Virtual Human Problem-Solving Environments[*,**]

Richard C. Ward[1], Line C. Pouchard[2], Nancy B. Munro[3], and Sarah K. Fischer[1]

[1] Computational Sciences and Engineering Division
wardrc1@ornl.gov, fischersk1@ornl.gov
[2] Computer Science and Mathematics Division
pouchardlc@ornl.gov
[3] Environmental Sciences Division
Oak Ridge National Laboratory
P.O. Box 2008, Oak Ridge, Tennessee, USA, 37831-6085
munronb@ornl.gov

Abstract. Interest in complex integrated digital or virtual human modeling has seen a significant increase over the last decade. Coincident with that increased interest, Oak Ridge National Laboratory (ORNL) initiated the development of a human simulation tool, the Virtual Human. The Virtual Human includes a problem-solving environment (PSE) for implementing the integration of physiological models in different programming languages and connecting physiological function to anatomy. The Virtual Human PSE (VHPSE) provides the computational framework with which to develop the concept of a "Virtual Human." Supporting the framework is a data definition for modeling parameters, PhysioML, a Virtual Human Database (VHDB), and a Web-based graphical user interface (GUI) developed using Java. Following description of the VHPSE, we discuss four example implementations of models within the framework.

Further expansion of a human modeling environment was carried out in the Defense Advanced Research Projects Agency Virtual Soldier Project. SCIRun served as the Virtual Soldier problem solving environment (VSPSE). We review and compare specific developments in these projects that have significant potential for the future of Virtual Human modeling and simulation. We conclude with an evaluation of areas of future work that will provide important extensions to the VHPSE and VSPSE and make possible a fully-integrated environment for human anatomical and physiological modeling: the Virtual Human.

Keywords: Virtual Human, Virtual Soldier, problem-solving environments, human modeling and simulation, anatomy, physiology, graphical user interface.

[*] Portions of this research were supported by a grant from the Defense Advanced Research Projects Agency, executed by the U.S. Army Medical Research and Materiel Command/TATRC Cooperative Agreement, Contract W81XWH-04-2-0012. Portions of this research were sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725.

[**] Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Y. Cai (Ed.): Digital Human Modeling, LNAI 4650, pp. 108–132, 2008.

1 Introduction

Interest in complex integrated digital or virtual human modeling has seen a significant increase over the last decade. Coincident with that increased interest, Oak Ridge National Laboratory (ORNL) initiated the development of a human simulation tool, the Virtual Human in 1996 [1,2]. The Virtual Human was envisioned as a research/testing environment having an integrated system of biochemical and biophysical models, associated data and advanced computational algorithms coupled with a digital, solid-body model of the anatomy. Our initial effort was focused on development of a problem-solving environment (PSE) for implementing the integration of physiological model components written in different programming languages. Construction of PSEs through software components is an approach that has engendered much recent interest [3]. A second objective was to connect physiological function to anatomy provided by high-resolution, three-dimensional (3-D) CT or MRI data. The Virtual Human PSE (or VHPSE) provided the computational framework with which to develop the concept of a "Virtual Human." This chapter will focus on the development of the VHPSE and the information management system to support the PSE. In addition, details are provided on a selection of example model implementations using the VHPSE.

Supporting the framework are a data definition for modeling parameters, PhysioML, a Virtual Human Database (VHDB), and a Web-based graphical user interface (GUI) developed using Java that provides easy, rapid access to models and data from several points of entry. In addition, a virtual human must be based on some model of human anatomical data; in our case all the examples discussed here use the National Library of Medicine (NLM) Visible Human male CT data set[1]. The physiological models are compartment or circuit models, termed high-level integrative physiological (HIP) models[2] and associated physiological model parameters and initial conditions.

We envision that the Virtual Human could serve as a platform for national and international users from governments, academia, and industry to investigate a wide range of human biological, chemical, and physical responses to both external and internal stimuli. Our effort will eventually incorporate mechanical and electrical tissue properties and biological responses from organ and cellular tissue function, with results viewed using 3-D anatomical models linked to anatomical ontologies.

Biomedical applications might include prosthesis design, evaluation of microgravity effects, and personal medical informatics for diagnosis, patient education, and selection of therapeutics. Eventually, use of the Virtual Human could minimize the need for human subjects being involved in testing and also reduce the need for animal studies. In conjunction with a time-serial history of telemetered medical data, Virtual Human will provide assistance in emergency medical procedures and triage.

For military research, such a virtual human incorporating both anatomy and physiology could provide a capability to evaluate the effectiveness and safety levels

[1] http://www.nlm.nih.gov/research/visible/ (last accessed 6 December 2007)

[2] http://nsr.bioeng.washington.edu/PLN/Members/mneal/integrated_html/view (last accessed 6 December 2007)

of non-lethal technologies and the effectiveness of advanced clothing and armor. A virtual human could also provide the ability to simulate training scenarios involving new equipment and methods and the capability of testing vehicle designs for safety.

We used the VHPSE to examine a few example human biomodeling applications including 1) simple cardiovascular function, 2) modeling lung sounds as signatures for pulmonary disease or injury, 3) response to environmental and occupational inhalation exposure, and 4) pulmonary edema. Details of each of these examples will be presented.

In 2004, the Defense Advanced Research Projects Agency (DARPA) initiated the Virtual Soldier Project[3]. The goal of this project was to predict the location of a fragment wound based on physiological response to aid the Medic in deciding how to treat a wounded soldier. Modeling of the physiology was validated against experimental porcine data to develop a rigorous predictive tool. In addition, detailed phantoms were constructed using automatic segmentation of CT imagery, with every voxel labeled anatomically. This 3-D human phantom connected to a detailed anatomical ontology and integrated with the physiological modeling was termed the Holographic Medical Electronic Record (or Holomer) [4]. The Virtual Soldier Project used SCIRun[4] as a platform for a Virtual Soldier PSE (VSPSE). SCIRun offered both a visualization environment capable of stereo rendering and a simulation environment [5, 6]. Using SCIRun as a platform for the VSPSE, significant progress was made toward the design of the human Holomer and the ability to predict physiological state from a limited number of measurements provided a significant advancement over previous human modeling and simulation efforts.

Finally, we will compare and contrast the VHPSE and the VSPSE, indicating where the different environments have contributed to the overall progress toward advancing human modeling and simulation. We also provide specific suggestions for future work that would strengthen each environment and provide a more fully integrated human PSE for human modeling and simulation in the future.

2 Background

ORNL has been involved in computational human modeling nearly since its inception. Early on, the focus was on building mathematical phantoms for determining radiation doses to workers and medical patients. The earliest computational models represented the body and its organs as homogeneous spheres, ellipses, etc. With the need for more accurate calculations of radiation dose, mathematical phantoms of the human body and its organs were developed [7, 8]. These phantoms used simple mathematical expressions to define the surfaces of the body and the organs. The masses of the organs were consistent with the original Reference Man data [9]. The 1970s and 1980s saw the application of these models in epidemiological studies including those of the survivors of the Hiroshima and Nagasaki bombings.

Paralleling the development of mathematical phantoms, efforts were also underway to model the behavior of inhaled or ingested radionuclides within the body. These models

[3] Virtual Soldier Web site (2006) http://www.virtualsoldier.us/ (last accessed 6 December 2007).

[4] SCIRun: A Scientific Computing Problem Solving Environment, Scientific Computing and Imaging Institute (SCI). http://software.sci.utah.edu/scirun.html 2007 (last accessed 6 December 2007).

have now evolved to include the relevant physiological and biokinetic processes and are represented as compartment models. Dosimetric considerations require that the compartments of the biokinetic models be identified with the specific organs and tissues and that proper spatial relationships of the organs and radiosensitive tissues be reflected in the mathematical phantoms.

ORNL researchers involved in the various aspects of human modeling realized that there were common threads as well as common needs among these various modeling efforts, particularly the need for a robust computational modeling environment. In October of 1999, at the instigation of the ORNL Laboratory Director, a Virtual Human Workshop, chaired by Charles DeLisi of Boston University, was held at the National Academy of Sciences in Washington, DC. A short time later, in November of 1999, the First Virtual Human Roadmapping Workshop was held in Rockville, MD. Organized by ORNL and cosponsored by the Joint Nonlethal Weapons Directorate, the consensus of the two workshops was that the Virtual Human concept was "an idea whose time has come" [1]. Other research projects, for example the Physiome Project[5] were also encouraging funding agencies to see the need for a large scale, integrated human modeling effort which bridged between the genome level (recent focus of intense research) and other levels of function including the protein function level or proteome, the biochemical level or metabolome, and the physiological level or physiome.

The Virtual Human Project led others to pursue the Digital Human Project[6], an NSF-funded examination of the possibility of developing an integrated human modeling approach. Led by the Federation of American Scientists, two conferences were held at the National Institutes of Health National Library of Medicine to explore the idea in 2001 and 2002.

3 Virtual Human Problem-Solving Environment

The overall objective of the ORNL Virtual Human Project was to develop a comprehensive capability for computationally simulating human response to any stimulus. To support this effort, work began on a distributed Virtual Human Problem-Solving Environment (VHPSE). Given the complexity of the human body, a Virtual Human PSE represents one of the most complex computational modeling and integration efforts ever undertaken.

The complexity of human modeling and simulation requires a new way of structuring scientific work - a paradigm change in the use of computational resources. Resources (hardware, software, instruments, databases, and people) will be collected into a *computational grid,* which has the ability to provide, on demand, the concentration of terascale computational and information resources required for simulation-intensive research. This type of all-encompassing computational environment is referred to as a problem-solving environment (PSE).[7]

[5] http://physiome.org/ (last accessed 6 December 2007)

[6] http://www.fas.org/main/content.jsp?formAction=325&projectId=12 (last accessed 6 December 2007).

[7] www.cs.purdue.edu/research/cse/pses/ (last accessed 6 December 2007).

Utilizing collaborative technology and the computational grid, a PSE is available over the Internet to connect researchers and other users throughout the world, thus allowing individual researchers to tap into a significant pool of research models for simulation. PSEs are computational environments that typically provide the following:

- A knowledge base or reference database, in this case of anatomical and physiological data and models of the human body,
- Interface modules that connect the output of one model to the input of another model,
- User-friendly graphical user interfaces (GUIs) to allow model building using icons,
- Data storage and data mining capabilities, and
- Advanced visualization capabilities for handling output of model simulations.

A PSE requires middleware software components to unify such a complex problem solving environment. To accomplish this for the VHPSE, we incorporated NetSolve [10] developed by the University of Tennessee. NetSolve is a software environment for networked computing designed to transform disparate computers and software libraries into a unified, easy-to-access computational service. It was designed to support applications that can deliver the power of computational grids to the desktops of users without being complicated to use or difficult to deploy.

Finally, a PSE requires a simple interface for easy accessibility by the user - a sophisticated graphical user interface (GUI). For example, the interface should allow the user to select the degree of sophistication of the phantom (spatial resolution, organs identified, etc.), the degree of sophistication of the physiological models, the spatial and temporal resolution of the computations, the temporal range of the problem, the desired solver to be used, and the nature of the output to be visualized. For the VHPSE we developed a Web-based Java client to serve as the GUI [11]. This interface (referred to as the View) is shown in Figure 1 and includes windows for user login, the anatomical model, the model schematic, and output (physiological response) of the model.

3.1 Java Client/Server Architecture

We chose to develop the VHPSE using Java as the programming language for a variety of reasons. First, Java is a powerful object-oriented language making possible the use of modern programming techniques. Java also provides a distributed object architecture in the form of Remote Method Invocation (RMI)[8] that is vital for the client/server infrastructure of the architecture presented here. The Java language also includes built-in support for multithreading, making possible simultaneous use of the server by multiple clients. Second, Java is platform independent. Any platform with a Java Virtual Machine (JVM) can run the client and gain access to the back-end simulation code. Finally, Java includes a full-featured class library for user-interface programming. These powerful Java classes facilitate writing the graphical input and output components.

[8] http://java.sun.com/javase/technologies/core/basic/rmi/ (last accessed 6 December 2007).

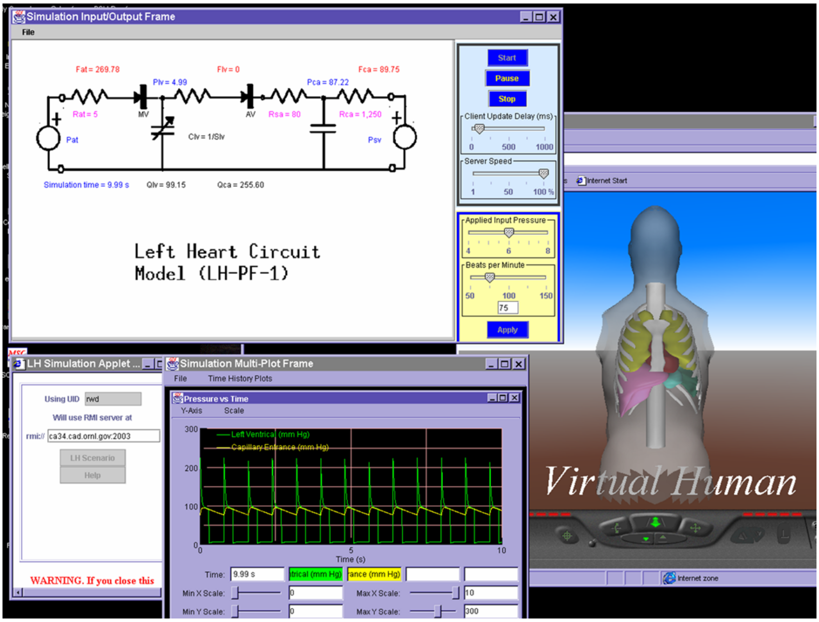


Fig. 1. Components of the VHPSE GUI include an anatomical window, a model diagram, and a physiology display window

An important part of our VHPSE is a user interface to the VHPSE to control the physiological and anatomical models. To implement this we utilized Java RMI. Using Java as our development language allowed us to create a cross-platform environment whose object-oriented development paradigm will facilitate incorporating additional physiological models. Java RMI using Java Native Interface (JNI).[9] JNI connects physiological models written in C or FORTRAN. Since we anticipate that simulations of physiology might be developed in a wide range of programming languages from FORTRAN, to C, to C++, to Java, we wanted an approach that allowed the flexibility to integrate codes using these languages, without having to rewrite the compute engines.

Furthermore, objects can be distributed across the Internet, enabling one to design systems that can be modified easily by dynamically adding new behavior [12]. The combination of Java Database Connectivity and RMI allows applications to communicate with existing database servers in non-Java languages. In the Virtual Human, our models reside on the server while the user interface resides on the client. Java RMI is used to provide communication between the user interface and model executable code.

Figure 2 shows a schematic of the Virtual Human Java Interface client and server architecture. The model executable code resides on the server. Both the model on the server and the Java client are configured using a configuration file specific to the model selected. The Java simulation manager reads the configuration file to construct the input (initial conditions, start and end times, time step, etc.) for the model

[9] http://java.sun.com/j2se/1.5.0/docs/guide/jni/ (last accessed 6 December 2007)

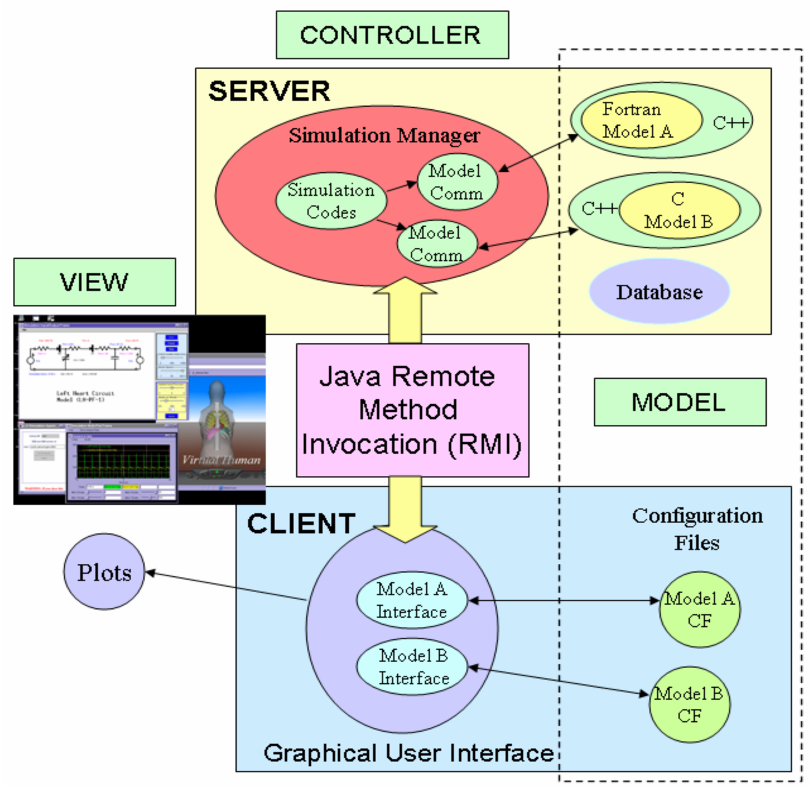


Fig. 2. A Model-Controller-View schematic of the Virtual Human Java client/server PSE

simulation. An unusual feature of the Virtual Human interface is the ability to control the simulation from the client using selected parameters. Prior to the start of the simulation, the user selects parameters that are allowed to vary. A slider bar is constructed for each parameter allowing modification over a specified range. As the simulation proceeds and the user moves a slider bar, the modified value for that parameter is sent from the client to the server and the simulation responds nearly instantaneously. This process, referred to as computational steering, gives researchers and students a very good feel for the effect of parameter changes on model outcomes.

VHPSE uses a Java client/server architecture developed for a legacy code known as the Graphite Reactor Severe Accident Code (GRSAC) [13]. GRSAC simulates a variety of accident scenarios in a graphite reactor [14], where intermediate results are continuously displayed to the user during the simulation rather than at the conclusion of the run. Because of this interaction between the user and the simulation, we require an architecture that supports interactive communication between client and server. Since these properties of the GRSAC simulation (continuous display of results and user interaction with the simulation) were similar to what is expected in physiological modeling and simulation, so it proved to be quite practical to adapt this framework for use as the basis for the VHPSE.

3.2 Model-View-Controller (MVC) Design

We used the Model-View-Controller (MVC) design pattern [15] to implement the VHPSE. As shown in Fig. 2, from the client's point of view there are two **Controllers**. The server represents a data controller since it continually recalculates and updates the data. The client also includes its own controller component with which the user controls the calculation. The client maintains a data model to store the incoming results, and the **View component** is, of course, the graphical output provided to the user, which is shown in Fig. 1 for a simple cardiovascular model. Java RMI provides the connection between the **View** (client) and **Controller** and between the **Controller** and the **Models**, consisting of the anatomical and physiological models, model equations, and the VHDB (Fig. 2). The models are stored on the server, but control both the computations on the server and the configuration of the client GUI.

3.3 Alternative Approach Using CORBA

An alternative approach [3] was also undertaken to leverage high-performance legacy codes by converting them to Common Object Request Broker Architecture (CORBA) components for the VHPSE, using a CORBA-oriented wrapper generator (COWG). Using CORBA guarantees that components written in different programming languages can interoperate with each other.

One example of legacy code that was implemented using the COWG approach was an Message Passing Interface (MPI)-based finite-element (FE) computational fluid dynamics (CFD) code [16], for simulating incompressible Navier–Stokes flows in the lungs. Wrapped as CORBA components, this legacy code can be reused in a distributed computing environment. A user can submit a task to the wrapped CFD component through a Web page without knowing the exact implementation of the component. In this way, a user's desktop computing environment can be extended to a high-performance computing environment using a cluster of workstations or a parallel computer. While the COWG approach worked well, we continued to focus our effort on the Java MVC design approach as described previously.

3.4 Anatomical Modeling

Some form of anatomy must be used in most human modeling efforts. This could be simply a compartment, i.e., organ or sub-organ level model up to a realistic FE mesh model of multiple organs and the exterior of the body (as an example for studies of the effect of external trauma). To provide anatomy for our VHPSE we chose to develop a human phantom from the Visible Human male data set using non-uniform rational B-Splines (or NURBS). Our selection of NURBS for the anatomical model was based on the ease with which NURBS surfaces can be modified and scaled. We expect this feature will be useful for parameterizing a human phantom across age and gender for radiation dosimetry studies (following the lead of the original ORNL "Mathematical Phantom" [7] or MIRD model [8]. It should also be possible to animate a NURBS-based phantom using animation software for developing avatars for medical training software and other applications.

3.5 NURBS Phantom for the VHPSE

Several steps are required to develop a high-resolution human phantom. The segmentation was performed using an IDL program[10].

1) Segment (identify) the organs in a CT image. This is accomplished automatically for some organs (e.g., lungs and skeleton), but must be done by hand for most (see Fig. 3).
2) Convert the output of the segmentation program to a DXF file containing points specifying the surface of a desired organ or skeletal component.
3) Create a separate set or slice of points, corresponding to each frozen CT cross-section.
4) Convert the DXF file into a NURBS file. Due to the large number of data points involved, only every third or fifth slice of the Visible Human frozen CT images might be used.

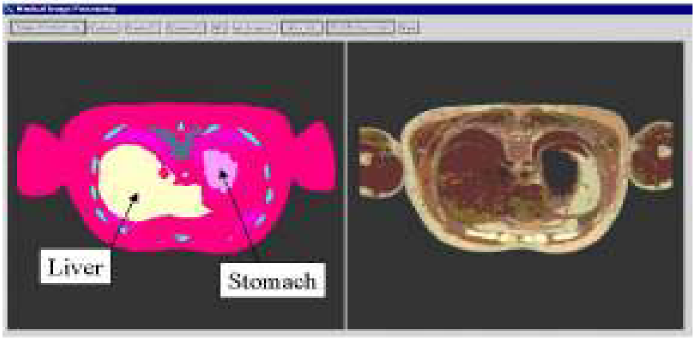


Fig. 3. Automatic (skeleton and lung) and Manual Segmentation (liver and stomach) of a Visible Human Image

We created NURBS surfaces using the Rhino software[11]. We discovered two methods to create a NURBS surface for an organ:

1) In the first method, the user must trace each CT slice, taking advantage of the "snap to points" capability of Rhino. The user then "lofts" the slices; i.e., creates vertical NURBS curves that connect the points on the horizontal slices.
2) In the second method, which works well for ellipsoidal or spherical objects such as the heart, the user starts with an ellipsoid and by "pulling and tugging" makes the ellipsoid match the CT image data.

Using either method, one can smooth the surface by selecting the number of control points in the horizontal and vertical directions. We found that the heart

[10] ITT Visual Information Solutions, IDL Software, http://www.ittvis.com/idl, (2007) (last accessed on 6 December 2007)

[11] Rhino, NURBS modeling tool by McNeel North America, http://rhino3d.com (2007) (last accessed 6 December 2007)

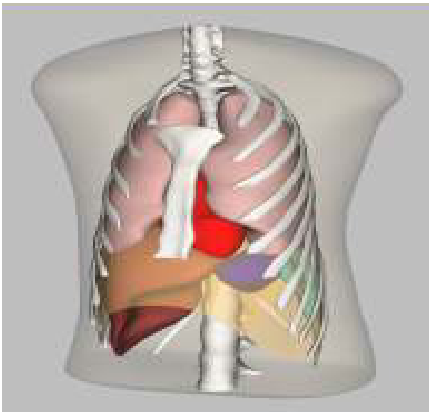


Fig. 4. NURBS Model of Visible Human (male) Thorax for use in VHPSE

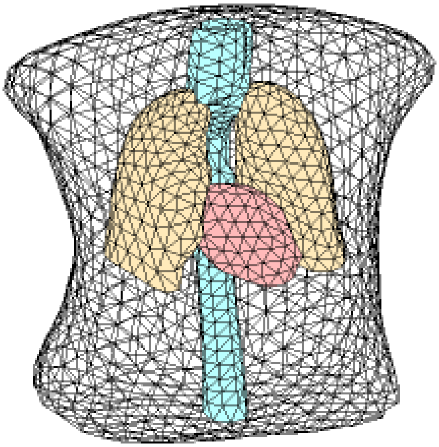


Fig. 5. Finite-element mesh of Visible Human (male) torso

required about 100 control points and each lung about 120 control points. This creates very high-resolution, smooth surfaces of these organs and illustrates a most important advantage of using a NURBS-based representation for the various organs. Whereas the CT representation of an organ must be stored as thousands of data points in files of many megabytes in size, the corresponding NURBS representation can be achieved with only a few hundred or so data points stored in files of a few kilobytes in size.

We manually segmented and classified the liver, spleen, stomach and esophagus. The diaphragm was produced by hand-drawing a surface to fit below the lungs and heart and above the liver, spleen and stomach. The diaphragm is needed for dynamic working models of the pulmonary system. The completed NURBS model of the torso data (the VH phantom) is shown in Fig. 4. The 3-D Visible Human male anatomical model was also rendered using Virtual Reality Modeling Language (VRML) and linked to one of the physiological models implemented in the VHPSE. Users could then click on a selected organ in the 3-D anatomy and display the appropriate physiological models (e.g., heart) and the corresponding simulation results. This prototype NURBS model forms the basis for developing FE mesh models for trauma simulation, for animations for medical training, and for creating a scalable "Reference Man" phantom for future work in radiation exposure and for automobile crash testing among many other possible applications.

A FE mesh model would be required for detailed simulation of the effects of trauma resulting from vehicular crash testing [17]. We found that creating a NURBS surface of the very convoluted parts, such as the spine with its irregularities resulting from repeated vertebrae and disk units, was exceptionally difficult. For purposes of the FE modeling, where such details can often be ignored, we chose to simplify the spine to a cylinder of tapered, elliptical cross-section that retained the curvature of the spine (Fig. 5).

3.6 Physiological Modeling

Physiology is modeled using compartment models or circuit models, approaches that we will refer to as high-level integrative physiological (HIP) modeling, which was the term adopted in the Virtual Soldier Project. These models are described by complex sets of coupled non-linear ordinary differential equations (ODEs), for which parameters are derived by experiments on animals or, in some cases, from medical data obtained from humans. Each model must be provided with a set of initial conditions to describe the initial state of the system.

One important step toward developing a collaborative simulation and modeling environment that uses legacy coding is the development of standardized model description languages. One very common way to do this is to use Extensible Markup Language (XML) for the model description. XML is a self-descriptive text-based meta-language for defining what data are, as opposed to how they are displayed [18]. It is a subset of SGML (Standard Generalized Markup Language) developed for digitized documentation [19]. XML enables the user to create user-specified tags allowing the definition, validation, analysis, and transmission of data between applications [20]. A Document Type Definition (DTD) file defines how tags should be translated by the application presenting the data. Two well-known examples of XMLs for biosystems modeling are System Biology Markup Language (SBML)[12] and CellML[13].

3.7 PhysioML: An XML for Virtual Human

We developed [21] a physiological modeling language (called PhysioML) using XML. PhysioML has element tags for describing the model, providing transfer coefficients for linear models, and, possibly a unique feature, tags for controlling display of results to the GUI and for computational steering (see Table 1). While other XML languages such as SBML and cellML incorporated the first concept, PhysioML is unique in providing a capability to control the interface display and computational steering. For linear systems, the coefficients of the coupled differential equations can be prescribed as a transfer matrix using the ***transfer*** PhysioML element tag.

[12] http://sbml.org (last accessed 6 December 2007)

[13] http://www.cellml.org/ (last accessed 6 December 2007)

Table 1. Main XML tags used in PhysioML

Display	**Element Tag**	**User Interface Definition**
	Panel	defines a window panel
	Image	URL for screen image
	Label	Screen display
Model		**Model Definition**
	Variable	Define variable (name, initial value)
	Transfer	transfer matrix
	Box	Define a compartment
	Boxstuff	Image displayed in compartment
	Boxtrigger	Threshold for compartment

At the moment there is no means to incorporate the model description (functions) for nonlinear problems or those with complex mathematical description (e.g., using step functions) in the XML format. We envision eventually that this can be accomplished using MathML. For details on PhysioML and examples using PhysioML see: http://www.ornl.gov/~rwd/VH/xmlfiles.html.

While we describe PhysioML, we believe that in the long run modelers should subscribe to a well-established standard for model description. It appears that SBML is now being recognized as such a standard. We therefore intend to add the additional display and computational steering features of PhysioML as extensions to the popular SBML modeling language.

4 Information Management for the Virtual Human

Information management for the VHPSE includes database design and prototypical implementation (which utilizes Oracle) and associated tools to access that database. Figure 6 is a schematic for the Virtual Human information management approach [22]. For each model, a PhysioML file (the configuration file) is extracted to control the simulation and the appearance of the client screen for that specific model. Various tools were used for information management including 1) xCase Professional,[14] a graphical design tool for designing database tables using an entity relationship diagram (ERD), 2) a local Microsoft Access interface to enter data into the Oracle database tables, and 3) a Java/XML interface to enter data into tables remotely.

4.1 Virtual Human Database (VHDB)

A relational database was designed and implemented to support the Virtual Human project. Because of the evolving nature of the project, an iterative approach was adopted for database development that employed several techniques borrowed from different information engineering methodologies and adapted to a research environment. Use case scenarios were helpful in identifying the flow of information

[14] xCase User's Manual, Version 5.5, RESolution Ltd., 2000. See http://www.xcase.com/ (last accessed 6 December 2007)

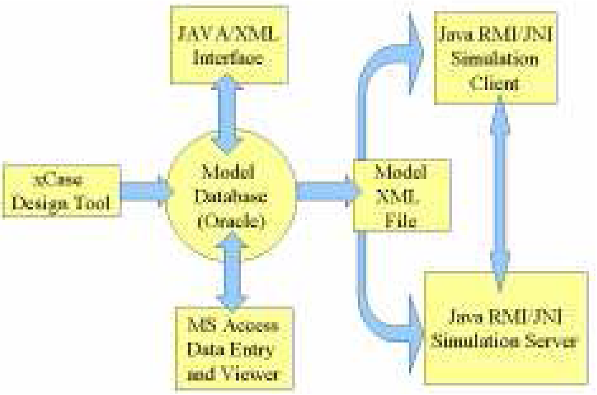

JAVA/XML Interface
xCase Design Tool
Model Database (Oracle)
Model XML File
Java RMI/JNI Simulation Client
Java RMI/JNI Simulation Server
MS Access Data Entry and Viewer

Fig. 6. Schematic for the Virtual Human data and information management

into and out of the database. From that point, the main steps involved in developing the VHDB were: 1) data modeling and 2) database design and construction.

4.2 Data Modeling

Data modeling is an iterative process. Different approaches to model integration were explored the Virtual Human project progressed. In the midst of this evolution, we gathered elements of data and information about the data (metadata) that would constitute the entities of the VHDB. Because of the evolving nature of the project, the data structure needed to be as generic as possible to accommodate new and different types of data as the project progressed. As the entities of the Virtual Human project were identified, they were stored in a text document for discussion. That specification was the precursor to the data dictionary of the VHDB. To analyze the relationships between the different entities, we employed ERDs.

4.3 Database Design and Construction

Because the nature of the VHDB changed as the Virtual Human project itself progressed, it was important to use a software tool to manage the information model, rather than coding and recoding Structured Query Language (SQL) Data Definition Language (DDL) scripts to create or change the database structure. The database development and maintenance tool xCase was used for the VHDB design and construction. The xCase graphical user interface makes it easy to create the ERDs. First the entities are entered. These become the tables of the VHDB when the database is constructed. Then, as more details are known about the entities, their attributes are entered in xCase. These become the columns (also known as fields in database terminology) in the database tables.

Other features of xCase were also used to plan for change and extensibility of the data model. Documentation from the data specification is entered into the tool so that the tool maintains the VHDB data dictionary. Domains are created in the tool for some types of attributes, such as comment and description fields and some types of integer fields. By using these domains instead of just specifying the Oracle datatype, provision is made for easier changes. For example, to change the length of all the comment fields in the database requires making only one change to the ***Comments*** domain definition. All fields using the ***Comments*** domain would automatically get the new length change, rather than having to edit every table in the xCase model tool to change the lengths of affected fields.

The Professional version of xCase can generate SQL DDL for a number of different database products, including Oracle, MySQL, SQL Server, and Microsoft Access. Not only does xCase generate code to create the tables, indexes, etc. that make up the database; it also can execute the scripts through use of an Open Database Connectivity (ODBC) connection. This allows one to forward engineer the database structure from what is contained in the tool, or to reverse engineer a model from an existing database. This database platform independence allows creation of the VHDB on another platform, if desired, for future work.

5 Applications

The Virtual Human Problem Solving Environment (VHPSE) was used to examine several possible biomodeling projects, of which we will focus on four: 1) simple cardiac function models, 2) production of lung sounds due to pulmonary disease or injury, 3) response to environmental and occupational inhalation exposure, and 4) simulating pulmonary edema.

5.1 Cardiovascular Function

To test the VHPSE, we implemented cardiovascular function models from Rideout [23], ranging from simple to more sophisticated. Here we will discuss the implementation of a simple left-heart model, the first discussed by Rideout, into the environment. The interface for the left heart cardiovascular model is shown in Fig. 1. The user login screen is shown in the lower left. This screen allows selection of a particular model. The circuit diagram for the left heart model is shown in the upper left corner window, where blood flow is current in the model and pressure is voltage. The left side of the model represented the left ventricle, the middle the left atria, and the right side is the rest of the entire capillary system. There is no pulmonary cardiovascular system and the pulmonary system is not represented. A plot of the output (in this case for left ventricular pressure and capillary entrance pressure (mm Hg) as function of time(s) is shown the lower left window. Finally, the human anatomy (using the VRML of the Visible Human data as described earlier) is shown in the window on the right side. In this case there was no direct coupling between the physiological model and the anatomy, although that is certainly possible. The left heart circuit model was coded in Java in this case.

5.2 Lung Sounds and Airway Flow

Lung sounds are thought to arise from turbulent flow or the formation of vortices in the third or fourth generation of lung bronchi, a complex computational problem due to the elastic, mucus-lined variable airway boundaries. Breath sounds monitored at anterior and posterior chest sites are used to assess lung diseases and/or injuries such as asthma, emphysema, pneumonia, pleural effusion, and pneumothorax. In combat situations, patient movement and the presence of chest wounds limit such assessment. The throat provides an alternative stethoscope location for comparison to thoracic sounds. To analyze the effectiveness of monitoring lung sounds at the throat for diagnosis of disease or physiological state, we embarked on a significant research effort, using the Virtual Human concept, to model the generation and propagation of sounds in the lung.

This problem naturally broke down into two subtasks. In the first task we modeled sound propagation and attenuation in the human body using the Virtual Human torso phantom and sound sources placed in the lung region. These were compared to analytical models for validation and verification of our sound propagation modeling approach. In the second task, we modeled sound generation in the airways using computational fluid dynamics (CFD). Once both of these are properly modeled, we intend to couple the lung sound generation to acoustic propagation, resonance, and attenuation in tissue and bone.

Sound propagation through surrounding lung tissues (parenchyma) and non-moving fluids (pleura) was modeled using the linearized inhomogeneous wave equation [24]. The tissue density and sound velocity were computed from the Hounsfield CT values for each point in the Visible Human thorax CT data. We assumed in most of our studies that there was no absorption of sound by the medium.

A parallel sound propagation code [24] that models the linearized inhomogeneous wave equation was developed as part of this project in collaboration with North Carolina State University. Based on a novel approach by Aroyan [25] this code was validated using well-known 1-D and 3-D analytical solutions for artificial sound sources. Special routines were written to read the NLM Visible Human Data into this code in a suitable form. Propagated sound data were obtained for both pulse and sinusoidal sources near the center of the lung in the thorax of the Visible Human. The sound propagation code was written using MPI for execution on the high-performance parallel computers at ORNL.

The second task involves actually modeling the generation of sound sources from the flow of air in the airways [26]. If the entire problem were solved, the computational fluid dynamics (CFD) would need to model sound generation due to vortices, turbulence, airway wall flutter, airway closure, mucosal lining effects, etc. for the entire lung. We selected to model airway flow at sub-turbulent Reynolds number (Re = 500).

Fundamental to the understanding of the various transport processes within the respiratory system, airway fluid dynamics plays an important role in biomedical research. When air flows through the respiratory tract, it is constantly changing direction through a complex system of curved and bifurcating tubes. As a result, numerical simulations of airflow through the tracheobronchial system must be capable of resolving such fluid dynamic phenomena as flow separation, recirculation, secondary flows due to centrifugal instabilities (Dean flows), and shear stress variation along the airway surface [27]. Anatomic complexities within the tracheobronchial tree, such as sharp carinal regions at asymmetric bifurcations, have motivated use of incompressible CFD codes to model flow. In our case, a serial code, PHI3D [16], was used to model airway flow. The PHI3D model was implemented

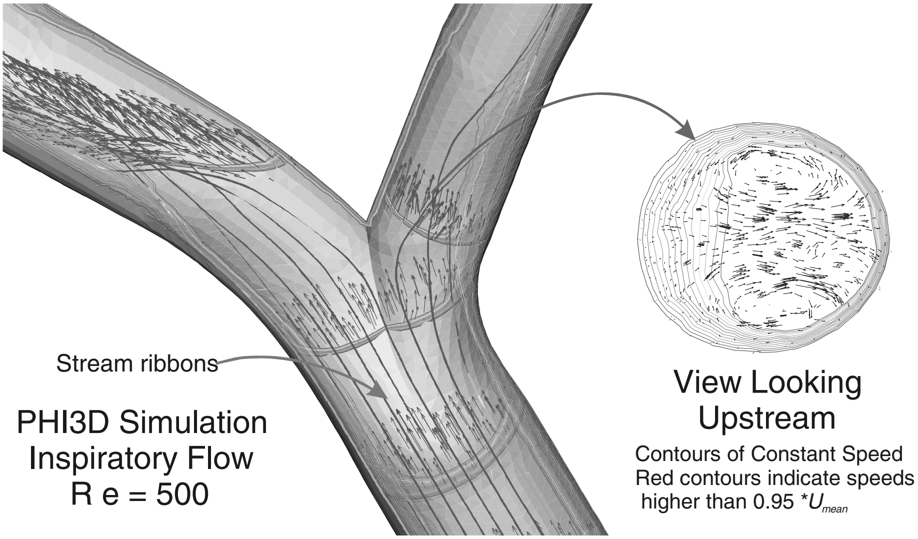

Stream ribbons
PHI3D Simulation
Inspiratory Flow
R e = 500
View Looking
Upstream
Contours of Constant Speed
Red contours indicate speeds
higher than 0.95 $*U_{mean}$

Fig. 7. Flow simulation with Reynolds number of 500 through bifurcating airway geometry

using the COWG alternative version of the VHPSE described in Section 3.3. Developed at ORNL, PHI3D implements the Continuity Constraint Method (CCM). Using a FE methodology, complex geometries can be easily simulated with PHI3D using unstructured meshes. A time-accurate integration scheme allows the simulation of both transient and steady-state flows.

The model studied was a realistic, 3-D single bifurcation geometry developed by Hammersley and Olson [27]. This model has circular cross-section in the middle straight sections of the airway tubes tapering to become more elliptical near the bifurcations. Our purpose was to verify that we could obtain the vortex flow field results using PHI3D [16] that were originally obtained by Gatlin et al. [27] with a different, finite-difference CFD code. Figure 7 shows PHI3D results for inspiratory flow. A dipole vortex flow field forms just past the bifurcation in each daughter airway. We confirmed results obtained earlier by Gatlin et al. [27].

5.3 Exposure to Environmental and Occupational Contaminants

To advance the determination of health risks associated with inhalation of environmental and occupational toxicants, we implemented a recently-developed model of inhalation of Hg vapor in the VHPSE (Fig. 8). The model incorporated the International Commission on Radiological Protection (ICRP) lung model [28]. The Hg vapor inhalation model was implemented using the Virtual Human PSE Java client interface to compute human Hg vapor uptake, retention, and excretion [29]. Since the model for Hg vapor is linear, it was implemented in Java by providing the transfer matrix using the PhysioML file.

We developed building blocks for simulation of various components of the human physiological system with the VHPSE. This approach enabled the simulation of

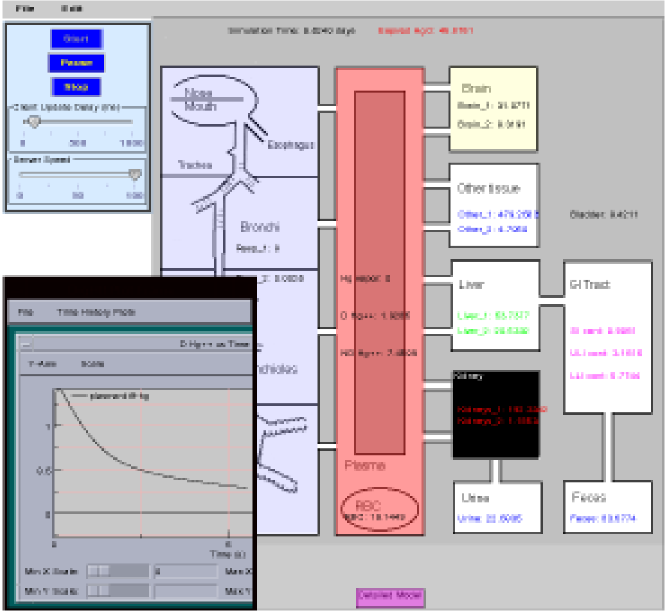


Fig. 8. Implementation of the Hg Vapor uptake, retention, and excretion compartment model in the Virtual Human PSE. The plot window in the lower left corner displays the log of Hg concentration in the plasma compartment as a function of time.

toxicant deposition in and clearance from regions of the lung for individuals of various ages, levels of physical activity, and lung functionality state – normal or compromised. The flexibility with which exposure models can be incorporated into the VHPSE and ease of modification of individual-specific model parameters will lead to greater understanding of the observed variability in deposition of airborne toxicants within the lung and the subsequent health effects (e.g., lung cancer induction).

5.4 Modeling Pulmonary Edema

The fourth legacy simulation code implemented in the VHPSE was a pulmonary edema model [30]. The original code was written in C, giving us a chance to test the use of JNI to couple to non-Java programming languages. It proved easy to integrate a non-native language code into the VHPSE. Results obtained using the VHPSE confirmed those obtained with the original code.

In our implementation of the pulmonary edema model we experimented with the capability in VHPSE to steer the simulation by altering a single parameter of the model, the venous resistance. A diagram of the pulmonary edema model was created and, as was the case with the Hg-vapor model, results of the simulation were displayed on the model diagram with placement of these results controlled by the PhysioML ***label*** element tag.

6 Virtual Soldier Project

Begun in 2004, the objective of the Defense Advanced Research Projects Agency (DARPA) Virtual Soldier Project was to utilize physiological models (both simple and complex) and experimental data to predict the location of a wound to the heart (either the left ventricle or right ventricle) resulting from a fragment wound.[15] The specific examples modeled included small fragment wounds to two different regions of the left ventricle with the medical consequences being either tamponade or exsanguination. While different software was used to implement the computational framework for Virtual Soldier, the VSPSE incorporated many of the concepts originally developed for the VHPSE, such as 3-D display of anatomy and display of the physiological results.

6.1 Virtual Soldier Holomer Concept

An important concept developed in the Virtual Soldier Project was that of the Holographic Medical Electronic Record (Holomer). The Holomer was a computational representation of all levels of medical properties of a human (molecular, biochemical, cellular, physiologic, organ, tissue and whole body), and was linked to predictive computational models, a display environment, and anatomical ontologies. Since the heart was the focus of the project, the anatomy considered in the Virtual Soldier Holomer was restricted to the heart and surrounding major vessels.

To predict the location of a wound to the heart, experimental data were evaluated using two types of computational modeling: 1) high-level integrative physiological (HIP) models (circuit models)[16] and 2) 3-D FE models, which included biochemical response, electrophysiology and mechanical motion. First the HIP models were optimized to the physiological characteristics and then results were passed, via file transfer, to the FE models. To organize the integration and display of modeling and experimental results, a PSE was developed using SCIRun [5, 6], a data flow PSE, wherein modules are connected by pipelines through which data flows. Figure 9 shows screenshots from the VSPSE, including, in the center, a typical SCIRun network used by the project. The integration of these various outputs of the SCIRun network (physiology, anatomy, and wound prediction) and its stereographic display was referred to as the Holomer.

6.2 Ontologies for the VSPSE

The Virtual Soldier Project placed considerable importance on the integration of ontologies of anatomy and physiology with the VSPSE. An ontology is a declarative, symbolic model that formalizes concepts, terminology, and abstract relationships between these concepts. It also contains logical rules on how relationships may combine concepts. All items in an ontology are called frames. An ontology is best understood as a semantic graph structured with links as edges and semantic definitions

[15] http://www.virtualsoldier.us (last accessed 6 December 2007)

[16] Integrated Cardiopulmonary Models (HIP models) of the Virtual Soldier Project http://nsr.bioeng.washington.edu/PLN/Members/mneal/integrated_html (last accessed 6 December 2007)

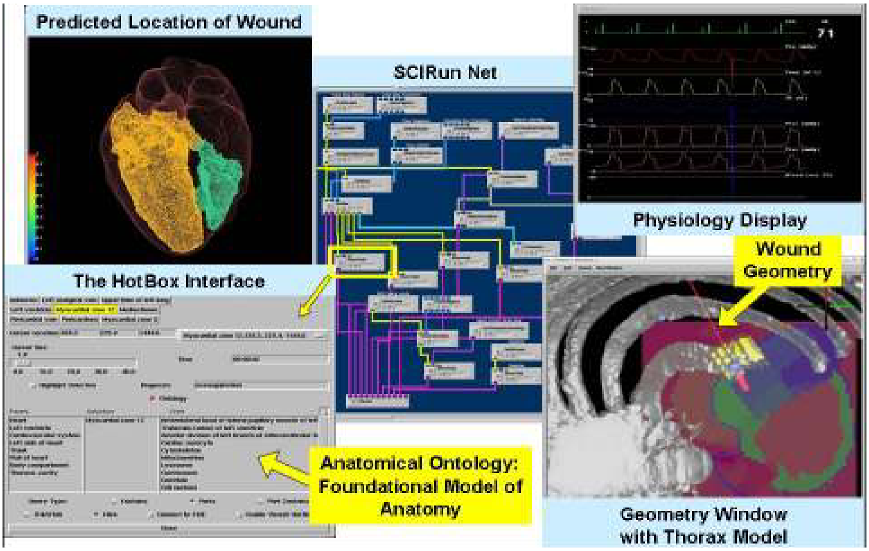
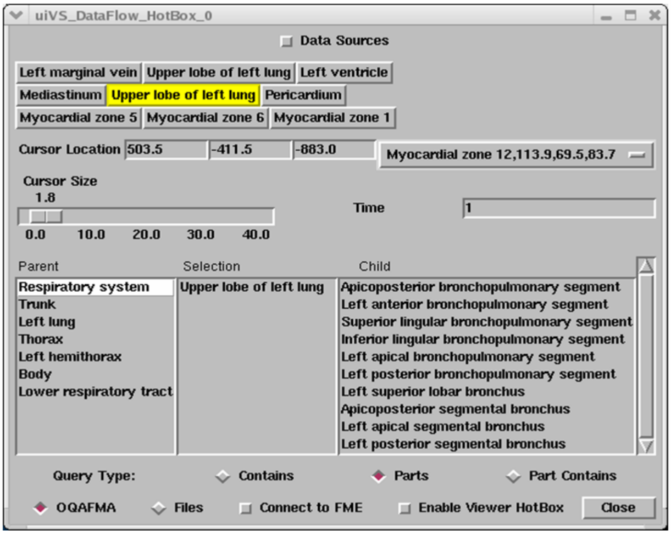

Predicted Location of Wound
SCIRun Net
Physiology Display
Wound Geometry
The HotBox Interface
Anatomical Ontology: Foundational Model of Anatomy
Geometry Window with Thorax Model

Fig. 9. Virtual Soldier PSE. The SCIRun *HotBox* UI connects physiology display, anatomical ontology, anatomical geometry, and prediction of wound location.


uiVS_DataFlow_HotBox_0

□ Data Sources

Left marginal vein	Upper lobe of left lung	Left ventricle
Mediastinum	Upper lobe of left lung	Pericardium
Myocardial zone 5	Myocardial zone 6	Myocardial zone 1

Cursor Location 503.5 -411.5 -883.0 Myocardial zone 12,113.9,69.5,83.7

Cursor Size 1.8

0.0 10.0 20.0 30.0 40.0

Time 1

Parent	Selection	Child
Respiratory system	Upper lobe of left lung	Apicoposterior bronchopulmonary segment
Trunk		Left anterior bronchopulmonary segment
Left lung		Superior lingular bronchopulmonary segment
Thorax		Inferior lingular bronchopulmonary segment
Left hemithorax		Left apical bronchopulmonary segment
Body		Left posterior bronchopulmonary segment
Lower respiratory tract		Left superior lobar bronchus
		Apicoposterior segmental bronchus
		Left apical segmental bronchus
		Left posterior segmental bronchus

Query Type: ◇ Contains ◆ Parts ◇ Part Contains

◆ OQAFMA ◇ Files □ Connect to FME □ Enable Viewer HotBox Close

Fig. 10. The anatomical ontology terms for the location upper lobe of the left lung

as nodes. The information may include text, parts of speech, concepts, images, or mathematical coordinates that are linked with each other by logical relationships described in rules. By design an ontology is both human and machine-readable, non-exhaustive and able to accommodate new types of information and concepts thanks to its formal specifications [31].

The Virtual Soldier Project used the Foundational Model of Anatomy [32] as the anatomical ontology standard. The implementation of the anatomical ontology is shown in Fig. 10. Note that the location is in the upper lobe of the left lung, the parents of this entity are shown in the lower left panel and the children are shown in the lower right panel. In addition, considerable work was done on a conceptual design and implementation of a physiological onotology to support physiological models, data acquisition, and automated reasoning.[17] The combination of ongologies for anatomy and physiology was called the Virtual Soldier Knowledge Base (VSKB).

6.3 Components of the VSPSE

The components of the VSPSE (see Fig 9) included the ability to display the results of experimental or computational modeling data in a physiology monitor (upper right). A 3-D view of the anatomy (lower right) along with the wound track indicating both destroyed and damaged tissue. The prediction of wound location was displayed (upper left) using different colors to represent the probability that the wound was to the left or right ventricle (upper left). An XML format was developed to describe fragment wounds and their location. To link visualization of anatomy to the VSKB, we developed the *HotBox* [4] (lower left).

The *HotBox*: A SCIRun module, the *HotBox*, facilitated interaction between 1) the VSKB ontologies and the geometric anatomical models and 2) the anatomy and associated physiology. Our objective was to display this information in such a way as to capture the 3-D nature of the human body and to correlate that with extensive information on both the anatomy and the physiology of the wounded soldier [4]. The Virtual Soldier *HotBox* UI implements the concept of *deep voxel* wherein a specific geometrical feature in the anatomical model at a given point in space is linked to the structural knowledge ontology at that particular location. In addition, the *HotBox* connects the anatomical geometry to the physiological state of the individual (the vital signs) and to a description of the wounds, if the individual is wounded.

Display and Identification of Wounds: A visual display of the wound track produced by the fragment was accomplished using simple geometric icons. The geometry and information about the wound was specified in an ontology (using Protégé[18]) and a corresponding XML format, a standard for which was developed by Stanford University and the University of Washington[19]. The icon depicting a wound is a set of primitives (cylinders of various diameter) representing the wound track through the tissue. Both ablated tissue (i.e., tissue removed) and "stunned tissue" (i.e., altered tissue) are displayed. The display of the wound damage track and the information regarding the wound was controlled by the *HotBox* SCIRun module.

Web Services: The project investigated a new PSE concept, implemented around a services-oriented architecture (SOA), which would incorporate data repositories,

[17] Cook, D. Ontology-Based Symbolic Models of Anatomy, Physiology and Pathology: Auto-generation of Mathematical Models? http://nbcr.net/physiome/presentations/Cook.pdf (last accessed 6 December 2007)

[18] http://protege.stanford.edu (last accessed 6 December 2007)

[19] http://nbcr.net/physiome/presentations/Cook.pdf (last accessed 6 December 2007)

computational engines, and visualization capabilities in a seamless, Internet-connected architecture. Integral to this design concept was the use of Web services to provide model components via a generalized request encoded in the Web Services Definition Language (WSDL).

ORNL and its partner, the Center for Information Technology at the University of South Carolina, developed middleware components of this SOA including Web services for the Virtual Soldier data repository, a client for the Web service for the VSKB, and Web services and associated client application programmer interface for the HIP model computations. The SOA provides a more sophisticated approach to human modeling and simulation, one that takes advantage of computing resources widely distributed across the Internet and which allows for seamless integration of model components provided by widely dispersed research teams. The future of human modeling and simulation depends on extending the development of this futuristic concept.

7 Lessons Learned

By examining the different approaches taken in developing the Virtual Human and Virtual Soldier PSEs, we learn lessons about what are effective approaches to the design of PSEs for human modeling and simulation. In this comparison it should be noted that a difference of more than five years separated the development of the VHPSE and the VSPSE, which is a significant time for changes in both the programming languages utilized and in advances in the conceptual design of PSEs.

Both environments emphasize the importance of displaying 3-D anatomical imagery for intuitive understanding of normal function or effects of injury or disease. The VHPSE utilized NURBS representations for 3-D rendering (on a flat computer monitor) of data transmitted remotely over the Web. The VSPSE utilized stereographic capability built into SCIRun to render the display of high-resolution anatomy. In addition, each voxel of the anatomical data in the VSPSE was identified using an anatomical ontology, the Foundational Model of Anatomy [32]. The link between a visual environment and an anatomical ontology made possible the connection of anatomical identification to physiological and pathophysiological response.

Both environments had capabilities to display physiological results. In the case of the VHPSE, our intent was to connect models written in different programming languages and hosted in different locations in a seamless, remotely coupled operation, through the use of Java RMI and JNI. This was tested only to a limited extent and we look forward to future, more detailed studies of the potential of this design. The VSPSE was designed around a very powerful, integrated data-flow PSE, SCIRun, which included the ability to handle computations and visualization of complex FE models. In addition, a SCIRun module was built to display the results of the HIP models. However, due to the complexity of the computations carried out at different institutions, there was no use in this project of the SCIRun platform to perform integrated computations.

A significant difference in the two environments was their utilization of distributed computing resources. At the time that VHPSE was being developed, the concept of grid computing was in its infancy. However, there were tools that provided access to distributed computing resources, one of which, NetSolve, was incorporated into one version of the VHPSE as noted earlier. In addition, as described in Section 3, an alternative CORBA-based VHPSE environment was developed which demonstrated yet another means for remote utilization of distributed computing resources within a PSE. While the Virtual Solder Project did not use integrated distributed computing, a future Virtual Human PSE should take full advantage of distributed computing, possibly by utilization of the Common Component Architecture (CCA) framework [33], which is part of SCIRun.

While the two environments had common features, the VSPSE more completely satisfied the general characteristics of PSEs in Section 3. It provided access to knowledge bases of anatomical and physiological data and geometric models of the human body; it used a data flow design with modules that connect the output of one model to the input of another model. Neither environment truly had a GUI that allowed model building using icons, but both provided a user-friendly GUI to facilitate evaluation of results from computations. Utilizing Tcl/Tk[20], VSPSE provided user control over data flow from computation modules to output analysis modules and, finally, to visualization modules. Built using Java, the strengths of the VHPSE were its focus on integration and interoperability for physiological models written in different programming languages and on the ability to steer the computations from the GUI.

Developing the Virtual Human PSE and working with a much larger team to implement a similar environment for the Virtual Soldier Project has given us insights for improving PSEs for human modeling and simulation.

1) Future multi-scale human modeling and simulation will require the integration of different modeling approaches, one based on discrete information (e.g., stoichiometric biochemical reactions) and one based on continuous, time-dependent simulation (e.g., differential equation-based systemic organ models). Hybrid modeling, an approach which integrates discrete-event simulation [34] with continuous modeling, could prove very useful for multi-scale computations where there is *loose coupling* between “fast” and “slow” states in the system being modeled [34].

2) Common standard model formats, such as SBML, should be expanded to include additional concepts present in ORNL PhysioML, namely implementation of model visualization and display of simulation results. By using SBML, with extensions from PhysioML, the input to the virtual human PSE could provide the model descriptions (equations), model parameters, initial conditions, and display characterization.

3) Developing a standard ontology and associated XML description for injuries and diseases would allow full characterization of models in situations of pathophysiological response, as pioneered in the Virtual Soldier Project.

4) Driven by the commercial sector, new approaches to distributed computing have emerged which would be better suited for implementation of virtual human PSEs. An example of such approach was investigated for the Virtual Soldier Project and Web services to support it were developed. Further development of this approach will significantly improve our ability to solve the complex problems posed by human modeling and simulation.

[20] http://tcl.sourceforge.net (last accessed 6 December 2007)

8 Summary

In conclusion, we have presented a brief historical account of development of integrative human modeling, using the ORNL Virtual Human and the DARPA Virtual Soldier Projects as examples. We described aspects of the VHPSE, including interoperability between models written in different programming languages and the ability for the user to perform remote computational steering to change model parameters. We described aspects of the VSPSE, including the concept of the Holomer and the implementation of the *HotBox,* a UI for connecting anatomical, ontological and physiological information. The lessons learned from implementing various physiological models within these two environments were discussed and we outlined the difficulties faced by these projects in attaining a truly integrated human modeling and simulation environment. Finally, we have addressed some future developments that will improve PSEs for human modeling and simulation. These include: 1) integrating discrete event and continuous modeling, 2) developing an enhanced version of SBML for physiological modeling, 3) developing a standard ontology and associated XML description for wounds, and 4) implementing new developments in distributed computing. By incorporating these improvements, the goal of a truly integrated computational environment for human modeling will be a reality in the not-too-distant future.

References

1. Krause, C.: The Virtual Human Project: An Idea Whose Time Has Come? ORNL Review 33, 1 (2000)
2. Easterly, C.E., Allgood, G.O., Eckerman, K., Knee, H., Maston, M., McNeilly, G., Munro, J., Munro, N., Toedte, R., Van Hoy, B., Ward, R.: Virtual Human: A Diagnostic Tool for Human Studies and Health Effects in the 21st Century. In: Leiberman, R.A., Vo-Dinh, T. (eds.) Proceedings of SPIE Biomedical Sensing and Imaging Technologies, vol. 3253, pp. 150–154 (1998)
3. Li, M., Walker, D.W., Rana, O.F., Huang, Y., Williams, P.T., Ward, R.C.: Engineering High-Performance Legacy Codes as CORBA Components for Problem-Solving Environments. J. Parallel Distrib. Comput. 63, 1152–1163 (2003)
4. Ward, R.C., Pouchard, L.C., Beckerman, B.G., Dickson, S.P.: The HotBox: a Visual User Interface to Medical Data. Information Visualization 5(4), 290–295 (2006)
5. Johnson, C.R., Parker, S., Weinstein, D., Heffernan., S.: Component-Based Problem Solving Environments for Large-Scale Scientific Computing. J. Conc. & Comp.: Prac. & Exper. 14, 1337–1349 (2002)
6. Parker, S.G., Weinstein, D.M., Johnson, C.R.: The SCIRun Computational Steering Software System. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (eds.) Modern Software Tools in Scientific Computing, pp. 1–40. Birkhauser, Boston (1997)
7. Cristy, M., Eckerman, K.F.: Specific Absorbed Fractions of Energy at Various Ages from Internal Photon Sources, ORNL/TM-8381/V1-V7, Oak Ridge, Tenn (1987)
8. Snyder, W. S., Ford, M. R., Warner, G. G., and Fisher, Jr., H. L.: MIRD Pamphlet No.5 - Estimates of Absorbed Fractions for Monoenergetic Photon Sources Uniformly Distributed in Various Organs of a Heterogeneous Phantom. J. Nucl. Med. (suppl. 3), 8 (1969)

9. International Commission on Radiological Protection. Report of the Task Group on Reference Man. ICRP Publication 23. Pergamon Press, Oxford (1975)
10. Arnold, D., Argawal, S., Blackford, S., Dongarra, J., Miller, M., Sagi, K., Shi, Z., Vadhiyar, S.: Users' Guide to NetSolve V1.4, Innovative Computing Laboratory, UT Computer Science Dept (2001)
11. Ward, R.C., Strickler, D.J., Tolliver, J.S., Easterly, C.E.: A Java User Interface for the Virtual Human. In: Proceedings of First Joint Conference of IEEE BMES/EMBS, Atlanta, GA, p. 1211 (1999)
12. White, S., Fisher, M., Cattell, R., Hamilton, G., Hapner, M.: JDBC API Tutorial and Reference, 2nd edn. Addison-Wesley, Boston (1999)
13. Tolliver, J.S.: Science Portals using Java RMI for Access to Legacy Scientific Simulation Codes. Poster presentation at Java Grande Conference (2000)
14. Ball, S.J., Nypaver, D.J.: GRSAC Users Manual, ORNL/TM-13697, Oak Ridge Nat. Lab., Oak Ridge, TN (1999)
15. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison-Wesley, Boston (1995)
16. Williams, P.T., Baker, A.J.: Incompressible Computational Fluid Dynamics and the Continuity Constraint Method for the 3-D Navier-Stokes Equations. Numerical Heat Transfer, Part B Fundamentals 29, 137–273 (1996)
17. Ward, R.C., Kruse, K.L., Allgood, G.O., Hively, L.M., Fischer, K.N., Munro, N.B., Easterly, C.E.: Virtual Human Project. In: Allgood, G.O., Faust, N.L. (eds.) Proceedings of the SPIE Visualization of Temporal and Spatial Data for Civilian and Defense Applications, vol. 4368, pp. 158–167 (2001)
18. Walsh, N.: What is XML? XML Com (1998), http://www.xml.com/pub/98/10/guide1.html
19. Kleinman, R.: Ask Expresso Man. Java Developer Connection, (2000 June 21) (last accessed December 6, 2007), http://java.sun.com/developer/technicalArticles/xml/expressomanxml.html
20. https://jaxp.dev.java.net/ (last accessed December 6, 2007) and Java API for XML Processing. Wikipedia, http://en.wikipedia.org/wiki/Java_API_for_XML_Processing (last accessed December 6, 2007)
21. Ward, R.C., Pouchard, L.C., Nutaro, J.J.: Integrative Computational Frameworks for Multiscale Digital Human Modeling and Simulation. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3994, pp. 814–821. Springer, Heidelberg (2006)
22. Ward, R.C.: An Information Management Approach to Describing Virtual Human Models. In: Biomedical Engineering Society Meeting, Seattle, WA (October 2000)
23. Rideout, V.C.: Mathematical and Computer Modeling of Physiological Systems. Prentice Hall, New Jersey (1991)
24. Narasimhan, C., Ward, R.C., Kruse, K.L., Guddati, M., Mahinthakumar, G.: A High Resolution Computer Model for Sound Propagation in the Human Thorax Based on the Visible Human Data Set. Computers in Biology and Medicine 34, 177–192 (2004)
25. Aroyan, J.L.: Three-dimensional Numerical Simulation of Biosonar Signal Emission and Reception in the Common Dolphin. Ph.D. Dissertation, U.C. Santa Cruz (1996)
26. Kruse, K.L., Williams, P.T., Allgood, G.O., Ward, R.C., Gleason, S.S., Paulus, M.J., Munro, N.B., Mahinthakumar, G., Narasimhan, C., Hammersley, J.R., Olson, D.E.: Flow Simulation in a 3-D Model of Pig Airways and Connection to Lung Sounds. In: Allgood, G.O., Faust, N.L. (eds.) Proceedings of the SPIE Visualization of Temporal and Spatial Data for Civilian and Defense Applications, vol. 4368, pp. 168–176 (2001)

27. Gatlin, B., Cuicchi, C., Hammersley, J., Olson, D., Reddy, R., Burnside, G.: Computation of Converging and Diverging Flow Through an Asymmetric Tubular Bifurcation. In: Conference Proceedings of the 1997 ASME Fluids Engineering Division Summer Meeting, FEDSM 1997, Vancouver, BC, Canada, June 22-26 (1997)
28. ICRP International Commission on Radiological Protection. Human Respiratory Tract Model for Radiological Protection, ICRP Publication 66. Pergamon Press, Oxford (1994)
29. Leggett, R.W., Munro, N.B., Eckerman, K.F.: A Proposed Revision of the ICRP's Model for Inhaled Mercury Vapor. Health Phys. 81(4), 450–455 (2001)
30. Roselli, R.J., Tack, G., Harris, T.R.: A Model of Fluid, Erythrocyte, and Solute Transport in the Lung. Ann. Biomed. Eng. 25, 46–61 (1997)
31. Pouchard, L.C., Dickson, S.P.: Ontology-Enriched Visualization of Human Anatomy. pp. 2–3 ORNL/TM-2005/113, Oak Ridge Nat. Lab., Oak Ridge, TN (2005)
32. Rosse, C., Mejino, J.L.V.: Ontology for Bioinformatics: The Foundational Model of Anatomy. Journal of Biomedical Informatics 36, 478–500 (2003)
33. Common Component Architecture. Oak Ridge Nat. Lab., Oak Ridge, TN (2001) (last accessed 6 (December 2007), http://www.csm.ornl.gov/cca
34. Nutaro, J.J.: Discrete Event Simulation of Continuous Systems. In: Fishwick, P. (ed.) Handbook of Dynamic Systems Modeling, Chapman&Hall/CRC, Boca Raton (2007)