INTEGRATION OF MULTI-SCALE BIOSIMULATION MODELS VIA LIGHT-WEIGHT SEMANTICS

JOHN H. GENNARI[1], MAXWELL L. NEAL[1], BRIAN E. CARLSON[2], & DANIEL L. COOK[3]

[1]*Biomedical & Health Informatics,* [2]*Bioengineering,* [3]*Physiology & Biophysics, University of Washington, Seattle, WA, 98195, USA*

Currently, biosimulation researchers use a variety of computational environments and languages to model biological processes. Ideally, researchers should be able to semi-automatically merge models to more effectively build larger, multi-scale models. However, current modeling methods do not capture the underlying semantics of these models sufficiently to support this type of model construction. In this paper, we both propose a general approach to solve this problem, and we provide a specific example that demonstrates the benefits of our methodology. In particular, we describe three biosimulation models: (1) a cardio-vascular fluid dynamics model, (2) a model of heart rate regulation via baroreceptor control, and (3) a sub-cellular-level model of the arteriolar smooth muscle. Within a light-weight ontological framework, we leverage *reference ontologies* to match concepts across models. The light-weight ontology then helps us combine our three models into a merged model that can answer questions beyond the scope of any single model.

1. Semantics for biosimulation modeling

Biomedical simulation modeling is an essential tool for understanding and exploring the mechanics and dynamics of complex biological processes. To this end, researchers have developed a wide variety of simulation models that are written in a variety of languages (SBML, CellML, etc.) and are designed for a variety of computational environments (JSim, MatLab, Gepasi, Jarnac, etc.). Unfortunately, these models are not currently interoperable, nor are they annotated in a sufficiently consistent manner to support intelligent searching or integration of available models.

In the extreme case, a biosimulation model contains no explicit information about what it represents— it is only a system of mathematical equations encoded in a computational language. The biological system that is the subject of the model is implicit in the code; the code is an abstraction of that system into mathematical variables and equations that must be interpreted by a researcher. If one researcher wishes to understand or use a model created by another, he or she must (usually) communicate directly with those that created the model. For complex, multi-scale models, this problem is a bottleneck to further progress—if models could be archived, re-used, and connected together computationally, we would avoid a great deal of work spent "re-creating the wheel", by leveraging more directly the work of others.

Recognizing this problem, there are on-going efforts to build repositories of annotated biosimulation models [1-4]. However, these annotations are predominantly human-interpretable and depend on local semantics. For example, repositories of JSim models [4] and CellML models [1] rely on in-line code annotations to explain mathematical equations—annotations that are not machine-interpretable. The BioModels repository [3] of SBML-encoded models uses XML-based annotations, but, we argue, these still lack the strong semantics required for computer-aided integration. (This library is also restricted to the scales of cellular and biomolecular problems). Given that the goal of multi-scale modeling is the flexible reuse and integration of models to solve large-scale modeling problems, we argue that a much stronger, machine-interpretable semantic framework needs to be applied to these biosimulation models.

In this paper, we propose a flexible solution that will allow biosimulation models to be re-used and re-combined in a plug-n-play manner. The thrust of our approach is to build light-weight ontological models of biological systems for annotating model variables in terms of the physical properties and the anatomical entities to which they refer, and for explicitly representing how these property variables depend upon each other. More concretely, we demonstrate how our ontologies can represent the semantics of three models, and then use this information to help merge these into a larger, multi-scale biosimulation model.

We begin by describing the three source models that make up a driving use-case for our research, and then show how each model is semantically mapped to our light-weight ApplModel Ontology framework (section 2). We can then analyze and visualize the semantics of the models using available software tools (Prompt [5], see section 3). Such tools help us merge the models, and we show that our merged model can answer multi-scale questions that are not answerable by single component models (section 4).

1.1 Motivating use-case: Arteriolar calcium uptake & heart rate

Our driving biological problem is to create a multi-scale cardiovascular model from three independently-coded models that contain overlapping parts of the cardiovascular regulatory system. Figure 1 provides both a view of our three 'source' models (top half) and our 'target'—a merged, multi-scale model (bottom half). Our use-case goal is to employ the merged model to answer a multi-scale, systems-level question such as "How do heart rate and blood pressure depend on calcium uptake into arteriolar smooth muscle cells?"—a question that cannot be answered by the individual source models.

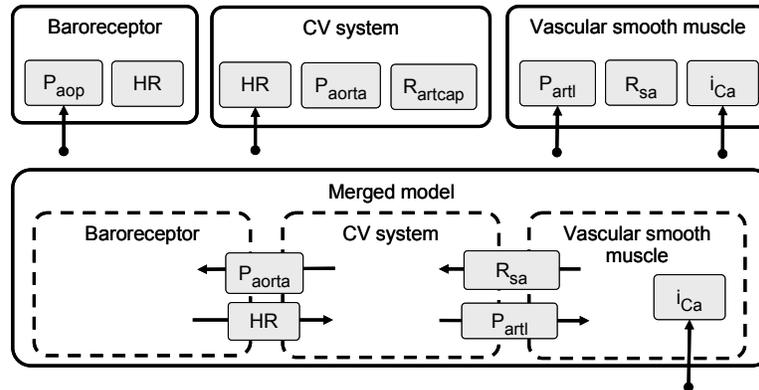The three source models at the top of figure 1 are each a lumped-parameter

Figure 1. A simple overview of our use-case and computational goals. We are building an infrastructure for querying, interpreting and merging biosimulation model such as the three models at the top of the figure, into larger, multi-scale models, such as shown on the bottom.

model independently encoded in the JSim simulation environment[6].[a] A cardiovascular model (CV) was coded by the second author and is a condensed version of a previously published model [7]. Using a constant heart rate input (HR) and other parameters, the CV model computes time-varying blood pressures and flows in a 4-chambered heart and in the pulmonary and systemic vessels. Our baroreceptor model (BARO) was originally coded by Daniel Beard and is based on Lu and Clark [8] and Spickler et al. [9]. The BARO model takes aortic blood pressure as input and computes a time-varying heart rate as a feedback signal to control blood pressure. A vascular smooth muscle model (VSM) was coded by the third and fourth authors to model the effect of Ca++ ion uptake into arteriolar smooth muscles cells and its consequent effect on arteriolar flow resistance. In section 4, we provide details about how we created the merged model, as well as descriptions of the parameters and variables listed in figure 1.

   As one measure of the challenges inherent in merging these models, our combined source models include over 190 named variables and parameters whose biophysical meanings are buried in code annotations (where available) that are specific to each model. To merge these models appropriately, we need to consider three sorts of challenges. First, we must discover identical biophysical entities. For example, heart rate is only coincidentally encoded as HR in both the CV and BARO models and in fact, represents the same biophysical entity. Sec-

---

[a] Full source code for these three models are available at
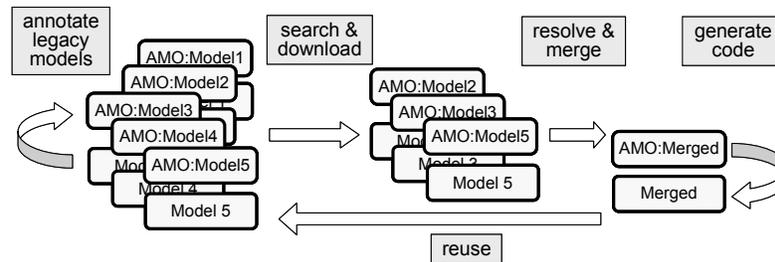   http://trac.biostr.washington.edu/trac/wiki/JSimModels

Figure 2. An approach to making biosimulation models "plug-n-play": annotate, search, resolve, merge, encode, and ultimately reuse.

ond, we must discover and resolve variables that are related, but not identical. For example, Rsa represents the arteriolar fluid resistance in VSM but the arterioles are only part of the systemic arterial vasculature whose fluid resistance is represented as Rartcap (arteries, arterioles and capillaries) in CV. Third, we must discover and resolve variable dependencies. HR in the CV model is an input or controlled variable whereas in BARO it is an output or computed variable that depends ultimately on aortic blood pressure (Paop). Thus, the HR variables from CV and BARO should be merged into a single variable, so that the heart rate calculated by BARO becomes an input to the CV model.

1.2 A solution: Light-weight ontological annotation

The above challenges all revolve around defining the biophysical semantics of the variables and parameters within models. As we describe in the next section, our solution begins by annotating biosimulation models with light-weight semantics, as provided by our Application Model Ontology (AMO, see also section 2.2). The AMO is small, and we envision tool support to make annotation as easy as possible for simulation modelers. More broadly, figure 2 shows how this annotation step is part of a more general architecture for reusable biosimulation models. Once models are annotated with AMO, model libraries can be more intelligently searched for relevant models. As we show in section 3, once selected, AMO annotations can help with the tasks of resolving differences between models to create merged models. Next, from the merged models, we plan to generate code in a variety of simulation languages using code-generation methods with which we have experience [10]. Ultimately, as with software reuse, merged models can be returned to the library for reuse by others.

2. Semantic annotation via ontologies

Computer-interpretable semantics are best captured by formal ontologies. In recent years, a wide variety of ontologies for biology have become available.

Prominent among these are the ontologies available at the Open Biological Ontologies (OBO), and its OBO foundry project (at www.obofoundry.org). These ontologies cover a variety of levels of formality and abstraction, as well as a variety of domain topics. However, although ontologies of physical entities such as genes, species, and anatomy have been well-developed, the domain of biosimulation also requires *properties* of anatomical entities (such as volume or fluid pressure) as well as some understanding of the *processes* by which these properties change over time.

In general, we posit that although formal, abstract, "heavy" ontologies are essential for unambiguous, machine interpretable annotation, end-users need a light-weight methodology for semantic annotation. Thus, we advocate using two sorts of ontologies: (1) *reference* ontologies, that allow us to ground our work in the formal semantics of structural biology and physics, and (2) *application model* ontologies that are tailored for the specific semantics of particular biosimulation models.

2.1   Reference ontologies: FMA and OPB

For our example, we use two reference ontologies: the Foundational Model of Anatomy (FMA, [11]), a mature reference ontology of human anatomy, and the Ontology of Physics in Biology (OPB), an ontology of classical physics designed for the physics of biological systems. The FMA is a nearly complete structural description of a canonical human body. Its taxonomy of *Anatomical entities* is organized according to kind (e.g., *Organ* system, *Organ*, *Cell*, *Cell part*) with parthood relations so that, for example, the *Cardiovascular System* has parts such as *Heart*, *Aorta*, *Artery,* and *Arteriole*. Parts are also related by other structural relations so that, for example, the *Aorta* is *connected_to* the *Heart* and the *Blood in aorta* is *contained_in* the *Aorta*.

The Ontology of Physics for Biology (OPB) is a scale-free, multi-domain ontology of classical physics based on systems dynamics theory [12-15]. It thus distinguishes among four *Physical property* superclasses for lumped-parameter systems: *Force, Flow, Displacement,* and *Momentum*. As shown in figure 3A, each of these *Physical property* classes has subclasses in seven "energy domains": *Fluid mechanics, Solid mechanics, Electricity, Chemical kinetics, Particle diffusion, Heat transfer,* and *Magnetism*. The OPB also encodes *Physical dependency* relations that include *Theorems of physics* (e.g., *Conservation of energy*) and *Constitutive property dependencies* (shown in figure 3B) such as the *Fluid capacitive dependency* relation that governs, say, how ventricular volume depends on ventricular blood pressure.

By combining the knowledge in the FMA and the OPB one can unambigu-

A) *Physical property*　　　　B) *Physical dependency*

Discrete rate property　　Discrete state property
● Discrete flow　　　　　　● Discrete displacement
⇨ ● Fluid flow　　　　　⇨ ● Fluid volume
　● Solid velocity　　　　　● Solid displacement
　● Electrical current　　　　● Electrical charge
　● Chemical flow　　　　　● Chemical concentration
　● Particle flow　　　　　● Particle number
　● Entropy flow　　　　　● Thermal entropy
● Discrete force　　　　● Discrete momentum
⇨ ● Fluid pressure　　　⇨ ● Pressure momentum
　● Solid force　　　　　　● Solid force momentum
　● Voltage　　　　　　　● Induced flux
　● Chemical potential
　● Particle concentration
　● Temperature

● Resistive dependency
⇨ ● Fluid resistive dependency
　● Mechanical resistive dependency
　● Electrical resistive dependency
　● Chemical reaction dependency
　● Diffusional resistive dependency
　● Heat transfer resistive dependency
● Energy storage dependency
　● Capacitive dependency
⇨ ● Fluid capacitive dependency
　　● Solid capacitive dependency
　　● Electrical capacitive dependency
　　● Chemical accumlation dependency
　　● Particle accumulation dependency
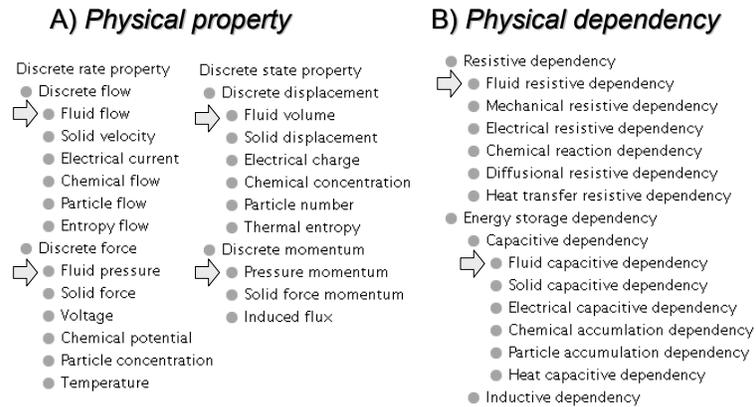　　● Heat capacitive dependency
● Inductive dependency

Figure 3. Main classes of the Ontology of Physics for Biology (OPB). The classes highlighted with arrows indicate the fluid mechanics aspects for both physical properties and dependencies.

ously annotate model variables by associating an FMA:Anatomical entity with an OPB:Physical property, creating duples such as [FMA:Blood of aorta :: OPB:Fluid pressure]. Thus, for modeling multi-scale biological systems, the FMA and OPB offer a wealth of machine-accessible anatomical and biophysical knowledge that can be leveraged for annotating biosimulation code.

2.2   *The application model ontology*

Our goal in developing the Application Model Ontology (AMO) is to provide an ontological framework for creating reusable, lightweight ontological annotations of biosimulation models—ApplModels (for application models). The fundamental idea of the AMO is to allow researchers to build models that use only very small subsets of very large and complex reference ontologies. A biosimulation researcher does not care or want to know about all of the anatomic entities in the FMA nor about theorems across all seven of the energy domains in the OPB. Thus, ApplModels exploit, but do not depend on, external reference ontologies, and yet can be "lightweight" and customized to represent idiosyncratic biophysical entities and relations. AMO classes are formally defined according to the principles espoused by the OBO foundry, and are created and edited within the Protégé environment [16].

　Figure 4a shows a screenshot of a portion of the AMO base classes in Protégé and some examples of how these classes are filled in by the BARO biosimulation model we described earlier. The higher-level classes such as *physical entity* or *physical property* are basic AMO classes, while the leaf nodes
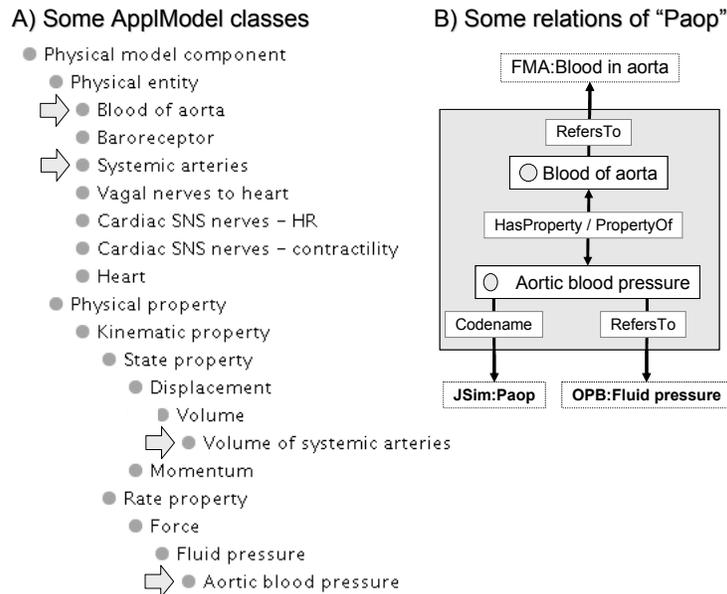
A) Some ApplModel classes

- Physical model component
  - Physical entity
    - ⇨ Blood of aorta
    - Baroreceptor
    - ⇨ Systemic arteries
    - Vagal nerves to heart
    - Cardiac SNS nerves – HR
    - Cardiac SNS nerves – contractility
    - Heart
  - Physical property
    - Kinematic property
      - State property
        - Displacement
          - Volume
            - ⇨ Volume of systemic arteries
      - Momentum
    - Rate property
      - Force
        - Fluid pressure
          - ⇨ Aortic blood pressure

B) Some relations of "Paop"

FMA:Blood in aorta

RefersTo

◯ Blood of aorta

HasProperty / PropertyOf

◯ Aortic blood pressure

Codename   RefersTo

JSim:Paop   **OPB:Fluid pressure**

Figure 4. The Application Model Ontology, as filled in for the BARO biosimulation model.

show how AMO was filled in for the BARO model. Figure 4b shows some detail of the annotation for the BARO variable "Paop", including links to reference ontologies. To capture the semantics for this variable, we first created ApplModel classes that refer to the corresponding reference ontology concepts (FMA: *Blood in aorta* and OPB:*Fluid pressure*) and then the specific class that represents one particular fluid pressure, namely "Paop". Figure 4a shows the entire set of "physical entities" for the BARO model; not shown are the constitutive relationships and dependencies that are represented by equations in the model code.

Wherever possible, users should refer directly to reference ontology classes—such ontologies make model integration possible, by enforcing a common semantics to particular terms. However, users can also create special-purpose (or idiosyncratic) subclasses for particular biosimulation models. For example, the CV model variable "Rartcap" is the resistance in a single entity that lumps systemic arteries and capillaries together; such an entity does not exist in the FMA reference ontology. However, with the ApplModel annotations, we can easily create a special subclass of *Physical thing* such as *Systemic-Arteries-Capillaries* that uses *AMO:HasPart* relations to the *Systemic arteries* and *Systemic capillaries* classes that are available in the FMA.

It is the ability to integrate idiosyncratic semantics with reference ontologies and other external knowledge resources that makes light-weight ApplModels a powerful approach to biosimulation model annotation. Once annotated with such semantics, we can both better understand these models, and use existing ontology analysis tools to help with the model integration task.

3. Comparing and merging models

When merging biosimulation models there are usually significant semantic differences that must be resolved. While some of these differences may be obvious and easy to find, automated analysis tools can greatly help researchers find and resolve such differences. One major advantage of annotating biosimulation models with ontologies, is that there are pre-existing tools to help with these sorts of tasks. In our case, we have employed Protégé's Prompt plug-in tool for ontology comparison and merging tasks [5].

Prompt is designed for interactive, semi-automatic model merging. Given two ontologies, Prompt analyzes the classes and relationships in the two models, and then suggests a set of mappings that connect concepts between the two models. The user can inspect these candidate matches, and confirm all or some of these matches. Prompt then uses this information to suggest additional matches, and this interactive cycle repeats.

For our use-case, when we gave Prompt the ApplModel ontologies for the BARO and CV models, it was able to recognize that, for example, "systemic arteries" is a shared concept—regardless of how it was coded in the source models—because it was annotated with the common FMA reference ontology term. Furthermore, the Prompt visualization tools reveal that there are significantly different relationships around "systemic-arteries" across the two models. Figure 5 shows the "neighborhood view" of nearby semantic relationships as presented by Prompt when it proposes the match for "systemic-arteries". Figure 5a shows that the BARO model links resistance as a direct property of systemic arteries, whereas the CV model (figure 5b) uses the set "systemic arteries and capillaries", which has a resistance. As figure 1 shows, there is a similar discrepancy about resistance between the CV model (Rartcap) and the VSM model, which only considers the systemic arterioles (Rsa). To appropriately merge models, researchers must resolve these sorts of semantic discrepancies.

Even when the underlying anatomy is consistent, models may use the properties of those entities in different manners. In our case, heart rate (HR in figure 1) is defined consistently across the BARO and CV models, but in the BARO model it is an output, whereas the CV model uses HR as one of its inputs. The difference can be readily visualized in Prompt, because the ApplModel ontology
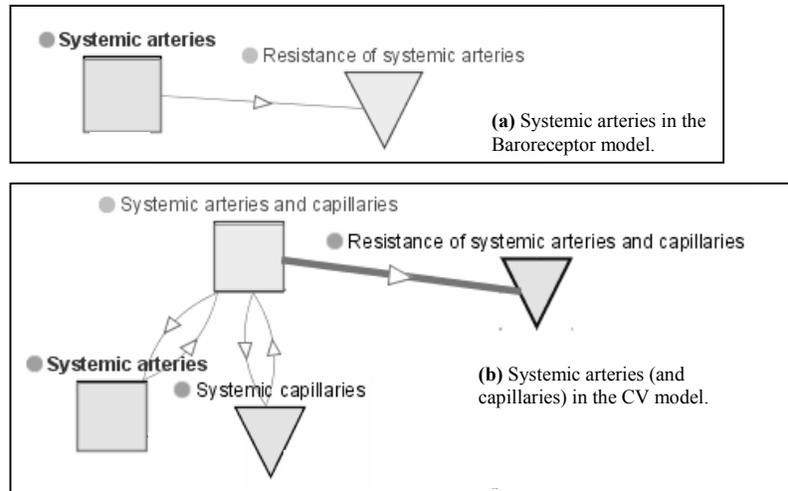
Figure 5. The two uses of the concept "Systemic arteries" in the BARO and CV models. These views were produced by Prompt when suggesting mappings between the models. (Arcs show relations, such as "part-of" between entities. Squares are "fully expanded" entities, whereas triangles are entities that can be further expanded.)

includes the relationships "dependsDirectlyOn" and "affectsDirectly". (Such relationships are shown as colored arcs in the Prompt visualization.)

As we show in figure 2, our overall expectation is that Prompt can be used as a step in the overall process of building multi-scale biosimulation models. However, even if researchers do not immediately expect to combine models directly, a Prompt comparison of closely related models can used to reveal model semantics and physiological relationships that are otherwise implicit in the mathematical code. By visualizing graphically the set of relationships among anatomic entities and their physical properties, biosimulation researchers can better understand how two models are and are not the same. In addition, Prompt may actually help de-bug biosimulation models by making it visually apparent when relationships are missing, problematic, or incorrect.

4. Status and Results

As a summary of our results so far, we have (a) annotated three related source models with the Application Model Ontology, (b) used the Prompt tool to ana-lyze these models, helping us visualize and understand the differences across models, and (c) hand-coded a merged model into JSim that implements the deci-sions made during the merge step. Thus, our product is an integrated, executable model that can indeed answer our original driving question: "How do heart rate and blood pressure depend on calcium uptake into arteriolar smooth muscle
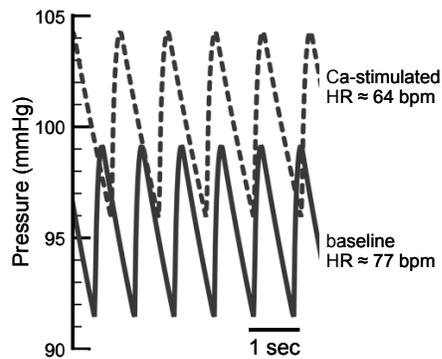
Figure 6. The result of increased Ca++ uptake, as an output from our merged JSim model.

cells?" Figure 6 shows an annotated output from the JSim execution of our model, showing the expected increase in blood pressure and decrease in heart rate when Ca uptake is increased.

We began integration of the JSim multi-scale model by first merging the CV and BARO models. To do this, we merged two shared concepts: heart rate and aortic blood pressure (see figure 1: HR, Paop, and Paorta).

- We changed the BARO term Paop from an independent input to a time-dependent variable output and set it equal to Paorta, the aortic blood pressure variable from the CV model.
- We removed the independent HR input from the CV model so that cardiac activation would depend on the BARO model's variable HR.
- We added a new discrete HR variable (HRdiscrete) that only updates at the end of the cardiac cycle to prevent intra-beat fluctuations in heart rate. (To do this, we needed to add some procedural code to the merged model.)

Next, we merged the result with the VSM model by combining representations of resistance:

- Given the high proportion of vascular resistance in the arterioles, we assumed that the time-varying arteriolar resistance (Rsa) computed in the VSM model to be the same as the resistance of arteries and capillaries from the CV model (Rartcap, a constant). Therefore, we changed Rartcap to a time-dependent variable equal to Rsa.
- To couple arteriolar resistance with the dynamics of the CV model, we changed the arteriolar blood pressure input (Partl) to a time-dependent variable equal to the average pressure between the CV model's arterial/capillary and venous compartments.

The resulting multi-scale model includes 65 algebraic and 25 ordinary differential equations. Although more detailed models of the cardiovascular system and smooth muscle cell dynamics exist, our system produces physiologically normal steady state averages for circulatory and smooth muscle cell dynamics and allows investigations into the influence of subcellular activity on tissue-level dynamics (as in figure 6).

5. Discussion and future work

There remains much work to do before our broad ideas of model integration (as shown in figure 2) can be implemented and fully tested. However, given our work to date, three aspects of our vision seem within reach: (1) improved use of Prompt for model merging, (2) improved use of inference over knowledge from reference ontologies, and (3) automatic generation of simulation code.

To date, we have only used Prompt as a visualization tool, allowing us to see discrepancies and understand linkages between models. However, as we have described, Prompt is designed to actually carry out model merging in an interactive manner. Furthermore, Prompt is designed with a plug-in architecture, which means it can be easily custom-tailored to meet our needs. Therefore, we will be able to use Prompt to carry out most of the model merging, although some parts of the work will remain manual (e.g., the addition of procedural code described earlier around the "HRdiscrete" variable).

As reference ontologies, the FMA and OPB both contain a wealth of knowledge that could be used to more intelligently guide model merging. For example, Prompt cannot currently notice that the diameter of an arteriole, a variable in the VSM model is related to arterial blood volume. However, the FMA knows that the arterioles are part of the systemic arterial tree, and the OBP knows that the diameter (along with the length) can determine the volume of an arteriole. We should therefore be able to use this sort of reference ontology knowledge to improve Prompt so that it can suggest mappings between variables such as arteriolar diameter and arterial/capillary blood volume.

We designed our ontologies and semantic markup methods to be independent of any particular biosimulation modeling language. We have so far worked exclusively with JSim models, but we believe that our ideas apply equally well to SBML and other simulation languages. We do not yet have a system for automatic code generation from our ApplModels, but we do have prior experience generating JSim code[10], and thus, we aim to build a code-generator for at least two targets: JSim and SBML. Such a tool would allow us to explore code-level semantic differences that might affect merging SBML models with JSim models. We hope that the semantic annotations provided by our ApplModel ontologies will help clarify these differences, but this intuition must be verified.

Our results represent a novel application of ontology-based semantics to help understand the deep biophysical meanings of terms used in biosimulation models. We have then used these semantics to facilitate merging models into larger, multi-scale biosimulations across very different physiological domains.

Acknowledgments

References

1. CellML. *Model Repository -- CellML*. http://www.cellml.org/models/

2. MATLAB. *MATLAB Central File Exchange*. http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do

3. Le Novere N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, et al. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res* 2006;34(Database issue):D689-91.

4. NSR-Physiome. *NSR Physiome Model Wiki*. http://www.physiome.org/model/

5. Noy NF, Musen, MA. The PROMPT suite: Interactive tools for ontology merging & mapping. *International Journal of Human-Computer Studies* 2003;59(6):983-1024.

6. JSim. *The JSim Home Page at NSR*. http://physiome.org/jsim/index.html

7. Kerckhoffs RC, Neal ML, Gu Q, Bassingthwaighte JB, Omens JH, McCulloch AD. Coupling of a 3D finite element model of cardiac ventricular mechanics to lumped systems models of the systemic and pulmonic circulation. *Ann Biomed Eng* 2007;35(1):1-18.

8. Lu K, Clark JW, Jr., Ghorbel FH, Ware DL, Bidani A. A human cardiopulmonary system model applied to the analysis of the Valsalva maneuver. *Am J Physiol Heart Circ Physiol* 2001;281(6):H2661-79.

9. Spickler JW, Kezdi P, Geller E. Transfer characteristics of the carotid sinus pressure control system. In: Kezdi P, editor. Baroreceptors and Hypertension. Dayton, OH: Pergamon. p. 31-40.

10. Cook DL, Gennari JH, Wiley JC. Chalkboard: Ontology-Based Pathway Modeling And Qualitative Inference Of Disease Mechanisms. *Pac Symp Biocomput* 2007;12:16-27.

11. Rosse C, Mejino JL, Jr. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J Biomed Inform* 2003;36(6):478-500.

12. Riggs DS. *Control theory and physiological feedback mechanisms*; 1970.

13. Borst P, Akkermans J, Pos A, Top J. The PhysSys ontology for physical systems. In: Bredeweg B, editor. Working Papers of the Ninth International Workshop on Qualitative Reasoning QR'95; 1995: University of Amsterdam; p. 11-21.

14. Karnopp D, Margolis DL, Rosenberg RC. *System dynamics: a unified approach*. 2nd ed. New York: Wiley; 1990.

15. Mikulecky DC. Network thermodynamics: a candidate for a common language for theoretical and experimental biology. *Am J Physiol* 1983;245(1):R1-9.

16. Gennari JH, Musen MA, Ferguson RW, Grosso WE, Crubezy M, Eriksson H, et al. The evolution of Protege: an environment for knowledge-based systems development. *Int. J. Human–Computer Studies* 2003;58:89-123.