

## LINKING SIMULATION AND VISUALIZATION CONSTRUCTION THROUGH INTERACTIONS WITH AN ONTOLOGY VISUALIZATION

Zach Ezzell  
Paul A. Fishwick

CISE Department  
University of Florida  
Gainesville, Florida USA

Juan Cendan

College of Medicine  
University of Central Florida  
Orlando, Florida USA

### ABSTRACT

An ontology is a formalized knowledge structure understandable by humans and machines. Positioned within the interface layer, domain-specific ontologies can afford simulation model building and visualization construction. Such an ontology-enabled interface would allow modelers to interact with the semantics they are already familiar with, due to their field-specific education and training, in order to build executable simulation models. We present a methodology in which ontology visualizations serve as interface to simulation model building and visualization construction activities. Further, we describe how the ontology can be used to link simulation variables to visualization parameters, thus supporting integrative multi-modeling by allowing simulations and their corresponding visualizations to be constructed within the same interface and interaction paradigm. To demonstrate the ontology-enabled interface, we present a case study: a physiological simulation of hypovolemic shock and its corresponding three-dimensional (3D) visualization.

### 1 INTRODUCTION

Computer simulations serve as an essential aid to those working in many scientific and engineering domains, because they can help predict the outcome of real-world scenarios. Further, visualizations based on simulation output are useful when analyzing simulation results and communicating ideas between peers or invested third parties (e.g., management, students). State-of-the-art simulation software tools, however, rely heavily on metaphors from engineering paradigms (e.g., block integrators, Matlab scripts, etc). Understanding the rules for simulation model building within these paradigms requires substantial training. Further, constructing a visualization that is based on simulation output requires a separate interface. For example, in Simulink (Simulink 2011), a model builder interface is used to construct the executable simulation model, a Virtual Reality Markup Language (VRML) editor is used to create a 3D visualization, and a third graphical interface is used to connect simulation variables to visualization parameters (e.g., the position or color of 3D meshes).

We address the issues caused by engineering model edifice and disjoint simulation and visualization authoring interfaces through a methodology in which ontologies are integrated into the simulation model building and visualization construction pipeline. In our methodology, the ontology is leveraged on a computational level and also visualized and used as an affordance in the interface layer. Ontology was defined by Gruber (1993) as a *formalization of a conceptualization*. For the purposes of this paper, let us use a more narrow definition of a *collection of concepts supplemented with semantic relations that further define how concepts are interrelated* (Tolk and Turnista 2007). A major benefit of the ontological structure is its ability to express semantics from an arbitrary domain (e.g., medicine or ecology) in a machine

understandable format consisting of concept, attributes, and relationships. If this semantic structure can be manifested in the interface layer of software tools, then it can provide a familiar starting point for domain specialists with a goal to construct simulations and visualize the simulation output to analyze and communicate results.

The methodology defined in this paper was originally presented in Ezzell et al. (2011). In this paper, we present refinements on the methodology which include: defining precisely how simulation variables can be linked through the ontology to visualization parameters so that when the simulation model executes, the visualization will animate; and expanding the visualization of relationships to allow for arbitrary 3D curves with more expressive potential. We also present a case study which details the construction of a simulation and corresponding 3D visualization of the physiological phenomenon of hypovolemic shock using a software prototype. The prototype employs the principles of integrative multimodeling to link graphical, dynamical (simulation), and knowledge models in the interface layer. These three models are encoded in an ontology and, in turn, a graph-based visualization of the ontology is leveraged as an interface affordance.

The remainder of this paper is structured as follows. Background on integrative multimodeling and ontology-based simulation frameworks is given in Section 2. In Section 3, the theoretical methodology of leveraging ontology visualizations as an interface to simulation and visualization functions is presented. Our hypovolemic shock case study is presented in Section 4. Finally, conclusions and future work are presented in Section 5.

## **2 BACKGROUND**

The work presented in this paper draws from previous work in simulation and integrative multimodeling. This section is dedicated to presenting background in both of these areas.

### **2.1 Simulation Ontologies and Ontology-Based Simulation Frameworks**

Berners-Lee et al. (2001) envisioned the Semantic Web, an evolution of the Internet that is encoded with machine understandable semantics. A key element of the Semantic Web is the ontology, which has adopted by many disciplines. In biological sciences, for instance, the Gene Ontology Project, created to aid in the construction of a unified vocabulary across different biological domains, has been successful and continues to grow (The Gene Ontology Consortium 2010). Other examples include a port ontology for use in mechanical engineering applications (Liang and Paredis 2003), and an anatomy ontology (Rosse and Mejino 2007), which is leveraged in our work. The remainder of this section is dedicated to presenting a background of those ontologies and ontology-based frameworks created in the domain of modeling and simulation.

A prime example of an ontology-based system in simulation is the Framework for Adaptive Modeling and Ontology-driven Simulation (FAMOS) (Benjamin and Graul 2006). Models in FAMOS are composed using components from both simulation and domain-specific ontologies. Benjamin et al. (2006) describe several ways in which ontologically-centered design is beneficial to simulation model development: (1) ontologies can aid in model construction by clearly and unambiguously separating levels of abstraction and determining simulation model objects, structure, and logic; (2) they can help identify data sources and can inform data mining techniques by disambiguating information gleaned from textual descriptions; and (3) they serve to harmonize terminology, promote a common knowledge and language, and help relate organizational goals to simulation goals.

In an effort to promote a common knowledge and language, Miller et al. (2004) formalized a set of agreed upon simulation concepts in the Discrete-event Modeling Ontology (DeMO). The semantics embedded in DeMO permit a certain amount of automation in simulation application development and execution. Since its deployment, others in the field have leveraged DeMO in support of their simulation applications. Silver et al. (2007) employed DeMO in the Ontology-Driven Simulation (ODS) framework. ODS provides an interface that allows users to map concepts in a given domain (e.g., healthcare, military)

to simulation concepts in DeMO. After this semantic mapping is established, ODS can be used to create executable simulation code that can run in the Java-based simulation environment JSIM (Miller et al. 1999) or the commercial simulation package ARENA.

Bhatt et al. (2004) observed that synthetic environments (e.g., a virtual town square) have many applications in simulation but can be expensive to recreate for use in different simulation packages. To this end, they created an ontology based on the Synthetic Environment Data Representation and Interchange Specification (SEDRIS). To encode SEDRIS in the Web Ontology Language (Knublauch et al. 2004), a novel translation tool was created, which resulted in the Synthetic Environment Data representation ontology (sedOnto) (Bhatt et al. 2005). Lacy and Gerber (2004) also used ontologies to address the issue of open information exchange by creating the Process Interaction Modeling Ontology for Discrete Event Simulation (PIMODES). PIMODES is a vendor-neutral intermediate model format encoded in OWL in support of the open interchange of simulation models between commercial simulation packages.

## 2.2 Integrative Multimodeling

Fishwick (2004) presented the challenge of integrative multimodeling, which calls for a closer linkage between the abstract model and the phenomena which it represents. One way to achieve this is to graphically represent the abstract model and phenomena within one cohesive interface. In this section, two previous research efforts that employ integrative multimodeling are presented.

An early application of the principles of integrative multimodeling is the work done by Park and Fishwick (2005). A framework was created that allowed for the integration of different model types within a 3D simulation environment. Model types included dynamic, interaction, ontological, and geometric models. The framework provides a Model Explorer for dynamic modeling and an Ontology Explorer with OWL integration so users can create OWL classes and subclasses. Users can create geometry using Blender (Blender 3D 2011) or import pre-existing meshes. Dynamic model components (e.g., blocks within a Functional Block Model) can be selected from RUBE (Hopkins and Fishwick 2003) and associated with geometry in the scene. Using Blender Game Engine functionality, an interaction model can be created and linked with geometry.

A more recent attempt to apply integrative multi-modeling was in support of training for anesthesia machine operation. A mixed reality interface (Ishii 2008) was created by Quarles et al. (2010) called the Augmented Anesthesia Machine (AAM). Within the AAM, abstract models used to teach anesthesia machine operation are collocated with the components on the physical anesthesia machine. User studies were performed that compared the AAM to traditional training methods (e.g., using only an instruction manual and the anesthesia machine) and web-applets that use abstract visualizations (Fischler et al. 2008). The results of the study showed that the AAM was most successful at bridging the gap between abstract and concrete knowledge, thus proving the approach is a potentially viable educational technique which should be explored further.

## 3 THE METHODOLOGY

The focus of this work is to define an interface to simulation and visualization functions that utilizes a graphical representation of the domain-ontology. The domain-ontology interface should leverage pre-existing relational semantics familiar to practitioners in a given domain and, in most cases, should not require extensive training on modeling metaphors and practices. For an example of such semantics, consider a practitioner in the medical domain. They most likely, for instance, understand the relational semantics of compartments of the human body with respect to blood flow (e.g., the dynamic relation: the right atrium *flows to* the right ventricle).

A methodology for leveraging domain-ontologies was originally presented in Ezzell et al. (2011). In this section, a refined version of the methodology is presented. Refinements include: allowing visualizations of ontological relationships to trace arbitrary 3D paths; and adding the concept of *influences* to link simulation variables to visualization parameters. The following sub-sections describe high-level func-

tionality requirements that together define a methodology. A prototype that implements this methodology as part of a case study is presented in Section 4.

### 3.1 Domain-Ontology Acquisition

In order to leverage an ontology visualization as an interface affordance, one must first obtain an ontology encoded with the desired semantics. Naturally, there are two ways to fulfill this requirement. The first is to seek out a pre-existing ontology which may be published by experts in the given field. For example, there is the Modelica ontology (Pop et al. 2004) that supports simulation of physical systems (e.g., those containing electrical, thermal, or hydraulic components). Other pre-existing ontologies include those that are intra-organizational and often not publically-accessible. These ontologies enable knowledge sharing and reuse across departments within a single organization (Blomqvist and Öhgren 2008). The second option is to simply construct the ontology oneself. In turn, the newly constructed ontology can be used as a starting point in future efforts.

If a pre-existing ontology is employed, then it should be analyzed and modified if the semantics required by the modeler are not already present. Examples of modifications include adding new sub-concepts (i.e., adding semantic precession) necessary for the desired simulation and visualization, adding entirely new concepts, or adding relationships between pre-existing concepts.

### 3.2 Simulation Model Building and Visualization Construction

To eventually execute a simulation and corresponding visualization, system dynamics need to be defined. To achieve this, relationships specific to simulation dynamics should be added between existing concepts in the ontology, if they are not already present. If more than one relationship type can denote dynamic flow, then the specific dynamics of interest need to be defined to avoid ambiguities when it is time to execute the model.

Concurrently with simulation model building, expressive 3D visualization can be achieved by adding visual parameters as attributes to ontology concepts. Examples of visualization attributes include 3D meshes and materials. When a visualization attribute is added to an ontology concept, it should be rendered in the ontology visualization and rooted at the node of the given concepts. Further, influences can be added to ontology members to link visualization attributes to simulation variables (e.g., attribute *A influences* attribute B, where A is a dynamic variable and B is a visualization parameter).

As a final construction requirement of the methodology, the graph-based ontology visualization itself can also be modified in order to express certain ideas. This requirement supports integrative multimodeling. For example, the visualization of a relationship denoting dynamic flow can be morphed into a curve that traces the path of the flow within the 3D visualization environment.

### 3.3 Execution

The resulting model can be executed once the needed ontological concepts are established, and the simulation model and visualization attributes are defined within the ontology. The method of execution will depend on the mathematical paradigm in which a given domain-ontology exists. For example, in physiology simulation models, blood flow is represented by the hydrodynamic metaphor where blood volume in compartments is updated by calculating pressures, resistances, and valves between adjacent compartments. The execution requirement implies that the specific solver connected to the interface depends on the desired application domain. As the defined simulation model is executed, simulation parameters should be updated at each time interval. In turn, visualization parameters that are linked to simulation attributes through influences are updated. This results in a real-time 3D animation based on simulation output.

## 4 A CASE STUDY: HYPOVOLEMIC SHOCK

We built a software prototype that adheres to the methodology presented in Section 3. In order to demonstrate the ontology-based methodology, we used the prototype to construct a simulation and visualization of hypovolemic shock. Hypovolemic shock is a type of shock resulting from severe blood loss. Symptoms of hypovolemic shock include vasoconstriction (resulting in a de-coloring of the skin), an increased heart rate, and a weakened pulse (Lawrence et al. 2006). This section is dedicated to describing the software prototype and detailing the process of using it to create an executable simulation and visualization of hypovolemic shock.

### 4.1 Technical Specifications

The software prototype created can be categorized as an ontology visualizer and editor, and a simulation model and visualization builder. The prototype was coded in C++ and leverages the Open-source Graphics Rendering Engine (OGRE) (OGRE 3D 2011) for managing the rendering of textured 3D geometry. A high-level depiction of the architecture of the software prototype is shown in Figure 1.

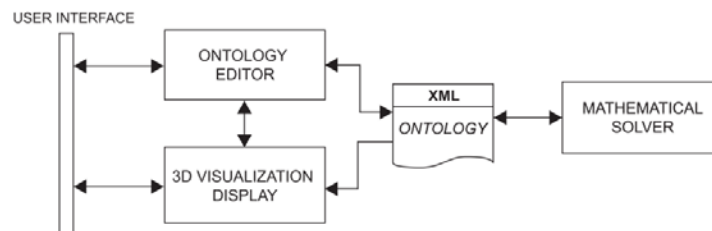


Figure 1: High level view of the architecture of the software prototype.

Many cardiovascular simulation models utilize the hydrodynamics metaphor through compartmental modeling (Jacquez 1996). In such models, chambers of the cardiovascular systems are represented by a series of inter-connected compartments. Inter-connections are supplemented with valves and resistances to modify flow; and blood levels are governed over time by differential equations that account for pressure, flow, valves and resistances. Blood is propagated through the compartments by a periodic pump function connected to the heart compartments. The equation solver in our prototype solves the network of dynamic relationship according to the hydrodynamics metaphor. Ontological concepts are given attributes that are in turn used as coefficients in the flow equations, which are solved using forward Euler's method.

### 4.2 Constructing the Simulation Model and Base Visualization

To begin construction, we needed to acquire a domain-ontology or build one ourselves (as defined in the methodology in Section 3.1). We decided to leverage a portion of the Foundational Model of Human Anatomy ontology (FMHAO) (Rosse and Mejino 2007). The sub-ontology within FMHAO that we leverage, shown in Figure 2, is related to cardiovascular function. Additional concepts of *intrathoracic arterial tree*, *extrathoracic arterial tree*, *intrathoracic venous tree*, and *extrathoracic venous tree* were added to this sub-ontology to further compartmentalize the concepts *systemic arterial tree* and *systemic venous tree*. Figure 3a shows the base ontology and the four additional concepts as visualized in our prototype's interface. The result of assigning *mesh* and *material* attributes to the concepts *human body* and *heart* is shown in Figure 3b. The assignment of these visualization attributes marks the transition to the simulation and visualization construction phase of the methodology (defined in Section 3.2). Meshes and material are required to be stored in the OGRE format. To allow meshes to be moved once they are loaded, a

mover widget will appear rooted at the visualization of a selected ontology node that allows users to reposition the meshes in 3D space.

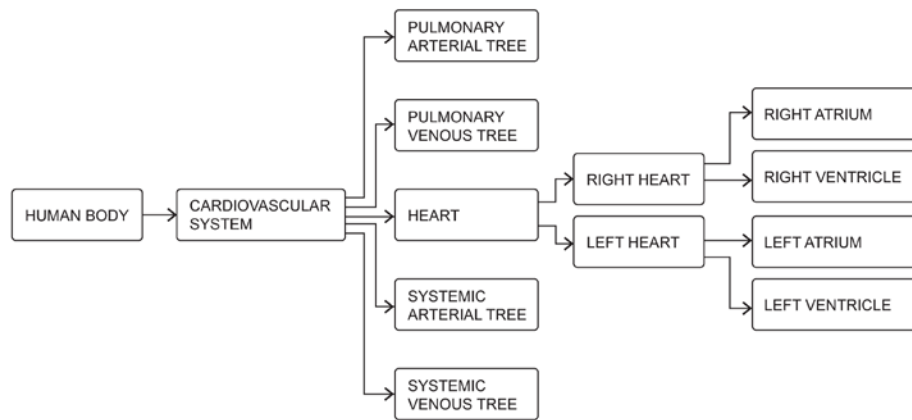


Figure 2: A portion of the Foundational Model of Anatomy Ontology. The *human body* concept serves as the root of the sub-ontology and all relationships shown are of the *has a type*.

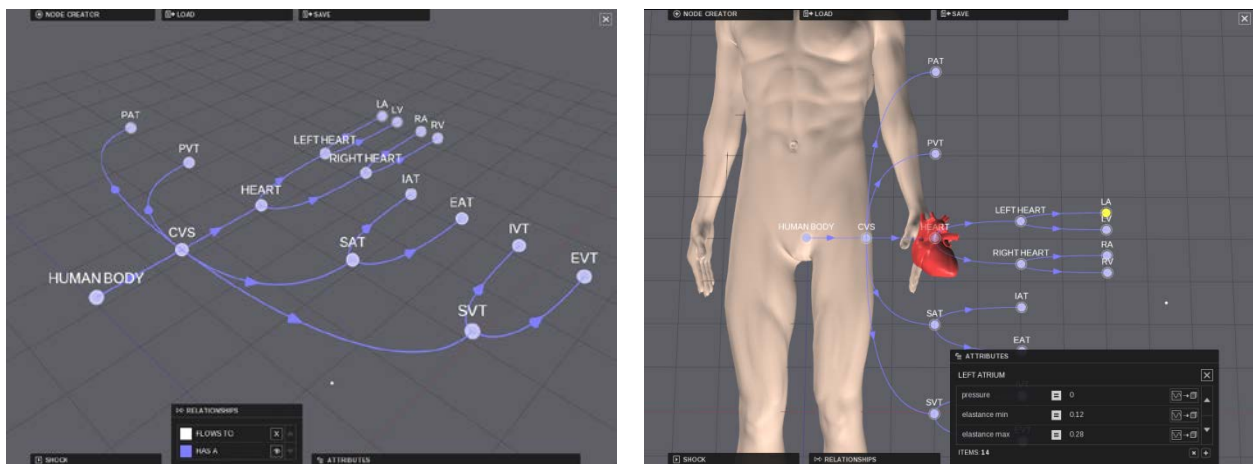


Figure 3a (left): A portion of the Foundational Model of Anatomy Ontology visualized in our software prototype. The abbreviations used in the prototype to avoid clutter are CVS: Cardiovascular system, PAT: pulmonary arterial tree, PVT: pulmonary venous tree, SAT: Systemic arterial tree, SVT: systemic venous tree, IAT: intrathoracic arterial tree, EAT: extrathoracic arterial tree, IVT: intrathoracic venous tree, EVT: extrathoracic venous tree, LA: left atrium, LV: left ventricle, RA: right atrium, RV: right ventricle. Figure 3b (right): The result of assigning *mesh* and *material* attributes to the concepts human body and heart. The attribute editor is expanded and shows three simulation variables that were added as attributes to the concept left atrium.

Along with the assigning of visualization parameters to ontology concepts, simulation variables need to be defined. In our case study, simulation variables were taken from the cardiovascular model created by Beneken (1965). Each compartment in the model has a blood volume level that is updated according to pressure and inward flow. Flow is calculated by considering resistances between compartments and valves. Figure 3b shows the expanded Attribute Editor with attributes *pressure*, *elastance min*, *elastance max* which were added to *left atrium*. The Attributes Editor allows the users to add and remove attributes to ontology concepts. In order for the mathematical solver to execute the simulation

model, all the necessary simulation attributes must be added to the concepts of the cardiovascular model. Further, relationships of type *flows to* must be added between the compartments in order to determine the cardiovascular network that is to be solved when the simulation is executed.

Resistance between compartments is a required variable for the hydrodynamic equation solver. Instead of requiring an *out-resistance* or *in-resistance* attribute be added to the ontology concepts, it seems natural to link this variable to the actual relationship connecting the compartments, as this relationship visually represents the flow. To this end, our software prototype allows attributes to be assigned to relationships as well. Resistance values for the Beneken model are defined as attributes added to the relationships connecting the 10 compartments. This is achieved by selecting the relationship and adding attributes using the Attribute Editor. The attribute *has valve* is also added to *flows to* relationships to signify a valve exists between two compartments.

Figure 4a shows the 10 compartment Beneken model collocated within the 3D visualization. Ontology concepts are positioned by the user and collocated with their respective locations on the 3D heart mesh. In this example, compartments representing sprawling, non-centralized anatomy are placed within the visualization in a symbolic fashion (e.g., *intrathoracic arterial tree* is placed somewhere within the thorax). As also depicted in Figure 4a, ontological relationships can be collocated within the 3D visualization. This is possible because relationships are rendered as splines, which are smooth interpolations of piece-wise linear polynomials (Bartels et al. 1983). In Figure 4a, the spline control hull is visible for the relationship connecting the *intrathoracic venous tree (IVT)* to the *right atrium (RA)*. Control points (the white spheres) on the curve's hull can be selected and positioned in the 3D space and the curvature of the relationship will update accordingly. Representing relationships as 3D curves allows the ontology visualization to positionally express dynamic flow between ontology members (e.g., the relationships trace the 3D path of blood flow through the heart).

With the appropriate variables added to the ontology and the cardiovascular network defined, the simulation model can be executed (execution is defined in the methodology in Section 3.3). This is done by pressing play in the Simulation Controller (shown in the bottom left corner of Figure 4b). Simulation variables can be selected from the Attribute Editor and placed within the visualization. This is also demonstrated in Figure 4b, where the *pressure* attribute of the *left ventricle* was placed in the 3D scene, resulting in a plot anchored to the left ventricle ontology concept. The *volume* attribute was then dragged to the x-axis of the *pressure* plot to replace the default x value of *time*. This resulted in the pressure-volume loop during simulation execution shown in Figure 5.

The final step of constructing the simulation model of hypovolemic shock is to introduce blood loss to the system. A linear rate of change can be introduced to any variable by adding attributes to ontology members of the form *variable\_name decrease rate* or *variable\_name increase rate*. *Variable\_name* should be the name of a variable that is defined within the same concept as the rate of change. In our case study, we added *volume decrease rate* to the concept *intrathoracic arterial tree (IAT)* and set its value to 25. This results in the blood volume in the intrathoracic arterial tree to be drained at a rate of 25 milliliters per second during simulation execution.

The introduction of blood loss to simulate hemorrhaging during shock elucidates a short-coming of the current implementation of attribute influences. In medical practice, recognizing shock in a patient can be difficult and requires developing insight into the possible underlying mechanism, consideration of potential injuries, as well as recognition of the patient's vital signs. Simulation and visualization can be a helpful aid when learning to recognize and properly respond to shock in a patient. Although estimations of hemorrhage may be inaccurate shock classifications as described by the world-wide accepted Advanced Trauma Life Support (Patton et al. 2001), guidelines can be helpful when considering the degree of hemorrhage a patient may have suffered (Geeraedts et al. 2009).

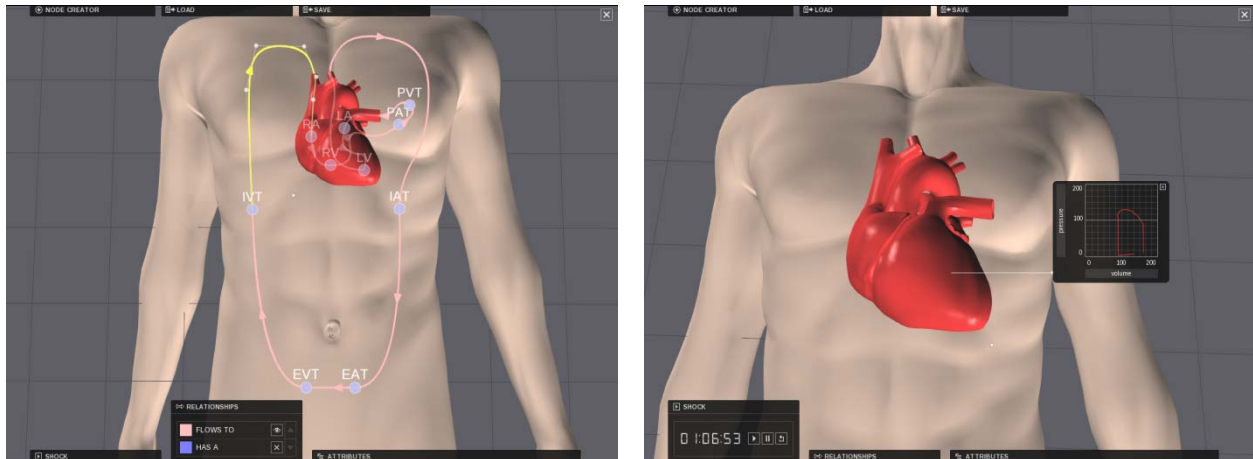


Figure 4a (left): A visualization of the Beneken (1965) cardiovascular model collocated with a 3D visualization of the human body and heart. The relationship connecting the intrathoracic venous tree concept to the right atrium is selected and shown in yellow. When a relationship is selected the control hull becomes visible, allowing users to modify the curvature of the relationship. Figure 4b (right): A plot of the pressure-volume loop is anchored to the left ventricle of the heart mesh. In this screen shot, the ontology is hidden so only the 3D meshes are visible.

A short-coming of our prototype is that currently there is no way to semantically supplement the ontology in order to define these guidelines because the guidelines are based on four “stages” that equate total blood loss to other symptoms. Currently, only linear relationships can be defined to link attributes within the ontology (this is described further in Section 4.3). It may be the case that these shock stages can be mapped to a linear function relating, for instance, heart rate and blood loss, but this doesn’t diminish the unique expressive power of the “stage” concept. As future work, we plan to incorporate semantics so that phenomena such as the stages of hypovolemic shock can be integrated into simulation models. Currently, the stages are hard-coded into the equation solver so when blood loss reaches certain levels, the variable *heart rate* of the concept *heart* is updated accordingly.

### 4.3 Ontology-Based Animation

After a simulation model is defined, it can be executed in our prototype. Further, we have shown the prototype supports integrative multimodeling by allowing the simulation and knowledge models to be visualized along with the more concrete geometric models (i.e., the textured 3D meshes). The ontology can also serve to connect simulation variables to visualization parameters resulting in dynamic 3D animations based on simulation behavior.

By adding *influence* (defined in Section 3.2) attributes to ontology concepts, our prototype will link the value of one attribute to another by mapping between linear ranges during simulation execution. To create the semantic link in the ontology, attributes *influence source*, *influence destination*, *influence source range*, and *influence destination range* are required. Ranges can be bounded by either vector or scalar values. The two influences in our hypovolemic shock case study are shown in Figure 5. The volume of the *heart* (calculated by summing the volume of all the *heart*’s subcomponents) is linked to the *scale*. The volume of the heart in the range (200ml, 600ml) is mapped to a scale in the range ([1.0,1.0,1.0], [1.3,1.3,1.3]). This results in a beating effect that can visually relay the pulse frequency and strength. As also depicted in Figure 5, total blood volume is mapped to the color of the skin. The blood volume influence ranges from 4240 ml (normal amount of blood for an adult male) to 2500 ml. The skin colors ranges from [red: 0.83, green: 0.82, blue: 0.81] (very pale) to [red: 0.86, green: 0.72, blue:



0.63] (a typical Caucasian skin tone). The result of these influence additions are shown in Figure 6, where three snap shots are taken while the simulation is executing.

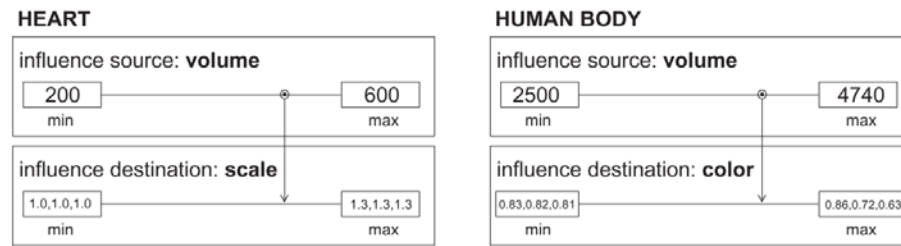


Figure 5: A diagram of the influences used to animate the 3D meshes of the heart and human body. The arrow denotes a linear mapping between ranges.

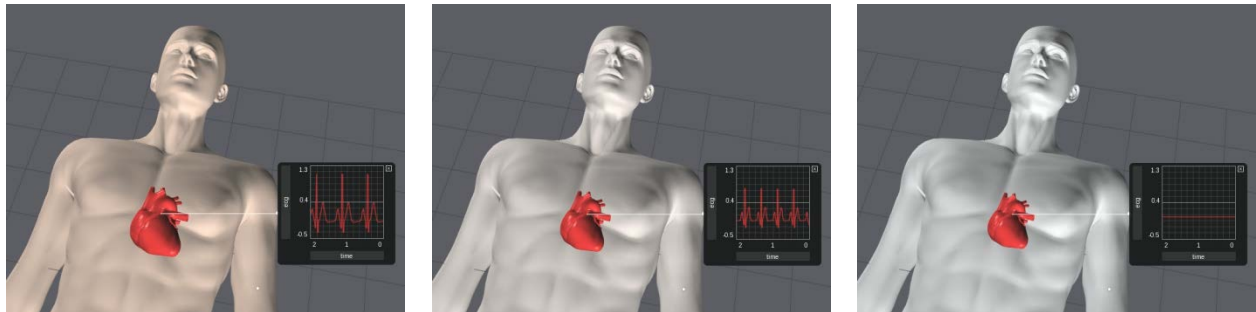


Figure 6: Snap shots from an executing simulation and visualization of hypovolemic shock. A plot of the electrocardiograph is anchored to the heart.

We plan to further explore the *influence* concept by relating the *influence source* to the output of equations that are listed as attributes. Consider, for instance, the electrocardiograph (ECG). The ECG is frequently used in clinical practice to monitor heart activity. Simulation models of the ECG do not employ the hydrodynamic metaphor and therefore cannot be created and executed in our current prototype. However, if ECG behavior can be represented by a single set of differential equations with the attribute *heart rate* as a coefficient, then there may be use in assigning those equations as attributes to ontology members. The equations would need to be solved at the same frequency that the general hydrodynamic solver solves its differential equations. We hard-coded the McSharry et al. (2003) ECG model to demonstrate this potential. The equations in the model are solved and total blood volume is used to scale the ECG output (much like it is used to color the skin of the human body). This linkage results in a weakened ECG signal as blood is drained, as depicted in the ECG plots in Figure 6 (note that ECG eventually flat-lines because the heart stops beating after 50% total blood loss occurs). Clearly, a medical researcher would desire a more sophisticated and accurate linkage of physiological parameters to ECG output, but this example demonstrates how using equations as attributes could enable executable ontology-based simulation models that cross mathematical paradigms.

## 5 CONCLUSIONS AND FUTURE WORK

We presented a methodology that employs ontology visualizations at the user interface level to support simulation model building and visualization construction activities. The visualization of an ontology serves as an anchor for other 3D visualization elements such as meshes and variable plots. Through the use of splines, relationship visualization can be sculpted to trace a meaningful 3D path within the visuali-

zation environment. To demonstrate the methodology, we used a software prototype to create an executable simulation and visualization of hypovolemic shock. Influence attributes are used to link simulation variables to visualization parameters. Future work on influences includes allowing equations as attributes to ontology concepts. Further, to allow more than linear mapping between attributes, we plan on adding the ability to create conditional functions as attributes (e.g., to model the stages of hypovolemic shock).

Thus far, our work has been in exploring the ontology affordances at the designer level. We are also interested in how manifesting ontological models (and embedded simulation models) within 3D visualizations can benefit education. This would require “viewer” software that allows users to view the integrative visualizations but not change any attributes or concepts in the ontology. Within the viewer software, we also plan to add a window that displays a traditional two-dimensional “textbook” view of the dynamic model. The 2D and 3D views will be co-interactive in that selecting model components in one view would cause the same model component to become highlighted in the alternate view. Such a visualization may help link concrete and abstract knowledge as previously suggested (Quarles et al. 2010). We plan to perform user studies with a group of physiology students presented with a physiological visualization constructed with our prototype. Study results could help inform future design decisions when building integrative visualizations for educational purposes.

## 6 ACKNOWLEDGEMENTS

The authors would like to acknowledge Jonathan Kibble, PhD and Judith Simms-Cendan, MD, College of Medicine, University of Central Florida; and Rebecca Lyons, Department of Psychology, Institute for Simulation and Training, University of Central Florida for their support during the development of the case study.

## REFERENCES

- Bartels R. H., J. C. Beatty, and A. Barsky. 1987. *An introduction to splines for use in computer graphics and geometric modeling*. Los Altos, CA: Kaufmann Publishers.
- Beneken J. E. W. 1965. *A mathematical approach to cardiovascular function: the uncontrolled human system*. P.h.D. Thesis, Medisch Fysisch Instituut TNO.
- Benjamin P., and M. Graul. 2006. “A framework for adaptive modeling and ontology-driven simulation”. In *Proceedings of the SPIE, Enabling Technologies for Simulation Science X*. Edited by D. A. Trevisani. Bellingham, WA: SPIE.
- Benjamin P., M. Patki, and R. Mayer. 2006. “Using ontologies for simulation modeling”. In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, B. Lawson, J. Liu, and F. P. Wieland, 1151-1159. Piscataway, NJ : Institute of Electrical and Electronics Engineers, Inc.
- Berners-Lee T., J. Hendler, and O. Lassila. 2001. “The Semantic Web.” *Scientific American*, 284(5): 34-43.
- Bhatt M., W. Rahayu, and G. Sterling. 2004. “sedOnto: A Web enabled ontology for synthetic environment representation based on the SEDRIS specification”. In *Fall Simulation Interoperability Workshop*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Bhatt M., W. Rahayu, and G. Sterling. 2005. “Synthetic environment representational semantics using the web ontology language”. In *Proceedings of the Sixth International Conference on Intelligent Data Engineering and Automated Learning*. Edited by M. Gallagher, J. Hogan, F. Maire, 9-16. Berlin: Springer.
- Blomqvista E., and A. Öhgren. 2008. “Constructing an enterprise ontology for an automotive supplier”. *Engineering Applications of Artificial Intelligence*. 21(3): 386-397.
- Blender 2011. “blender.org - Home”. Accessed May 1, 2011. <http://www.blender3d.org>
- Ezzell Z., P. A. Fishwick, B. Lok, A. Pitkin, and S. Lampotang. 2011. “An Ontology-enabled interface to simulation model building and visualization”. Upcoming special issue of *Journal of Simulation*.

- Fischler I., C. E. Kaschub, D. E. Lizdas, S. Lampotang. 2008. "Understanding of Anesthesia Machine Function is Enhanced with a Transparent Reality Simulation". *Simulation in Healthcare*, 3:26-32.
- Fishwick P. A. 2004. "Toward an integrative multimodeling interface: a human-computer interface approach to interrelating model structures". *Simulation*, 80: 421 – 432.
- The Gene Ontology Consortium. 2010. "The Gene Ontology in 2010: extensions and refinements". *Nucleic Acids Research* 38(1): 331-335.
- Geeraedts L. M. Jr., H. A. Kaasjager, A. B. van Vugt, J. P. Frölke. 2009. "Exsanguination in trauma: A review of diagnostics and treatment options." *Injury*, 0(1): 11-20.
- Gruber, T. R. 1993. "A Translation approach to portable ontology specifications." *Knowledge Acquisition*, 5: 199-220.
- Hopkins J. F., and P. A. Fishwick. 2003. "The rube framework for software modeling and customized 3-D visualization." *Visual Language and Computing*, 14(1): 97-117.
- Ishii H. 2008. "The tangible user interface and its evolution". *Communications of the ACM*, 51(6): 32-36.
- Jacquez J. A. 1996. *Compartmental analysis in biology and medicine*. 3rd ed. Ann Arbor, MI: BioMedware.
- Knublauch H., R. W. Ferguson, N. F. Noy, M. A. and Musen. 2004. "The Protégé OWL plugin: An open development environment for semantic web applications". In *Proceedings of the 2004 International Semantic Web Conference*. Edited by S. A. McIlraith, D. Plexousakis, and F. Harmelen F, 229-243. Berlin: Springer.
- Lacy L., and W. Gerber. 2004. "Potential modeling and simulation applications of the web ontology language – OWL". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, B. A. Peters, 265-270. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- Lawrence P. F., R. M. Bell, and M. T. Dayton. 2006. *Essentials of General Surgery*. Philadelphia, PA: Lippincott Williams & Wilkins.
- Liang V., and C. J. Paredis. 2003. "A port ontology for automated model composition". In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 613-622. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.
- McSharry P. E., G. D. Clifford, L. Tarassenko, and L. A. Smith. 2003. "A Dynamical Model for Generating Synthetic Electrocardiogram Signals." *Transactions on Biomedical Engineering*, 50(3): 289 – 294.
- Miller J. A., G. T. Baramidze, A. P. Sheth, and P. A. Fishwick. 2004. "Investigating ontologies for simulation modeling". In *Proceedings of the 2004 Symposium on Simulation*. 57-71. Washington, DC: IEEE Computer Society.
- Miller J. A., A. F. Seila, and X. Xiang. 1999. "The JSIM web-based simulation environment". *Future Generation Computer System*. 17: 119-133.
- OGRE 2011. "OGRE - Open Source 3D Graphics Engine". Accessed May 1, 2011. <http://www.ogre3d.org/>
- Park M., and P. A. Fishwick. 2005. "Integrating dynamic and geometric model components through ontology-based inference". *Simulation* 81(12): 195-813.
- Patton K., D. L. Funk, M. McErlean, J. M. Bartfield. 2001. "Accuracy of estimation of external blood loss by EMS personnel". *Trauma* 50(5): 914-916.
- Pop A, P. Fritzson, O. Johansson 2004. An integrated framework for model-driven design and development using Modelica. In *Proceedings of the 45th Conference on Simulation and Modeling of the Scandinavian Simulation Society*, edited by B. Elmegaard, J. Sparring, K. Erleben, K. Sørensen, 23-24. Accessible online at <http://www.scansims.org/sims2004/>.
- Quarles J., P. A. Fishwick, S. Lampotang, I. Fischler, and B. Lok. 2010. "A mixed reality approach for interactively blending dynamic models with corresponding physical phenomena". *ACM Transactions on Modeling and Computer Simulation*, 20(4): 22.

- Rosse C., and J. L. V Mejino. 2007. "The foundational model of anatomy ontology". In *Anatomy Ontologies for Bioinformatics: Principles and Practice*. Edited by D. Davidson, and R. Baldock, vol. 6, 59-117. London: Springer.
- Silver G. A., O. A. Hassan, and J. A. Miller. 2007. "From domain ontologies to modeling ontologies to executable simulation models". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1108-1117. Piscataway, NJ : Institute of Electrical and Electronics Engineers, Inc.
- Simulink 2011. "Simulink - Simulation and Model-Based Design". Accessed May 1, 2011. <http://www.mathworks.com/products/simulink/>
- Tolk, A. and C. D. Turnitsa. 2007. "Conceptual Modeling of information exchange requirements based on ontological means". In *Proceedings of the 2007 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1100-1107. Piscataway, NJ : Institute of Electrical and Electronics Engineers, Inc.

**ZACH EZZELL** is a PhD candidate in the department of Computer and Information Science and Engineering at the University of Florida. Ezzell's research interest include simulation and visualization frameworks, and human computer interaction.

**PAUL A. FISHWICK** (Ph.D., University of Pennsylvania) is Professor of Computer and Information Science and Engineering at the University of Florida. Fishwick's research interests are in modeling methodology, aesthetic computing, and the use of virtual world technology for modeling and simulation. He is a Fellow of the Society of Modeling and Simulation International, and recently edited the CRC Handbook on Dynamic System Modeling (2007). He served as General Chair of the 2000 Winter Simulation Conference in Orlando, Florida

**JUAN CENDAN** (M.D., University of Florida) is Assistant Dean for Simulation and Associate Professor of Surgery at the University of Central Florida. Cendan is responsible for identifying, implementing and coordinating the use of simulation technology for conducting and improving medical education. He is a board-certified surgeon and active with the American College of Surgeons. His work has appeared in both medical and technology publications including *The American Journal of Surgery*, *Medical Hypotheses*, *Studies in Health Technology and Informatics*, *Proc. of the IEEE Symposium on 3D User Interfaces*, and the *Journal of Bio-algorithms and Med-systems*.