saveCourse | Saves a Course to the Biolucida database | The serialized Course object |
| getSavedPath | Retrieves a saved Path from the Biolucida database | The Path database ID |

| | | |
|---|--|---|
| savePath | Saves a Path to the Biolucida database | The serialized Path object |
| getNewSceneContent | Retrieves new content associated with a Representation which was added or changed in some way since the last update was performed. | The Representation ID |
| getUpdatedInstanceIDs | Retrieves a list of instance IDs which were updated since this request was last made. | The Representation ID |
| <i>Dataset Manipulation and Management</i> | | |
| createDataset | Creates a blank dataset | None |
| getAvailableDataHandlers | Retrieves a list of Data Handler descriptions which are currently available in the Biolucida system | None |
| getDataPreview | Gets a fixed-size subset of the contents of the Dataset found on the Biolucida server | The Dataset ID |
| getAttributedData | Gets the data attributed to a given Entity within a specified Representation | The Dataset ID, Representation ID, and Entity ID |
| getDataSubset | Gets a small subset of the contents of the Dataset found on the Biolucida server | The Dataset ID |
| instantiateDataHandler | Initializes an instance of a Data Handler, and assigns it to a given Dataset | The Dataset ID, and the serialized Data Handler Description |
| performExtraction | Initiates the data extraction operations in all Data Handlers assigned to a dataset | The Dataset ID |
| updateDataHandler | Updates a Data Handler's attributes to values selected by the author | The Dataset ID, Data Handler ID, and serialized Data Handler Description |
| getCurrentFieldSet | Retrieves the descriptions of all fields found in the Record Manager in a given Dataset | The Dataset ID |
| getAppropriatePropagator-Packages | Retrieves a list of appropriate Propagator Packages, given the available fields in the current Dataset | The Dataset ID |
| updatePropagatorPackage | Updates a given Propagator Package's attribute values to those selected by the user | The Dataset ID, and the serialized version of the Propagator Package |
| addPropagatorPackage | Adds a Propagator Package to a given dataset | The Dataset ID, and the Propagator Package ID |
| uploadDataFile | Uploads a data file to the Biolucida server, making it available for future sessions | The transaction key |
| uploadTemporaryDataFile | Uploads a data file to the Biolucida server for use during the current working session only | The transaction key |
| getDataFileCatalog | Retrieves a list of the files currently available as data sources on the Biolucida server | None |
| <i>System Administration Management Commands</i> | | |
| createEntity | Creates a concept within the Biolucida catalog | The concept name and description |
| createAppearance | Creates a Material type within the Biolucida database | The Material name, description, and the serialized version of the Material object |
| deleteAppearance | Removes a Material type from the Biolucida database | The Material's database ID |
| updateAppearance | Updates a Material type within the Biolucida database | The Material's database ID, and the serialized version of the Material object |

| | | |
|--------------------------------------|--|--|
| uploadNewConceptAssociated-Shape | Creates a new concept, along with its associated Shape | The concept name, description, and the textual contents of the X3D or VRML file used to create its shape |
| associateMeshWithShape | Creates an association between a Shape entry and a Mesh entry in the Biolucida database | The mesh database ID, and the Shape database ID |
| updateFromReferenceOntology | Updates the internal Ontology storage within Biolucida via a selected reference ontology | The reference ontology ID |
| createConcept | Creates a concept in the Biolucida database | The concept name and description |
| deleteConcept | Removes a concept within the Biolucida database | The concept ID |
| updateConcept | Updates a concept's information within the Biolucida database | The serialized form of the Item Profile which describes the concept |
| deleteConcept | Removes a concept from the Biolucida database | The concept ID |
| createConceptRelation | Creates an ontological relation between concepts within the Biolucida database | The "from" concept ID, the relation ID, and the "to" concept ID |
| removeConceptRelation | Removes an ontological concept-to-concept relation from the Biolucida database | The concept-concept relation ID |
| associateTextureWith-Appearance | Creates an association between a texture entry and an appearance entry within the Biolucida database | The texture ID and the Material database ID |
| associateAppearanceWithShape | Creates an association between a Shape entry and an Appearance entry in the Biolucida database. | The Material ID and the Shape ID |
| General Query Commands | | |
| getRepresentation | Retrieves the serialized version of a Representation for reconstitution at the client | The Representation ID |
| getAllConcepts | Retrieves a list of the concepts and their descriptions | None |
| getRelations | Obtains the list of relation IDs contained within the Biolucida server | |
| getAppearance | Retrieves the Material type in serialized XML format | |
| getRelation | Retrieves the XML serialization of a relation in the Biolucida system, including concepts and their relations among each other | |
| getInstance | Retrieves an Entity instance from a Representation | |
| getCompatibleRepresentations-Catalog | Retrieves a list of Representations compatible with the Dataset of interest | |
| getClientSideRepresentation-Class | Retrieves the client-side class of a Representation, to be instantiated on the client side via reflection | |

3D Scene Access and Manipulation Commands

| Command | Description |
|---|---|
| <i>Entity Manipulation</i> | |
| addEntity | Adds a series of Entities to the scene |
| removeEntity | Removes the currently selected Entity from the scene |
| setEntityPosition | Changes an Entity's position instantaneously |
| setEntityRotation | Changes an Entity's rotation instantaneously |
| setEntityScale | Changes an Entity's scale instantaneously |
| changeScale | Changes an Entity's scale, animating the change over a period of time |
| rotateEntity | Changes an Entity's rotation, animating the change over a period of time |
| moveEntity | Changes an Entity's position animation the motion over a period of time |
| isolateEntity | Highlights an Entity and sets all others semi-transparent, deactivating their touch-sensors |
| deisolateEntity | Brings all Entities in the scene to the foreground, setting their Materials' attributes to their "scene" value and activating their touch sensors |
| showEntity | Restores an Entity to its scene-based appearance |
| hideEntity | Makes an Entity totally transparent and deactivates its touch sensors |
| highlightEntity | Changes an Entity's appearance to that of the default highlight material |
| dehighlightEntity | Returns an Entity to its scene-based appearance |
| setAmbientIntensity | Sets the ambient intensity value of an Entity's Material |
| setEnableTexture | Switches the Appearance used by an Entity to that of the texture-based Material node contained in the Entity |
| setDiffuseColor | Changes an Entity Material's diffuse color to a desired value |
| setEmissiveColor | Changes an Entity Material's emissive color to a desired value |
| setShininess | Changes an Entity Material's shininess to a desired value |
| setSpecularColor | Changes an Entity Material's specular color to a desired value |
| setTexture | Changes an Entity Material's texture to a desired graphic |
| setTransparency | Changes an Entity Material's transparency to a desired value |
| setMaterial | Changes an Entity's Material to a supplied Material instantaneously |
| applyMaterial | Changes an Entity's Material to a supplied Material over a period of time |
| setTouchSensorEnabled | Toggles an Entity's touch sensor on and off |
| setEnableEntityRotationControl | Toggles an Entity's rotation control, which manipulates its orientation |
| <i>Scene Manipulation Commands</i> | |
| clearScene | Removes all Entities and Context Objects from the scene |
| bringAllEntitiesToForeground | Brings all Entities in the scene to the foreground, setting their Materials' attributes to their "scene" value and activating their touch sensors |
| setEnableAllTouchSensors | Toggles the activation of all Entity's touch sensors on/off |
| setHighlightMaterial | Sets the default highlight Material to the supplied Material |
| sendAllEntitiesToBackground | Makes all Entities semitransparent and deactivates their touch sensors |
| setUniversalScale | Sets the scale for all objects in the scene |
| setUniversalEntityScale | Sets the scale of the transform containing the Entities in the scene |
| setUniversalEntityTransform | Sets the position of the transform containing the Entities in the scene |
| replaceWorldWithNewContent | Swaps the content of the current world for what is supplied |
| <i>Camera Manipulation</i> | |
| moveView | Moves the camera to the designated position and orientation |

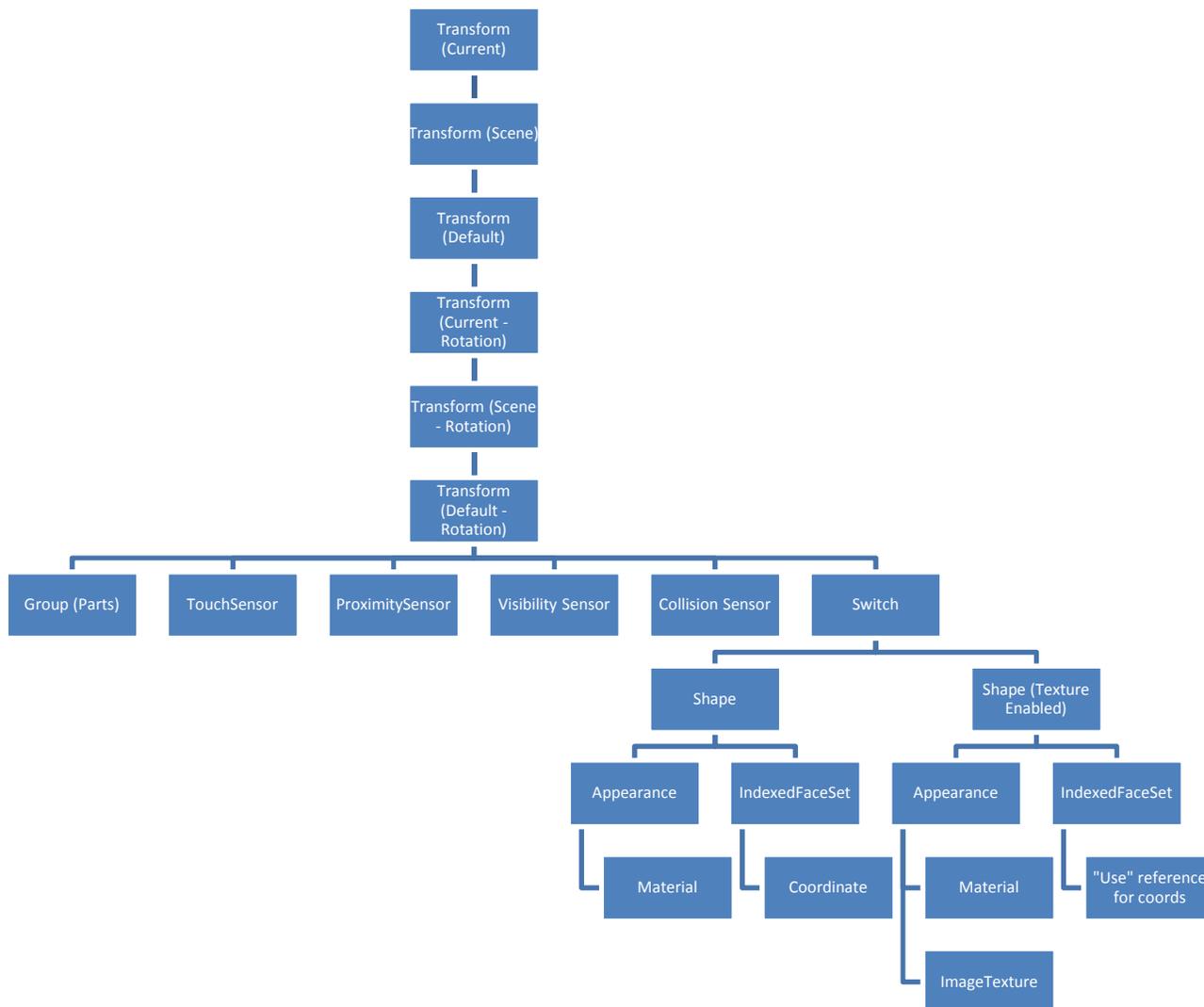
| | |
|--|--|
| lookAt | Moves the camera to a position and orientation where it is pointed at a selected object |
| moveCamera | Moves the camera to a designated position over time |
| rotateCamera | Rotates the camera to a designated orientation over time |
| moveAndRotateCamera | Moves the camera to a designated position and orientation over time |
| switchToFrontView | Moves the camera to the “front” of an object, viewing the object from a optimal distance |
| switchToLastUserView | Moves the camera to the saved camera position and orientation |
| switchToLeftSideView | Moves the camera to the “left” of an object, viewing the object from a optimal distance |
| switchToRightSideView | Moves the camera to the “right” of an object, viewing the object from a optimal distance |
| switchToOverheadView | Moves the camera to the “top” of an object, viewing the object from a optimal distance |
| switchToRearView | Moves the camera to the “rear” of an object, viewing the object from a optimal distance |
| switchToUndersideView | Moves the camera to the “underside” of an object, viewing the object from a optimal distance |
| <i>Contextual and Temporary Object Manipulation</i> | |
| moveContextualObject | Moves a contextual object to a desired position |
| addEntityPin | Adds a label which identifies an Entity via a connecting line |
| removeEntityPin | Removes a pin associated with an Entity |
| removeContextualObject | Removes a Contextual Object from the scene |
| rotateContextualObject | Changes a Contextual Object’s rotation |
| moveClipPlane | Changes the position of the scene’s contents relative to the world’s cut-plane, which can be toggled on/off |
| addAxes | Add a contextual model of axes into the scene |
| addGhost | Adds a “ghost” of an Entity into the scene, which is basically a semi-transparent grey rendition of the Entity, used for animation transitions |
| addXAxis | Adds the model representing the X axis only |
| addYAxis | Adds the model representing the Y axis only |
| addZAxis | Adds the model representing the Z axis only |
| addCameraGhost | Adds a ghost-like rendition of a camera, used to visualized viewpoints from a 3 rd person perspective |
| removeCameraGhost | Removes a camera ghost from the scene |
| highlightContextualObject | Sets the material of a contextual object to that of the default highlight |
| insertImageStack | Inserts a slice-based image stack for visualization within the scene |
| setImageStackSlice | Sets the displayed image within an image stack |
| setLabelVisible | Toggles the visibility of a label |
| setPathVisible | Toggles the visible display of a path in the scene |
| setViewPointVisible | Toggles the visible display of a View Point in the scene |
| setWayPointVisible | Toggles the visible display of a Way Point in the scene |
| setCourseVisible | Toggles the visible display of a Course in the scene |
| setEnabledClipPlane | Toggles the world clip plane on/off |
| <i>Query Commands</i> | |
| getClipPlanePosition | Get the current position of the clip plane relative to the rest of the scene’s other contents |
| getClipPlaneRotation | Get eh current rotation of the world clip plane |
| getEntityPosition | Get the current position of a given Entity |
| getEntityRotation | Get the current rotation of a given Entity |
| getEntityScale | Get the scale of a given Entity |

| | |
|------------------------------|---|
| getGhostPosition | Get the current position of a given Contextual Object |
| getGhostRotation | Get the current rotation of a given Contextual Object |
| getGhostScale | Get the scale of a given Contextual Object |
| getCameraLinearDistance | Get the distance between a given point and the current position of the camera |
| getCameraOrientation | Get the camera's current rotation |
| getCameraPosition | Get the camera's current position |
| getContextualContent | Get the specified Contextual Content object (by ID) |
| getContextualContentPosition | Get the specified Contextual Content object's current position |
| getContextualContentRotation | Get the specified Contextual Content object's current rotation |
| getContextualContentScale | Get the specified Contextual Content object's current scale |
| getSelectedEntity | Get the currently selected Entity |

APPENDIX C

Default Entity X3D Node Structure

The following hierarchical structure represents the standard X3D node tree used as the Web3D compatible implementation of the default “Entity” in a Biolucida 3D representation, visualized by the Xj3D browser.



APPENDIX D

User Evaluation Task List and SUS Questionnaire

The following is a list of tasks which each subject was asked to perform in the user evaluation exercise, described in chapter 9. Subjects were given a brief tutorial on the use of the Biolucida software and subsequently asked to complete a series of tasks, while adhering to a “think aloud” protocol. Following the exercise, subjects answer the SUS items described below.

Task List

1. Create a scene containing the heart, lungs, and circulatory system
2. Add all of the nerves in the catalog
3. Add the major organs contained in the thorax
4. Add the set of ribs and the spinal column
5. Select each set of ribs and make them semi-transparent
6. Zoom in on the heart and lungs
7. Move the left set of ribs aside so as not to obscure the left lung
8. Create an animation where the lungs change color to blue, followed by the heart, and then the arterial tree
9. Publish the animation in a Web3D file

SUS Questionnaire

1. I think that I would like to use this system frequently.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|---------------------------|-----------------------|-----------------------|-----------------------|------------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

2. I found the system unnecessarily complex.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|---------------------------|-----------------------|-----------------------|-----------------------|------------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

3. I thought the system was easy to use.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

4. I think that I would need the support of a technical person to be able to use this system.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

5. I found the various functions in this system were well integrated.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

6. I thought there was too much inconsistency in this system.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

7. I would imagine that most people would learn to use this system very quickly.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

8. I found the system very cumbersome to use.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|--------------------------------------|-----------------------|-----------------------|-----------------------|-----------------------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

9. I felt very confident using the system.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|--------------------------------------|-----------------------|-----------------------|-----------------------|-----------------------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

10. I needed to learn a lot of things before I could get going with this system.

| Strongly Disagree 1 | 2 | 3 | 4 | Strongly Agree 5 |
|--------------------------------------|-----------------------|-----------------------|-----------------------|-----------------------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

APPENDIX E
SUS Questionnaire and Task Time Results

| Question Number | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------------|---------------------------------|---|---|---|---|---|---|---|---|---|----|
| | Time | | | | | | | | | | |
| | Participants | | | | | | | | | | |
| | <i>3D Design Software Users</i> | | | | | | | | | | |
| 1 | 24.2 | 2 | 2 | 4 | 2 | 4 | 2 | 3 | 2 | 4 | 1 |
| 2 | 24.8 | 1 | 3 | 4 | 1 | 4 | 1 | 3 | 2 | 4 | 1 |
| 3 | 27.9 | 2 | 2 | 3 | 1 | 3 | 2 | 3 | 1 | 3 | 1 |
| 4 | 21.8 | 2 | 3 | 3 | 2 | 4 | 2 | 4 | 1 | 4 | 1 |
| 5 | 29.1 | 3 | 3 | 3 | 1 | 5 | 1 | 3 | 2 | 3 | 1 |
| 6 | 32.0 | 2 | 3 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 2 |
| 7 | 45.2 | 2 | 3 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 1 |
| 8 | 23.5 | 1 | 3 | 4 | 1 | 3 | 2 | 3 | 1 | 4 | 1 |
| 9 | 22.0 | 2 | 2 | 3 | 1 | 4 | 2 | 3 | 1 | 4 | 1 |
| 10 | 24.5 | 2 | 2 | 3 | 1 | 4 | 2 | 3 | 2 | 4 | 1 |
| | <i>Massage Practitioners</i> | | | | | | | | | | |
| 11 | 23.1 | 3 | 1 | 4 | 3 | 4 | 2 | 4 | 2 | 5 | 2 |
| 12 | 19.9 | 3 | 2 | 4 | 2 | 5 | 2 | 4 | 2 | 5 | 2 |
| 13 | 24.0 | 4 | 3 | 3 | 2 | 5 | 2 | 3 | 2 | 4 | 1 |
| 14 | 23.8 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 2 | 3 | 2 |
| 15 | 26.2 | 3 | 3 | 4 | 2 | 5 | 3 | 3 | 3 | 4 | 1 |
| 16 | 21.6 | 4 | 3 | 3 | 2 | 4 | 2 | 4 | 2 | 3 | 1 |
| 17 | 28.2 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 2 | 3 | 1 |
| 18 | 24.4 | 3 | 1 | 4 | 3 | 3 | 2 | 4 | 2 | 4 | 1 |
| 19 | 25.0 | 4 | 1 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 1 |
| 20 | 25.8 | 4 | 1 | 3 | 2 | 4 | 1 | 3 | 1 | 4 | 2 |

VITA

Wayne Warren graduated with a Doctor of Philosophy in Biomedical Informatics and Medical Education from the University of Washington in 2012, advised by James F. Brinkley III. He received his Bachelor's degree in Molecular Biology and Biochemistry from Middlebury College in 1998 advised by Grace Spatafora.

His research interests primarily focus on utilizing 3D technologies for data visualization and educational activities, as well as enabling collaborative work.

He welcomes your comments at ww@uwashington.edu.