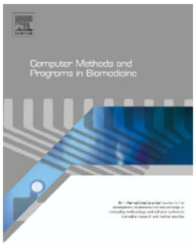
computer methods and programs in biomedicine xxx (2012) xxx–xxx

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



Design of a framework for modeling, integration and simulation of physiological models[☆]

E. Zeynep Erson*, M. Cenk Çavuşoğlu

Case Western Reserve University, Electrical Engineering and Computer Science Department, United States

ARTICLE INFO

Article history:
Received 22 September 2010
Received in revised form
11 October 2011
Accepted 30 November 2011

Keywords:
Information flow
Integrative physiology
Multiscale physiological model
Software framework

ABSTRACT

Multiscale modeling and integration of physiological models carry challenges due to the complex nature of physiological processes. High coupling within and among scales present a significant challenge in constructing and integrating multiscale physiological models. In order to deal with such challenges in a systematic way, there is a significant need for an information technology framework together with related analytical and computational tools that will facilitate integration of models and simulations of complex biological systems. Physiological Model Simulation, Integration and Modeling Framework (Phy-SIM) is an information technology framework providing the tools to facilitate development, integration and simulation of integrated models of human physiology. Phy-SIM brings software level solutions to the challenges raised by the complex nature of physiological systems. The aim of Phy-SIM, and this paper is to lay some foundation with the new approaches such as *information flow* and *modular* representation of the physiological models. The ultimate goal is to enhance the development of both the models and the integration approaches of multiscale physiological processes and thus this paper focuses on the design approaches that would achieve such a goal.

1. Introduction

Emergence of systems biology provided a comprehensive and integrative perspective to examine the structure and function at the cellular and organism levels of complex biological systems instead of focusing on the isolated parts [1]. However due to the complex nature of the physiological systems, development, integration of multiscale models and linking the layers stand as one of the challenges for the model developers [2]. In order to increase the effectiveness of multiscale integration of physiological processes, it is obvious that information technology approaches are required.

Physiological Model Simulation, Integration and Modeling Framework, Phy-SIM, is an information technology framework with related analytical and computational tools that facilitates development, integration and simulation of physiological models. Besides providing tools to develop physiological models, the strongest feature of the framework is providing the environment to aid the development of integration approaches. Since the problem of multiscale integration of physiological models, is itself an open research area, frameworks such as Phy-SIM providing tools to enhance the process is very critical.

A module is a structurally and functionally meaningful part of a system that can be separated from other

---

[☆] This work was supported in part by National Science Foudation under grants CISE CNS-0423253, IISE-0805495, IIS-0905344, and CNS-1035602, and US DoC under grant TOP-39-60-04003.

* *Corresponding author at*: Case Western Reserve University, Electrical Engineering and Computer Science Department, 10900 Euclid Av., Cleveland, OH 44106, United States. Tel.: +1 216 368 4575.

E-mail addresses: eze@case.edu, zerson@gmail.com (E.Z. Erson), cavusoglu@case.edu (M.C. Çavuşoğlu).

0169-2607/$ – see front matter

doi:10.1016/j.cmpb.2011.11.010

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

components. Modularity is advantageous for model storing, sharing and reproducibility [3]. Although modularity is a desirable feature for the definition of physiological models, it is not usually properly implemented [4].

Modularity in living organisms is studied from an evolutionary perspective and it is stated that, modular architectures with functional separation are more robust and amenable to design and adaptation. In evolution, modularity brings an advantage to modifications in modules without changing intrinsic behaviors and so providing high reusability [5]. Evolution in nature is actually not so different than evolution in software environments. Reusability, easy extension/modification are desirable attributes in software life-cycles as well. Therefore in addition to the structural modularity in anatomy, importance of the protocols and hierarchy in functionality for biological systems should also be considered when developing models of physiological processes as they simplify modeling, abstraction and enable robustness [6]. Based on these realizations, Phy-SIM proposes two levels of modularity, *structural modularity* and *functional modularity*, which are new perspectives toward multilevel and multiscale integration of physiological processes. The proposed mechanism of functional modularity through the information flow approach is a novel contribution to the physiological model development domain. Integration of physiological processes is conceptualized by the transfer, access or sharing of information among the models representing the processes, and is defined as *information flow* by the authors. Structural modularity on the other hand is observable in the anatomical and physiological organization of the human body. Phy-SIM uses the ontological representation of the anatomical and physiological information to achieve structural modularity.

The heuristic guidelines in software engineering design principles aim for low coupling and high cohesion [7]. For the domains such as physiological system models, where the domain problem itself is inherently coupled and tangled, software engineering principles are very crucial. Therefore we adapt the similar design approaches in software engineering to make the problem more manageable for model developers by reducing the effects of high coupling in multiscale physiological models. As detailed in Section 4, layered design to achieve high modularity and the mediator design pattern are used to manage the communication among highly integrated modules. This way Phy-SIM achieves a software level improvement in the coupled nature of physiological models. The details of the proposed software level solutions for the multiscale physiological model integration and the sample use scenarios to show how these design decisions improve the model development process will be the focus of this paper.

2. Background

Integrative physiology following the emergence of systems biology is perceived to be central for better interpretation of physiological data starting from organ or system level down to genomic and proteomic data through the integration of these different levels of models [8]. In recent years, several big initiatives that try to create environments for researchers from various disciplines to achieve a collaborative environment and develop tools for integrative physiology research were launched. Digital Human Project introduces the idea of development of a “functional” visible human and emphasizes the importance of multilevel and multiscale modeling starting from system down to molecule level [9]. Physiome Project aims to build a database of physiological models with different scale and levels. Currently, models in this project are accessible through a web interface and some are supported with computer models [10]. Virtual Physiological Human Network of Excellence (VPH) aims to support research for developing tools, standards, models and simulations of the human body [11].

As the modeling efforts accelerated, the standards for storing and sharing mathematical models became very important especially for the cellular level data and models. CellML and SBML (Systems Biology Markup Language) are the most widely used standards storing and sharing of model descriptions. CellML is used for mathematical descriptions of cell functions. SBML defines an XML-based mark-up language standard for describing biochemical reaction networks models. Besides these domain specific standards, some physiological models are represented using MML (Mathematical Modeling Language).

Parallel to the modeling efforts, development of tools and software frameworks for integrative physiology studies have also gained attention. Physiome Project provides tools to enable integration with quantitative descriptions of relations among models and parameter sets to identify these relations. JSIM [12], which is a Java-based system, is used to simulate the models in Physiome model repository [13]. In [14], a new approach from a semantic simulation framework, SemSim, is presented for the integration of multiscale models of cardiac circulation. In this study the focus is on the integration of cross-platform models and translating procedural simulation code to a procedural code.

3. Objectives and barriers in multiscale physiological model integration domain—a software perspective

Phy-SIM aims to achieve the following in the domain of multiscale physiological model development and simulation:

1. Separation of structure from function: by the adopted modular design, definitions of mathematical models, anatomy and physiology are separated from the integration mechanism. The functional modularity is also achieved by defining protocols separately to handle information flow and integration which helps to deal with the highly coupled physiological models (detailed in Section 4.1.2).
2. Anatomical and physiological annotations: building complex models out of individual, multiscale models would be a challenging process. The challenge would increase as more collaborative efforts, share models developed by various institutions/experts. In order to integrate the models, there should be more metadata available for the models. Phy-SIM proposes an ontology based design for representing anatomical and physiological information associated with the mathematical models of physiological processes.

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

With the tools in Phy-SIM to access the metadata provided by the annotation process, access and thus integration of models would be a more systematic process.

3. Enhanced integration step: if the models are annotated well with the ontological information, determination of the integration points during model integration is easier and is guided by the framework.
4. Easy model sharing, extension and reusability: by providing the rich set of annotations (anatomical, physiological); sharing, extending, storing of models are easier. Moreover by the modular design, exchanging and reusing models as well as the integration approaches are possible.

Even though Phy-SIM proposes mechanisms to help developing integrated physiological models, it should be noted that due to the inherent problems, there are still some barriers the developers should take care during integration and simulation. The issue of computation time inherent in multiscale models is still an important challenge [15]. Implementation, testing and validation of integration approaches together with the readjustments of parameters and modifications on models due to integration remain as the processes that require supervision and management of the model developer. Although there are different approaches for multiscale integration of physiological models such as bottom-up, top-down or middle-out, it is clear that there cannot be one, unique algorithm for integrating various multiscale models. This is due to the fact that models differ based on the available level of information and desired resolution of integration results. Feed-down effects and upwards causations in the system requires a modeling approach which will have an interaction within and among various hierarchical levels [16]. There are different approaches to handle this highly coupled and integrative problem. Full-scale bottom-up approaches are hard to implement due to computational requirements. The other alternative, top-down approach lacks the accuracy bottom-up approaches will provide. Another approach, which can be thought of as a hybrid model is middle-out approach and it goes out to the other levels starting from the level with sufficient details. Application of any of these approaches may be valid based on the specific case. For instance, in [17] the results of the middle-out integration approach applied to the multiscale modeling of heart are presented. Therefore, integration of multiscale and multilevel models should be developed in environments where different approaches can be employed based on the detail of available models and computational requirements of the specific problem.

Moreover on the modeling side, as the coupling among physiological models is very high, integration of models often require managing this coupling at the model level by domain knowledge.

In linking the layers from various scales, the trade-off between accuracy and complexity stands as an issue [18]. While addressing this trade-off, biological facts, constraints and the robustness of models should be kept [19]. Since the decisions on the trade-off are case specific, such evaluations will remain as the developer's responsibility.

Following the stated objectives with the proposed methods, and also observing the inherent research problems mentioned above, which are outside the scope of the present study, we present a general simulation framework. The ultimate aim of Phy-SIM is to provide new perspectives to integrate multiscale models of physiological processes. Addressing all of the issues in the domain is not in the scope of Phy-SIM, but with the introduced principles and architecture, it will be a development framework to test various approaches for individual solutions. For instance, testing various multiscale integration approaches on a fixed set of models, would be highly facilitated with the modular design of Phy-SIM, which separates the models from the integration principles.

4. Overview of Phy-SIM

Phy-SIM is a modeling, integration and simulation environment for physiological models from tissue level up to organ-organism levels. Phy-SIM is designed as a layered system, separating anatomical, physiological and computational models from the functionality. The separated functionality includes both the domain functionality which is the integration of these models as well as from the application specific functionality, simulation. With this design decision reflected to the modeling process, Phy-SIM introduces a new perspective for multiscale model creation and integration, which is the separation of structure from function. Specifically Phy-SIM has three basic advantages over general purpose simulation tools. First, the ontology based design providing systematic domain knowledge representation for both anatomy and physiology. Second, the capability of easy annotation of models using the ontology based design. Third, modularization of the integration mechanism with the information flow mechanism, with which the development of various integration approaches will be possible besides the models themselves.

Target users of Phy-SIM are physiological model developers, who are familiar with the computational and integrative requirements of the domain, seeking collaborative work with integrative focus. Even though there are software tools as mentioned in Section 2 facilitating modeling and simulation of individual models, model developers need tools that will aid the process of integrating models to build more complex ones. Phy-SIM users can (i) create/save/load mathematical models of physiological processes, (ii) annotate the models with anatomical and physiological ontology, save/load this information, (iii) perform queries on the models of interest based on anatomical and physiological information and (iv) program various integration algorithms for models, perform the algorithm and test it. These use cases separate Phy-SIM from other biological simulation tools such as Matlab (Systems Biology Toolbox) [20] and JSIM [12].

Models of physiological processes to be loaded and saved are represented in Mathematical Modeling Language (MML) standard in Phy-SIM. However, import/export option for other standards such as Systems Biology Markup Language (SBML) will be available. For the simulation of computational events, Phy-SIM supports ordinary differential equations for continuous time events as well as discrete time and discrete event models. Structural modularity is achieved through the use of ontological representations of anatomical and physiological information.

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

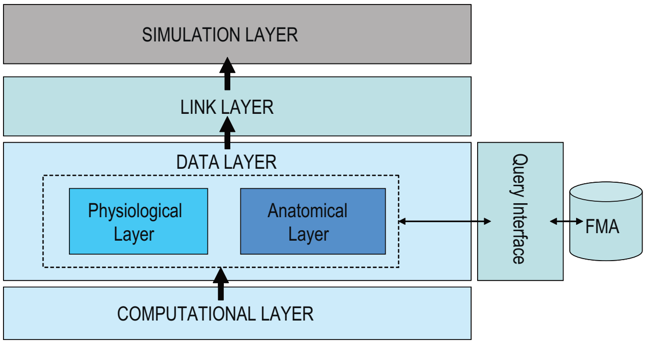


Fig. 1 – Phy-SIM System Architecture: Phy-SIM has a layered design composed of computational layer, data layer, link layer and simulation layer. The data layer is connected to the relational databases storing the anatomical ontology via a query engine interface.

Phy-SIM is designed to be an open source framework, where both easy plug-and-play type modeling and simulation is supported together with a programming interface for more ambitious integration projects. The Web Ontology Language (OWL), which is a semantic markup language used to represent, share and store the hierarchical structure of an entity, is used to represent the anatomical structure of the human body. The framework is developed in C++and the ontology data (anatomical ontology) in OWL format is stored in a relational database which uses Jena/ARQ query engine (in Java). Therefore to handle the communication between the application and the query engine, JNI interfacing is used (Fig. 1).

4.1. *System design of Phy-SIM*

A layered architecture is adopted for Phy-SIM as seen in Fig. 1. The four main layers are *computational layer*, *data layer*, *link layer* and *simulation layer*. The dependencies among the layers are in one direction keeping the coupling among separate layers low. Representation of structural information is divided into two lower layers, as *computational layer* and *data layer* and functional information is defined in *link layer*. Functional component of a physiological process is related to the coupling among models. Highly coupled functionalities of physiological processes determine the state of the organism, like homeostasis [21]. In addition to the coupling among physiological components, integration of multiscale models falls within our definition of domain specific functions, thus is separated from the structural components and defined in the *link layer*.

Computational layer is used to handle the mathematical representation of the physiological models. Mathematical equations describing the dynamics of models are defined at the computational layer. Simulation specific attributes, such as simulation step type, step size, simulation start and end times are defined and managed at the simulation layer.

4.1.1. *Data layer: structural modularity*

Data layer is responsible for the anatomical and physiological information of the models. Anatomical layer is designed to manage structural modularity using an ontology, Foundational Model of Anatomy (FMA) [22,23]. FMA represents the taxonomy and part-whole relations for the anatomical information. Phy-SIM adopted the abstraction mechanism proposed by anatomical ontology and anatomical structural abstraction components of FMA, which together provide the sufficient and necessary information for conceptualizing the hierarchical structure of the human body and its parts [22]. Every physiological variable in a process can be associated with an anatomical structure corresponding to an entity in the anatomical ontology. The taxonomic level of the association depends on the detail the developer knows or is interested in. Every physiological variable aggregates such an anatomical association defined in data layer. The FMA ontology is accessible to the users at various steps of the model development and integration steps. Taxonomic representation is easily accessible through a tree like presentation throughout the modeling process for performing the annotation or for reference (see Section 5.1).

Once the anatomical associations are bound to the physiological models, this information is saved in an XML file which later can be loaded together with the models. This structured information binding improves the integration, exchange and sharing of models and therefore increases the efficiency of collaborative works. The model developers benefit from the structured annotations during the integration processes. By the use of the querying interface and the anatomical associations, the developers view the models with these criteria. For instance, integrating a circulatory system model with an organ level model, requires to identify the physiological variables based on the anatomical associations such as the arterial supply of the organ and the constitutive relations of the arteries. In other words, standardized annotations of the models through the use of ontologies provide the mechanism to investigate, analyze the models and physiological variables with different perspectives to identify the correct models integration points. Such a capability would be crucial in particular for large scale models.

Similar to the anatomical layer, the physiological layer is designed to add the physiological associations of mathematical models using ontological representations. As there are currently no well established or comprehensive ontologies available for physiology, in the current implementation, Phy-SIM uses a high-level ontology like representation for physiological information. There are some proposed ontologies for physiological processes which are in preliminary form, and are not as established or comprehensive as, for example, the FMA. Foundational Model of Physiology (FMP) [24] and Ontology of Physics for Biology (OPB) [25] are examples of such efforts. Currently Phy-SIM only utilizes FMA for anatomical information. For physiological information representation, Phy-SIM uses a high level ontology like representation for physiological models, since FMP is not as comprehensive and available as FMA.

In the ontology-like representation used for physiology, physiological variables are defined based on their *qualitative*, *quantitative* and *temporal* attributes. Each physiological variable can have different temporal characteristics (*single time* or *spanning a time course*), can be *scalar* or *vector* valued and can be defined at a *single point* or over a *field*. Taxonomic organization of physiological variables are based on the type of information they carry and the information flow medium

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

they are being used: (i) circulatory physiological variable, (ii) nervous physiological variable and (iii) spatial physiological variable. Circulatory physiological variables are further specialized based on the type of information they carry. These variables are *pressure*, *flow*, *volume*, *temperature*, *concentration* variables and they determine the dynamics of the circulation.

Using modular design principles, Phy-SIM architecture is designed, such that, when a comprehensive physiological ontology becomes available, it can be directly used to replace the existing ontology-like representation. (The same advantage is also true for the modification or alteration of the anatomical ontology.)

4.1.2. *Link layer: functional modularity*

Link layer is the layer where the models are integrated based on the anatomical, physiological and computational information provided from the lower layers. The concept of a coupled dynamics among physiological variables is represented at this layer.

Physiological processes at the organ level models, are conceptualized by the physical laws, such as conservation of mass, momentum, etc. Hence physiological processes have high coupling with other organs and physiological variables [26]. Furthermore, integration of multilevel and multiscale models increases the couplings among processes.

The mathematical models for highly coupled physiological processes defining the current state of the whole system represent the regulation, control and modification of physiological variables [21]. A change in a physiological variable has a direct or indirect effect on processes determining other physiological variables in defining states of the system. In other words, every physiological variable carries an information which needs to be accessed, used, modified or integrated by other variables. Based on this observation integration of physiological processes is conceptualized by the transfer, access or sharing of information among the models representing the processes, and is defined as *information flow* by the authors. Integration of multiscale processes can also be defined with information flow. Such a processing among scales, or even within the same scale should be implemented independent from the definition of the models themselves. With the layered design proposed (see Fig. 1), link layer handles the interpretation, integration and thus the flow of information is defined at this layer.

In the presented study, integration of physiological processes are grouped into two basic groups based on the scale and level of the anatomical and physiological structure that the processes occur. *Vertical integration* refers to the integration of physiological processes via variables which carry information from different hierarchical levels of anatomical and physiological structures. *Horizontal integration* is the type of integration that uses the modularity for the biological components within the same level.

For the current status of Phy-SIM we have focused on modeling horizontal integration and horizontal information flow. However since the ultimate aim is to facilitate both vertical and horizontal integration, the proposed modular design mechanism with the information flow idea, enables the implementation of vertical integration approaches.

At the structural level, the nature of the horizontal interaction between the different components in complex biological systems is well-structured. The inputs and outputs of the organs are well-defined and limited to: (i) electrochemical signals transmitted through the nervous system, (ii) hormones and other material transmitted through the circulatory system, (iii) mechanical interaction with environment and neighboring structures, and (iv) material transport through the surfaces, which is significant only in limited cases, such as, skin, lungs, and gastrointestinal tract. Based on these observations, we made a generalization and defined the circulatory system and nervous system to be the major horizontal information flow mediums. Use of circulatory and nervous systems as the main mediums and separating the structural information from the functionality is based on the analogy with computer networks, where the communication protocols are defined separately from the hardware of the network. As separating the layers in computer networks improves the development of elements in each layer, abstracting the circulatory and nervous systems and modularizing their functionality independent of the hardware of the network, i.e. anatomical and physiological organization, enhances the development process of physiological models.

Extending the above analogy with layered network design, the presented framework defines the circulatory and nervous systems as the protocols where the information is carried. Protocols of the circulatory system correspond to the variables which determine the dynamics of the circulation, such as blood pressure, cardiac output, etc. Similarly in the nervous system, the protocols correspond either to electrical or chemical transmission of neural signals. The flow of information in the circulatory system can be thought of as a broadcasting mechanism, where information in the form of physiological variables are transported in the blood stream. On the other hand, the nervous system can be thought of as a point to point communication mechanism where the information in the form of electrical or chemical signals are transmitted.

There are three basic architectural elements that handle the information flow, *component*, *semantic converter* and *mediator* in link layer in Phy-SIM. Using these elements, every physiological model is defined as a graph as seen in Fig. 3.

Component: Every component aggregates an anatomical, physiological and computational information of a physiological variable (Fig. 3). There is a one-to-one relationship between every variable/parameter and the component in the model. Each component represents a node in the graph in link layer which are connected with directed links.

Semantic converter: This element defines the links between the components. The link that is defined with semantic converters not only define the direction of information flow among components but also provide a mechanism to define semantics of the flow. They behave like a programmable black box, where the information can be processed based on the integration approach. The functionality of these semantic converters depends on the specific models and the integration approach adopted. The programming facility of semantic converters is provided through the user interface, which helps to motivate the model developers to implement their models modularly separating the models themselves from the integration mechanisms. An example of a semantic converter

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

programming is depicted in Section 5.2. There are two types of semantic converters designed based on the integration, *vertical semantic converter* and *horizontal semantic converter*. With the current design of semantic converter, the semantics of vertical and horizontal integration can be implemented and extended depending on specific model requirements. The presented case scenario (Section 5), which is a horizontal integration case, maps the variables and parameter to perform the integration. However, the task of semantic converter can be as elaborate as the aggregation of variables from a lower scale so that the information can be used in the higher scales. Moreover, dispersion of a variable to a set of various sub models, conversion of units or physical meaning (like converting flow to a pressure variable) can be performed with the programmable black-box architecture of semantic converters.

One specific type of semantic converter, namely the *replacer semantic converter* is available to be used in the plug and play user interface. This semantic converter is for horizontal integration, where a variable or a parameter is to be replaced with another variable or a parameter from another model. The user provides the inputs and outputs to this semantic converter, and the required replacements are done automatically within the models and thus within the graph represented in the link layer. This type of semantic conversion modifies the ordering of the nodes since the links are redefined by the use of the replacer semantic converter. In Section 5.2, we will show the sample uses of a replacer semantic converter and the mechanism to perform a modification on a model upon integration.

Mediator: This is the control unit of the graph, which is created by components and the semantic converters. The topology of the graph is perceived by the mediator, so that the serializable ordering of the mathematical models and mathematical cycle resolution steps can be performed before the simulation layer can execute the simulation itself. Once the topology of the graph is tested by the mediator, non-algebraic cycles are resolved and a directed acyclic graph is created by the mediator. Until the link layer, there is no connectivity information attached to the nodes representing the physiological variables in the model. The nodes contain only anatomical, physiological and computational information. It is the responsibility of the mediator to *compile* the whole model to build an order of execution to be passed to the simulation layer. Performing validity checks on the models to avoid algebraic loops is another responsibility of the mediator. Once the topology of the graph is tested by the mediator, non-algebraic cycles are resolved and a directed acyclic graph is created by the mediator. Directed acyclic graph is important to achieve a sequential simulation in the simulation layer. For instance, if we extract a couple of equations from the gas exchange model [27] in consideration as follows:

$$
\begin{aligned}
&event(t + t.delta >= (t_{shift} + ((1/HR) * 59.999999999999986)))\\
&T_{sv} = (T_{svK} * sqrt((1/HR)/1)) * 7.745966692414832;\\
&T_{sa} = (T_{saK} * sqrt((1/HR)/1)) * 7.745966692414832;\\
&t_{shift} = t;\\
&afs - con2 = afs - con;
\end{aligned}
$$

In this example every variable ($T_{sv}, T_{sa}, etc.$) are represented as nodes and there should be an order imposed during the simulation of these equations due to their dependencies. In this case, $afs - con2$ depends on the current value of $afs - con$ and T_{sa} depends on T_{saK}. Therefore, when the node information is passed to the mediator together with the equations, the mediator will compile the model and create the directed links so that the node T_{saK} and thus its calculation will precede the node T_{sa} (see Fig. 2(a) and (b)). It is the mediator's responsibility to create the correct topology parsing the given equations. In doing this if the complete graph contains loops due to the initial value problems, due to differential equations, it will resolve it automatically. However, the algebraic loops are not resolvable and it is the mediator's responsibility to warn the model developer to correct this loop.

If we revisit the broadcasting network analogy presented before; in circulatory system through the bloodstream, any information like the cardiac output is disseminated to the whole system. In this sense, circulatory system is the broadcasting system where there is not a specific destination for specific information. On the other hand for the nervous system, the information is specific to a destination, carried either by chemical or electric impulses. This specification overlaps with a point-to-point network communication. Having made this analogy, we can get the benefit of modular design in Phy-SIM and introduce modular definition in the link layer using the mediator. For any specific mechanics determining the underlying network structure for the specific type of information flow, mediators can be extended accordingly. In the presented sample scenario of the integration of cardiopulmonary mechanics and the gas exchange model (see Section 5.2), the circulatory system is the main medium for the dissemination of information. Therefore in this pilot scenario, we implemented the broadcasting mechanism within the mediator, where any information provided by the cardiopulmonary mechanics model or the gas exchange model is made available to all the sub models accessing the circulatory system. In Fig. 4, the conceptual perception of the broadcasting mechanism can be visualized. Mediator is responsible for the availability of this information to the other sub models in the medium. The extended mediator, *circulatory mediator* implements this mechanism.

5. Case study

The case models used in the following two sections are chosen from the Physiome repository [13]. First model is the cardiopulmonary mechanics model with four-chamber varying-elastance heart, pericardium, systemic circulation, pulmonary circulation and coronary circulation, baroreceptor and airway mechanics models [27]. In the Physiome repository more complex cardiopulmonary mechanics models are developed incrementally by adding sub-models, such as gas exchange, to this model.

5.1. *Anatomical annotation scenario*

In this section, we will illustrate how the ontology based design improves the integration of physiological models using example models obtained from Physiome repository.

For the demonstration of anatomical annotation we isolated the gas exchange model from the compound model, cardiopulmonary mechanics model with gas exchange. In

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

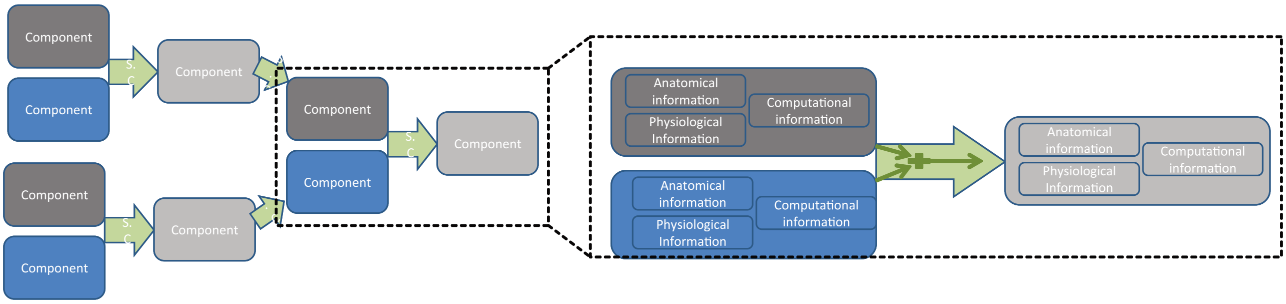


Fig. 2 – Graph representation of a physiological process. The zoomed part of the graph shows that the components encapsulating structural and computational information. Semantic converters define the links between the nodes in the graph and provide a programmable unit to implement the semantics of the flow of information.

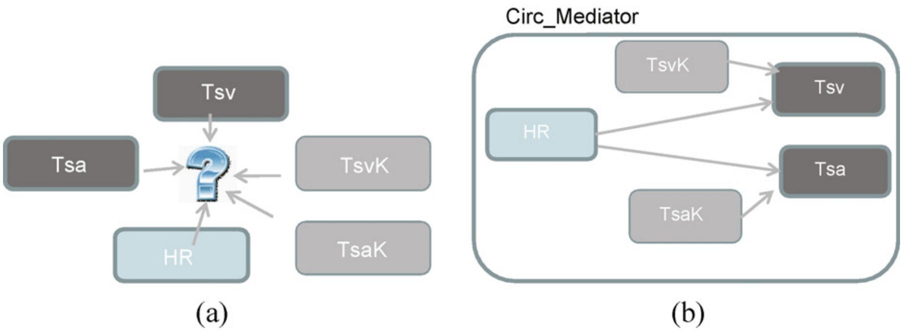


Fig. 3 – (a) The nodes before the mediator performs any graph theoretic operations, do not have the concept of order, direction or topology. (b) Once the mediator creates a topology from the individual nodes using their computational information a directed acyclic graph is created.

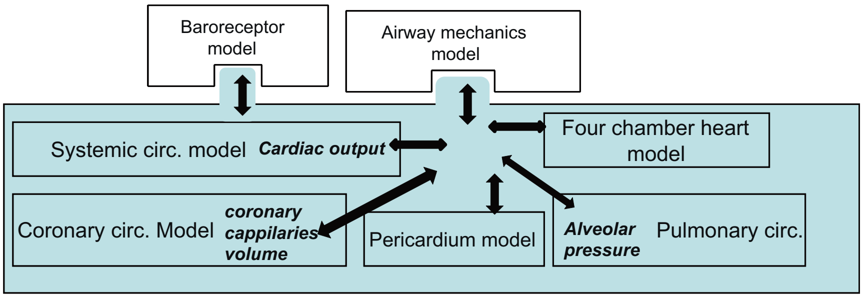
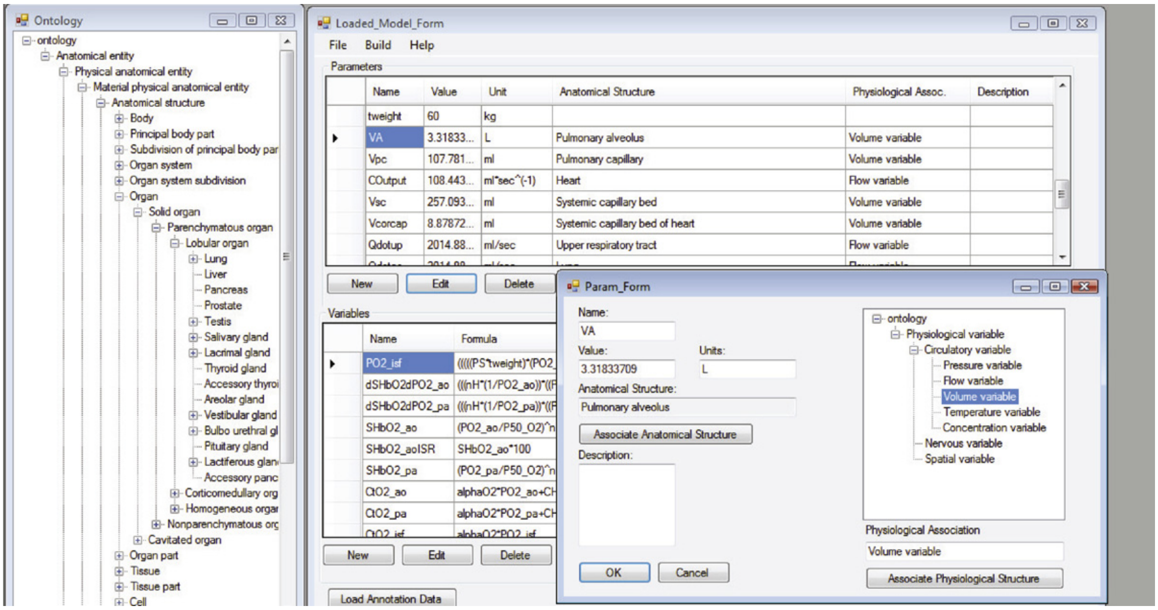


Fig. 4 – Circulatory system behaving as a broadcasting network. The extended mediator, *circulatory mediator* implements this mechanism.

the cardiopulmonary mechanics; gas exchange model is coupled with the pulmonary circulation model, airway mechanics model, heart model, systemic circulation and coronary circulation through a set of variables. These variables are alveolar volume (*VA*), pulmonary capillaries volume (*Vpc*), cardiac output (*COutput*), systemic capillaries volume (*Vsc*), coronary capillaries volume (*Vcorcap*), flow through upper airways (*Qdotup*), flow through collapsible airways (*Qdotco*), flow through small airways (*Qdotsm*), collapsible airways chamber pressure (*Pcc*), alveolar pressure (*PA*), lung dead space (*VD*), volume of collapsible airways (*Vc*) and dead space chamber pressure (*PDc*). These variables are used to define the dynamics of oxygen and carbon dioxide pressure, concentration and saturation in the gas exchange model. These variables (*VA*, *Vpc*, *COutput*, *Vsc*, *Vcorcap*, *Qdotup*, *Qdotco*, *Qdotsm*, *Pcc*, *PA*, *VD*, *Vc*, and *PDc*) are defined as constants/parameters in the cardiopulmonary mechanics model before the integration. These variables are used to define the dynamics of oxygen and carbon dioxide pressure, concentration and saturation in the gas exchange model. In isolating the gas exchange model, we redefined the variables as parameters with their initial values and created a standalone model which can be simulated as is.

Identification of above mentioned parameters and variables in both models are critical to perform the integration of these models. Therefore, an automated, standardized mechanism is essential. As seen in Fig. 5, Phy-SIM gives a user friendly interface to annotate the variables. While the models

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010



Parameters

Name	Value	Unit	Anatomical Structure	Physiological Assoc.	Description
tweight	60	kg			
VA	3.31833...	L	Pulmonary alveolus	Volume variable	
Vpc	107.781...	ml	Pulmonary capillary	Volume variable	
COutput	108.443...	ml*sec^(-1)	Heart	Flow variable	
Vsc	257.093...	ml	Systemic capillary bed	Volume variable	
Vcorcap	8.87872...	ml	Systemic capillary bed of heart	Volume variable	
Qdotup	2014.88...	ml/sec	Upper respiratory tract	Flow variable	

Variables

Name	Formula
PO2_isf	(((((PS*tweight)*(PO2_
dSHbO2dPO2_ao	(((nH*(1/PO2_ao))*((F
dSHbO2dPO2_pa	(((nH*(1/PO2_pa))*((F
SHbO2_ao	(PO2_ao/P50_O2)^n
SHbO2_aoISR	SHbO2_ao*100
SHbO2_pa	(PO2_pa/P50_O2)^n
CtO2_ao	alphaO2*PO2_ao+CH
CtO2_pa	alphaO2*PO2_pa+CH

Name: VA
Value: 3.31833709
Units: L
Anatomical Structure: Pulmonary alveolus
Description:
Physiological Association: Volume variable

Fig. 5 – Annotation interface of Phy-SIM: physiological variables of models are listed and the mechanism to select an anatomical entity from the tree like representation of the FMA ontology is provided. The annotations can be saved in XML file to be loaded later with the corresponding MML file.

are being loaded on the right hand side frame, the navigation on the tree representation of the anatomical ontology is possible on the left of the frame. The models are presented as variables and parameters each of which is detailed with the equation describing its dynamics, initial value, unit and description. For each parameter, Phy-SIM provides the mathematical information such as the value, unit and anatomical and physiological information. For each variable mathematical information (initial value, unit, equation) and structural information (anatomical and physiological association) is also provided. Also the reference ontology for anatomical and physiological information are presented with tree like structures for easy access. The equation for each physiological variable indicates the dynamics that is associated with that variable. The interface provides the facility to analyze and edit the individual variables/parameters with their computational, anatomical and physiological attributes clearly marked and separated. Instead of removing the complexity of the models, this interface provides a modular representation for each type of information. For instance, model developer interested in the anatomical annotation of the models, would be able to analyze and edit the corresponding attributes by use of tree like representations of the ontologies.

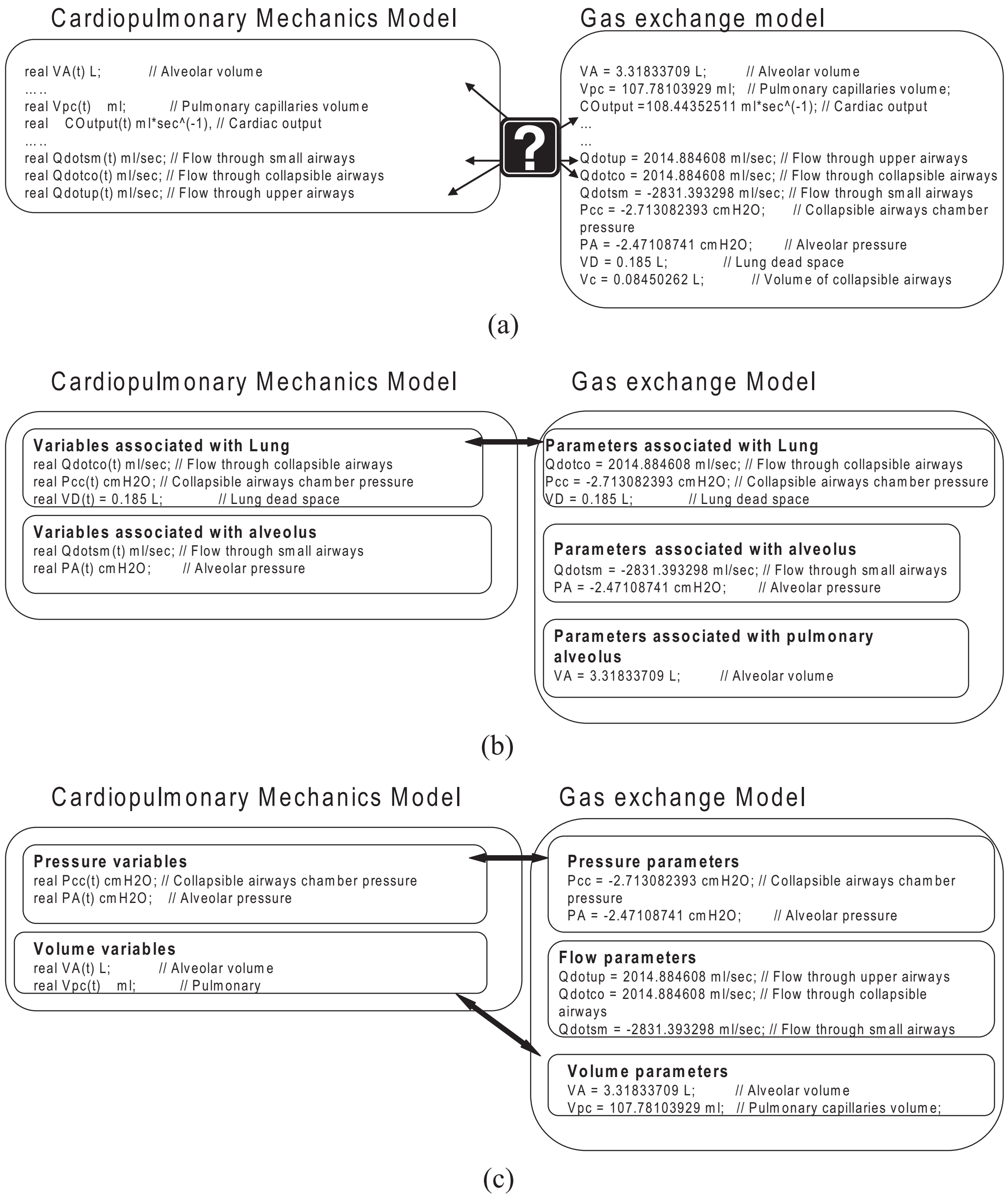
Based on the hierarchical information provided by the individual parameters' and variables' anatomical and physiological associations, and relationships among them, model developer will decide on the proper integration approach for the multiscale models. This is the reason why we emphasized the representation of structural information in the user interface. The annotations are saved in an XML file, which can later be loaded together with the model files (MML files).

Based on the level of anatomical information available, developers can query the FMA ontology to retrieve the associated anatomical structure using the querying interface in Phy-SIM. Since the ontology provides a taxonomic and part-whole relationship, different levels of anatomical structure are used to annotate the models. For instance, in the presented case of cardiopulmonary mechanics model, a physiological variable representing the pressure at the systemic circulation can be associated as *vena cava*, as *systemic vein* or just as *vein* depending on the detail of the model itself.

Investigating the two models in detail without the use of any anatomical or physiological information and trying to find out the integration points would be very inefficient (Fig. 6(a)). However, if one would be able to investigate the model based on their anatomical associations, and ask questions such as: *What are the physiological variables/parameters which have the lung as an anatomical association?* or *What are the physiological variables/parameters which have an anatomical association with an organ part of lower respiratory tract?*; then investigation of the results of such an inquiry would provide a more systematic approach to perform the integration of the models over these variables (Fig. 6(b)). Similarly based on the annotations performed, the models can be analyzed with the physiological view as seen in Fig. 6(c) and performing queries to handle questions such as: *What are the physiological variables/parameters which are used to define a pressure?*

As seen in Fig. 7(a) and (b), Phy-SIM currently provides the different interfaces to query the models based on anatomical information associations. This mechanism queries the database storing the anatomical ontology based on the taxonomical and other relations. Based on the retrieved anatomical entity results, the models are searched for corresponding annotation. The first querying option is based on an RDF query language, SPARQL, and requires the user to type the query to retrieve the anatomical entities. This mechanism

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010



```
real VA(t) L;            // Alveolar volume
.....
real Vpc(t)    ml;           // Pulmonary capillaries volume
real    COutput(t) ml*sec^(-1), // Cardiac output
.....
real Qdotsm(t) ml/sec; // Flow through small airways
real Qdotco(t) ml/sec; // Flow through collapsible airways
real Qdotup(t) ml/sec; // Flow through upper airways
```

```
VA = 3.31833709 L;          // Alveolar volume
Vpc = 107.78103929 ml;   // Pulmonary capillaries volume;
COutput =108.44352511 ml*sec^(-1); // Cardiac output
...
...
Qdotup = 2014.884608 ml/sec; // Flow through upper airways
Qdotco = 2014.884608 ml/sec; // Flow through collapsible airways
Qdotsm = -2831.393298 ml/sec; // Flow through small airways
Pcc = -2.713082393 cmH2O;       // Collapsible airways chamber
pressure
PA = -2.47108741 cmH2O;        // Alveolar pressure
VD = 0.185 L;              // Lung dead space
Vc = 0.08450262 L;           // Volume of collapsible airways
```

```
Variables associated with Lung
real Qdotco(t) ml/sec; // Flow through collapsible airways
real Pcc(t) cmH2O; // Collapsible airways chamber pressure
real VD(t) = 0.185 L;              // Lung dead space

Variables associated with alveolus
real Qdotsm(t) ml/sec; // Flow through small airways
real PA(t) cmH2O;        // Alveolar pressure
```

```
Parameters associated with Lung
Qdotco = 2014.884608 ml/sec; // Flow through collapsible airways
Pcc = -2.713082393 cmH2O; // Collapsible airways chamber pressure
VD = 0.185 L;              // Lung dead space

Parameters associated with alveolus
Qdotsm = -2831.393298 ml/sec; // Flow through small airways
PA = -2.47108741 cmH2O;        // Alveolar pressure

Parameters associated with pulmonary
alveolus
VA = 3.31833709 L;          // Alveolar volume
```

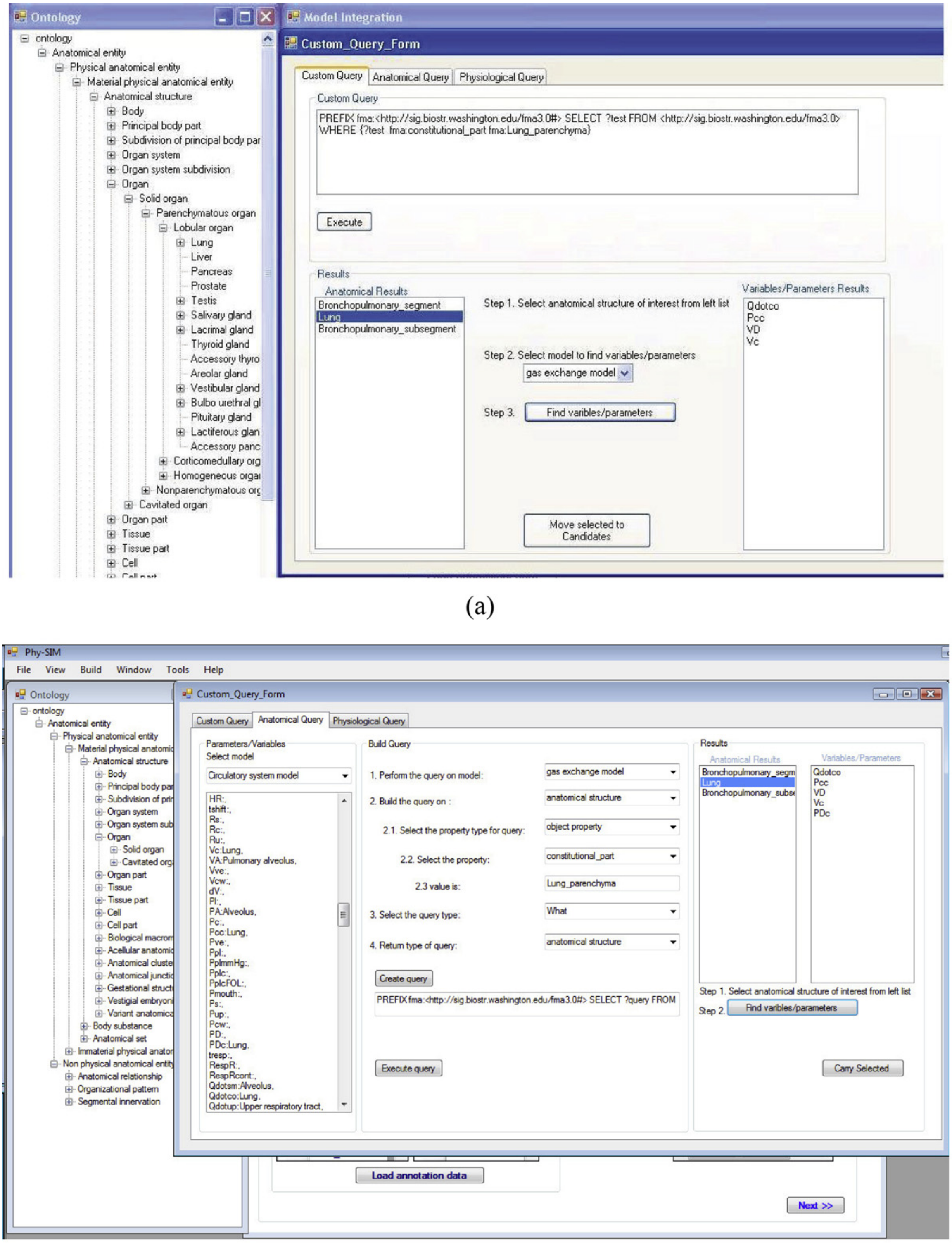
```
Pressure variables
real Pcc(t) cmH2O; // Collapsible airways chamber pressure
real PA(t) cmH2O;    // Alveolar pressure

Volume variables
real VA(t) L;           // Alveolar volume
real Vpc(t)    ml;           // Pulmonary
```

```
Pressure parameters
Pcc = -2.713082393 cmH2O; // Collapsible airways chamber
pressure
PA = -2.47108741 cmH2O;        // Alveolar pressure

Flow parameters
Qdotup = 2014.884608 ml/sec; // Flow through upper airways
Qdotco = 2014.884608 ml/sec; // Flow through collapsible
airways
Qdotsm = -2831.393298 ml/sec; // Flow through small airways

Volume parameters
VA = 3.31833709 L;          // Alveolar volume
Vpc = 107.78103929 ml;   // Pulmonary capillaries volume;
```

Fig. 6 – (a) Without the ontology associations, interpretation of the physiological variables and their domains is difficult to resolve. With the annotation and hierarchical organization provided by the ontology, physiological variables and parameters can be viewed from the desired systematic view, i.e. anatomical as in (b) or physiological as in (c).

provides the flexibility to perform any kind of query as seen in Fig. 7(a). The text box on the upper part of the interface is for typing any type of custom SPARQL query. The query is executed in the anatomical ontology (FMA) stored in the local database. The result is therefore a list of anatomical entities listed in the lower left list box (bronchopulmonary segment, lung and bronchopulmonary subsegment). Based on the results returned, the model developer further queries the model(s) under development or integration process, assuming they have already been annotated with the anatomical ontology. The steps in the middle part of the frame present, the order of actions to be taken by the model developer. First the interested anatomical entity from the result of the query in the left list box should be selected. Then the model to be queried based on the selected anatomical entity is to be determined. Once the selections are made, the results are returned on the right list box. In the presented case in Fig. 7(a), the variables which have been annotated with the anatomical entity *lung* are listed. This way the model developer would be able to filter the variables based on the anatomical associations

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010



Custom Query

PREFIX fma:<http://sig.biostr.washington.edu/fma3.0#> SELECT ?test FROM <http://sig.biostr.washington.edu/fma3.0>
WHERE {?test fma:constitutional_part fma:Lung_parenchyma}

Results

Anatomical Results

Bronchopulmonary_segment
Lung
Bronchopulmonary_subsegment

Step 1. Select anatomical structure of interest from left list

Step 2. Select model to find variables/parameters: gas exchange model

Step 3. Find varibles/parameters

Variables/Parameters Results

Qdotco
Pcc
VD
Vc

(a)

Parameters/Variables

Select model: Circulatory system model

HR:,
tshift:,
Rs:,
Rc:,
Ru:,
Vc:Lung,
VA:Pulmonary alveolus,
Vve:,
Vcw:,
dV:,
Pl:,
PA:Alveolus,
Pc:,
Pcc:Lung,
Pve:,
Ppl:,
PplmmHg:,
Pplc:,
PplcFOL:,
Pmouth:,
Ps:,
Pup:,
Pcw:,
PD:,
PDc:Lung,
tresp:,
RespR:,
RespRcont:,
Qdotsm:Alveolus,
Qdotco:Lung,
Qdotup:Upper respiratory tract,

Build Query

1. Perform the query on model: gas exchange model

2. Build the query on : anatomical structure

2.1. Select the property type for query: object property

2.2. Select the property: constitutional_part

2.3 value is: Lung_parenchyma

3. Select the query type: What

4. Return type of query: anatomical structure

PREFIX fma:<http://sig.biostr.washington.edu/fma3.0#> SELECT ?query FROM

Results

Anatomical Results

Bronchopulmonary_segm
Lung
Bronchopulmonary_subse

Variables/Parameters

Qdotco
Pcc
VD
Vc
PDc

Step 1. Select anatomical structure of interest from left list

Step 2. Find varibles/parameters

(b)

Fig. 7 – (a) Custom query interface for querying physiological variables based on anatomical associations. The user can type the SPARQL and get the list of physiological variables based on the retrieved anatomical association. (b) Anatomical query interface in Phy-SIM: this interface helps the users to build anatomical queries using the predetermined sets of predicates. The corresponding SPARQL is created and executed.

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010

which would help them through the integration process. The other querying mechanism as seen in Fig. 7(b), helps the users to build the query by providing the predefined query type. The user is given the options to select the model to perform the query on, the anatomical structure and the type of anatomical relation for that structure. Based on the selections made by the user, the SPARQL query is automatically generated and executed.

5.2. *Integration scenario*

Semantic converter, detailed architecturally in Section 4.1.2, handles the programming of the integration mechanism. In this section we will show how such a user programmable element will help the integration of building more complex models. The case models are chosen from the Physiome repository. In the Physiome repository cardiopulmonary mechanics models are developed incrementally by adding sub-models, such as gas exchange or airways mechanics models to the core circulation model. This incremental procedure is actually a very common practice and will be applied more due to the availability of various models from different researchers. For instance a subgroup from VPH [28] integrated a pulsatile ventricular model to the cardiovascular model developed by Guyton et al. [29] to have a more detailed model of the cardiovascular system to represent various pathologies. We will demonstrate how such an integration will be performed with Phy-SIM framework.

The steps as perceived from the user perspective in performing the integration are as follows:

- Load the models to be integrated: the cardiopulmonary mechanics model and the gas exchange model will be loaded from the MML files. The models will be visualized based on the parameters and variables. If anatomical or physiological associations are added to these model parameters or states, these associations will be saved and can be loaded for later use (see Fig. 8(a)).
- Identify the integration points: once the models and their associations are successfully loaded, the second step is to identify the parameters and variables through which the integration will be performed. This can be done by either hand selecting the desired variables or performing queries on the list of variables based on the anatomical and physiological annotations (see Fig. 8).
- Program the integration: since the models can vary in scale and level, integration approaches are dependent on these variations. The very basic integration approach, which replaces a variable or a parameter with another parameter or variable in the other model is represented with the *replacer semantic converter* as described previously. The user can select this functionality and the replacement will be performed automatically. The replacement will effect the oxygen/carbon dioxide pressure, concentration and saturation, which will now consider the dynamics related to the cardiopulmonary mechanics variables. The replacer semantic converter will take the parameter to be replaced as input and the variable to replace as the output. Any additional programming will not be required for the semantic converter. However in the topological level, the links between nodes will be updated and thus the dynamics will change. Upon integrating the gas exchange model with the cardiopulmonary mechanics model, equations for carbon dioxide and oxygen pressures across alveoli require a modification to consider the rate of change of alveolar volume. The user can perform such modifications in models, by selecting the modifier option in the integration step, and by typing the updated equation (see Fig. 8(b)). This operation will be treated differently than a semantic converter; since it will not only change the topology but it will also change the structure of the nodes in the graph.
  For multiscale integration cases, more complex integration approaches are required, such as vertical integration. For such cases, the user defines the input–output of the semantic converter and programs the semantic converter (see Fig. 8(b)). In such cases, where a domain expert knowledge is required to define how to map variables in models from different scales, the programming capability is required. Therefore, the textbox in Fig. 8(b) is provided to the model developer to type in the equation(s) that the semantic converter would implement to integrate the multiscale models.
- Compile the model: once the integration steps are complete, the compilation goes through the model definition files, modifications and programmed semantic converters to test the validity of the models. Once the model is identified as being valid, the graph representation of the model(s) are created by the mediator as described in Section 4.1.2. The topological updates due to the semantic converters and modifications are handled at this step. Later, simulation specific parameters such as simulation step are identified.

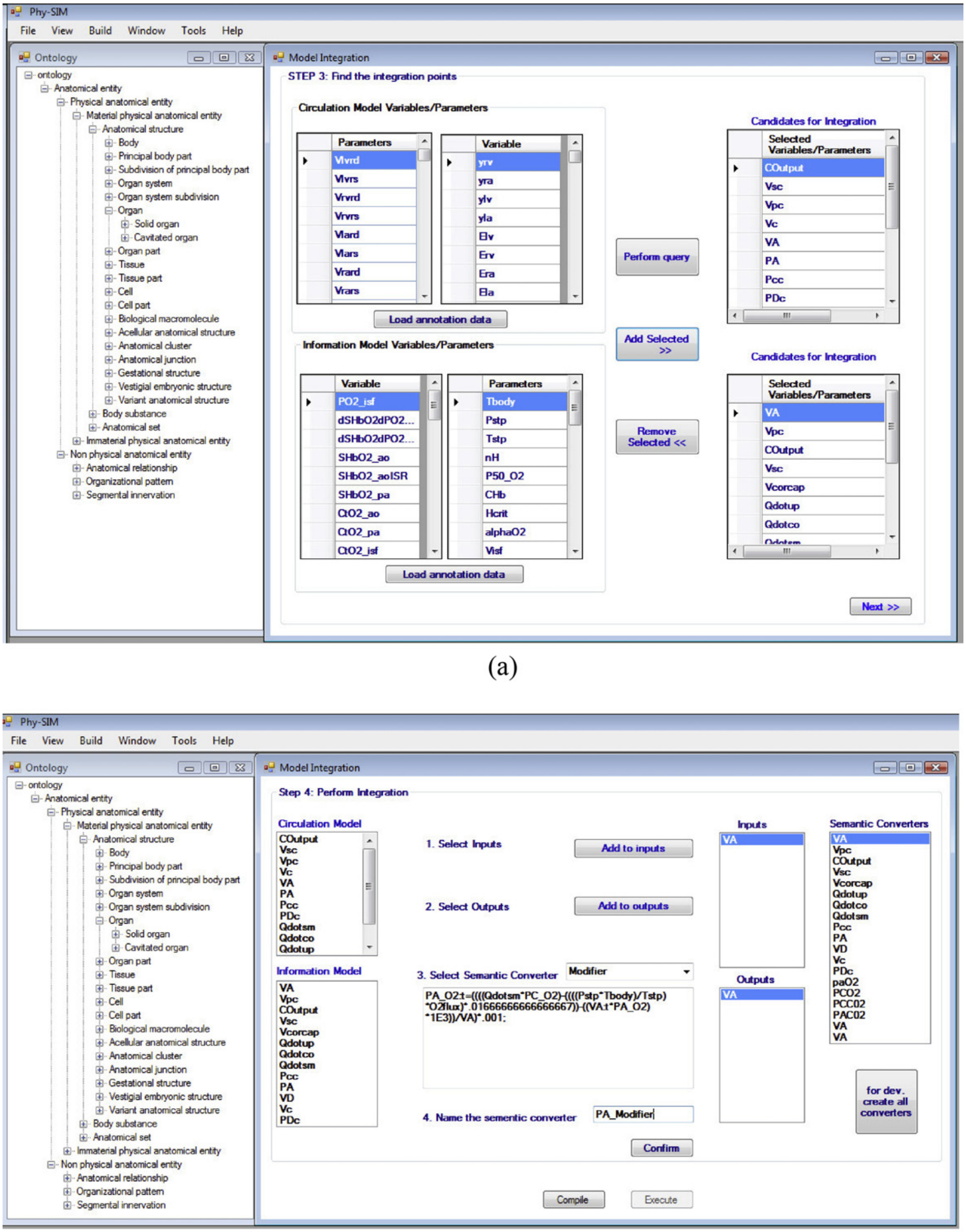
5.3. *Graphical representation of simulation results*

The results of simulations in Phy-SIM are saved in a format suitable to load in MATLAB. Moreover the users can choose to visualize the simulation results with the plotting facility provided by Phy-SIM. For this purpose we have used a cross-platform software package for creating scientific plots, “PLplot” [30]. The package provides the tools to plot the data with different chart types, axis/graph names and output formats. Currently minimal set of options are provided within Phy-SIM to present the graphical results (see Fig. 9).

6. Current status and mode of availability of the framework

Design and implementation of the components to execute the major functionality of Phy-SIM is complete. Once the design of Phy-SIM was complete, the implementation was on the scope of the sample models (see Section 5) with the features that would be helpful to present as a proof-of-concept, such as modular representation of physiological models and the representation of information flow among physiological models to improve the integration process. Phy-SIM is still under development for beta release and more advanced features will be implemented in the future. Phy-SIM is designed to be an open source framework and the current version is available upon request from the authors via e-mail. The details of the

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010



STEP 3: Find the integration points

Circulation Model Variables/Parameters

Parameters	Variable
Vlvrd	yrv
Vlvrs	yra
Vrvrd	ylv
Vrvrs	yla
Vlard	Elv
Vlars	Erv
Vrard	Era
Vrars	Ela

Information Model Variables/Parameters

Variable	Parameters
PO2_isf	Tbody
dSHbO2dPO2...	Pstp
dSHbO2dPO2...	Tstp
SHbO2_ao	nH
SHbO2_aoISR	P50_O2
SHbO2_pa	CHb
CtO2_ao	Hcrit
CtO2_pa	alphaO2
CtO2_isf	Visf

Candidates for Integration

Selected Variables/Parameters
COutput
Vsc
Vpc
Vc
VA
PA
Pcc
PDc

Candidates for Integration

Selected Variables/Parameters
VA
Vpc
COutput
Vsc
Vcorcap
Qdotup
Qdotco
Qdotsm

(a)

Step 4: Perform Integration

Circulation Model: COutput, Vsc, Vpc, Vc, VA, PA, Pcc, PDc, Qdotsm, Qdotco, Qdotup

Information Model: VA, Vpc, COutput, Vsc, Vcorcap, Qdotup, Qdotco, Qdotsm, Pcc, PA, VD, Vc, PDc

1. Select Inputs

2. Select Outputs

3. Select Semantic Converter: Modifier

```
PA_O2:t=((((Qdotsm*PC_O2)-((((Pstp*Tbody)/Tstp)
*O2flux)*.01666666666666667))-((VA:t*PA_O2)
*1E3))/VA)*.001;
```

4. Name the sementic converter: PA_Modifier

Inputs: VA

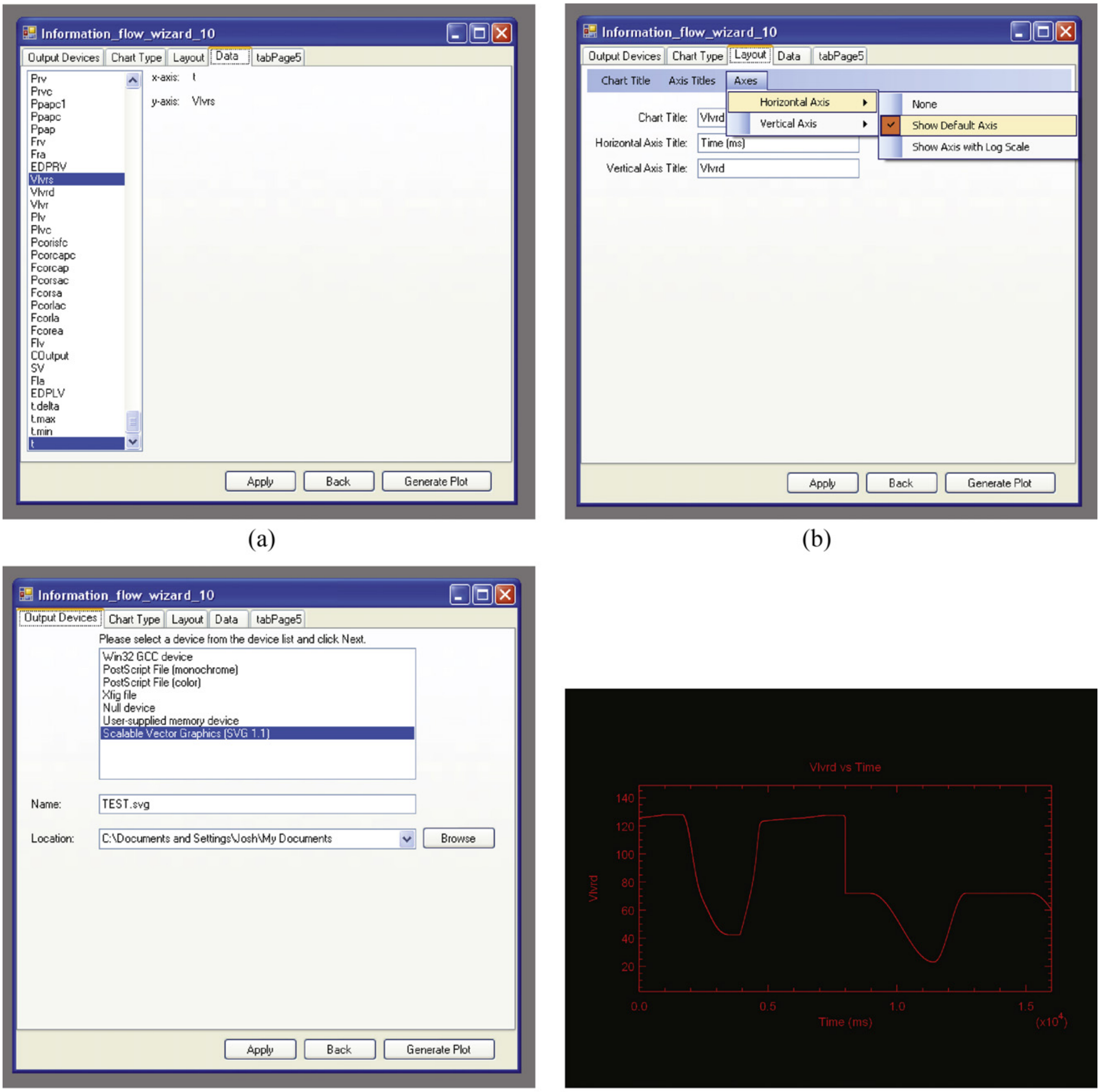
Outputs: VA

Semantic Converters: VA, Vpc, COutput, Vsc, Vcorcap, Qdotup, Qdotco, Qdotsm, Pcc, PA, VD, Vc, PDc, paO2, PCO2, PCC02, PAC02, VA, VA

(b)

Fig. 8 – Screen shots for integration of circulatory model with the gas exchange model. (a) The information model, gas exchange model is loaded. The variables and parameters of the models are listed. The user can select the parameters/variables from these list to perform the integration. (b) The selected parameters/variables can be assigned as inputs/outputs to the semantic converter, based on the type of integration specific type of semantic converter can be selected and using the text box provided, the programming of the semantic converter can be performed.

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010



x-axis: t

y-axis: Vlvrs

(a)

Chart Title: Vlvrd

Horizontal Axis Title: Time (ms)

Vertical Axis Title: Vlvrd

(b)

Please select a device from the device list and click Next.

Name: TEST.svg

Location: C:\Documents and Settings\Josh\My Documents

(c)

(d)

Fig. 9 – Plotting of simulation results in Phy-SIM. In (a), the user is given the list of variables to select for plotting on x and y coordinates. In (b) naming of the chart, axis titles are entered. In (c), the type of the graph is selected and the name is given to the image file. In (d), the graph is plotted with determined attributes.

project and demo movies describing the details of a case scenario can be found at http://robotics.case.edu/PhySIM.

7. Discussion and conclusions

Due to the complex nature of the physiological systems, development, integration of multiscale models and linking the layers stand as one of the challenges for the model developers. Besides the computational issues in multiscale models and the barriers presented in Section 3, high coupling within and among scales is a problem for performing the integrative studies. It is clear that to study the complex system in detail, integrative approaches should be adopted, for which the software frameworks, like Phy-SIM will be the biggest aid to overcome these problems. Hence in contrast to trying to solve the challenge in the mathematical model development level, we propose to minimally reflect the challenges to the developer during model integration process. In order to achieve this, we propose a modular programming environment, which will promote the separation of structure and function in the physiological models. However it should be noted that the model developers still face the issues raised by the multiscale model integration domain. Identification of the integration approach, especially for the multiscale models depend on the level of detail available to the developers, computational constraints and also to the desired resolution of simulation results. Development of the integration approach and its application to the models with the correct set of parameters, initial conditions and assumptions are still the responsibilities of the model developer.

For the current status of Phy-SIM we have focused on modeling horizontal integration and horizontal information flow. However since the ultimate aim is to facilitate both vertical and horizontal integration, the proposed modular design mechanism with the information flow idea, enables the implementation of vertical integration approaches. Since the target users are model developers involved in collaborative studies for multiscale physiological models, Phy-SIM's user interface will be enhanced based on the feedbacks from the users. As Phy-SIM evolves, addition of more user-friendly interfaces using standard representations would help to expand the users to non-experts. Integration of the parts plays a big role in defining the whole complete organism, which can be perceived as building a jigsaw puzzle [31]. Phy-SIM provides a development environment for this jigsaw puzzle, where physiological models can be integrated into the complex whole and simulated. The two levels of modularity proposed in Phy-SIM, structural modularity and functional modularity, are new perspectives toward multilevel and multiscale integration of physiological processes. As the model developers use Phy-SIM for their model integration needs, they will also benefit from the modular thinking Phy-SIM advocates. For instance, testing various multiscale integration approaches on a fixed set of models, would be highly facilitated with the modular design of Phy-SIM, which separates the models from the integration principles. The ultimate goal is to enhance the development of both the physiological models and the integration approaches.

Conflict of interest

None declared.

REFERENCES

[1] H. Kitano, Systems biology: a brief overview, Science 295 (2002) 1662–1664.
[2] D. Noble, Systems biology and the heart, Biosystems (2006).
[3] J. Bassingthwaighte, P. Hunter, D. Noble, The cardiac physiome: perspectives for the future, Experimental Physiology (2009).
[4] P. Kohl, N. D, Systems biology and the virtual physiological human, Molecular Systems Biology (2009).
[5] D. Floreano, S. Mitri, S. Magnenat, L. Keller, Evolutionary conditions for the emergence of communication in robots, Current Biology 17 (2007) 514–519.
[6] M.E. Csete, J.C. Doyle, Reverse engineering of biological complexity, Science 295 (2002) 1664–1669.
[7] R.S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Higher Education, 2001, ISBN 0072496681.
[8] E.J. Crampin, M. Halstead, P. Hunter, P. Nielsen, D. Noble, N. Smith, et al., Computational physiology and the physiome project, Experimental Physiology 89 (2003) 1–26.
[9] G. Higgins, B. Athey, J. Bassingthwaighte, J. Burgess, H. Champion, K. Cleary, et al., Final report of the meeting "modeling & simulation in medicine: towards an integrated framework", Computer Aided Surgery (2001) 32–39.
[10] Physiome project. Available from: http://www.physiome.org/, 2011.
[11] Virtual physiological human network of excellence. Available from: http://www.vph-noe.eu/, 2011.
[12] JSIM. Available from: http://www.physiome.org/jsim/, 2011.
[13] Physiome project model repository. Available from: http://www.physiome.org/model/doku.php, 2011.
[14] M.L. Neal, J.H. Gennari, T. Arts, D. Cook, Advances in semantic representation for multiscale biosimulation: a case study in merging models, in: Proceedings of the Pacific Symposium on Biocomputing, 2009, pp. 305–315.
[15] J.B. Bassingthwaighte, H.J. Chizeck, L.E. Atlas, Strategies and tactics in multiscale modeling of cell-to-organ systems, in: Proceedings of the IEEE; vol. 94, 2006, pp. 819–831.
[16] D. Noble, The future: putting humpty-dumpty together again, Biochemical Society Transactions (2003) 31.
[17] P. Kohl, D. Noble, R.L. Winslow, P.J. Hunter, Computational modelling of biological systems: tools and visions, Philosophical Transactions: Mathematical, Physical and Engineering Sciences, 2000.
[18] J.B. Bassingthwaighte, H.J. Chizeck, The physiome projects and multiscale modeling, IEEE Signal Processing Magazine (2008).
[19] M. Cassman, A. Arkin, F. Katagiri, D. Lauffenburger, F.J. Doyle, C.L. Stokes, Barriers to progress in systems biology, Nature (2005).
[20] H. Schmidt, M. Jirstrand, Systems biology toolbox for matlab: a computational platform for research in systems biology, Bioinformatics Advance Access (2005) 22.
[21] R.M. Berne, M.N. Levy, Principles of Physiology, Mosby, 2000.
[22] C. Rosse, L.G. Shapiro, J.F. Brinkley, The digital anatomist foundational model: principles for defining and structuring its concept domain, in: Proceedings, American Medical Informatics Association Fall Symposium, 1998, pp. 820–824.
[23] C. Rosse, J.L.V. Mejino, B.R. Modayur, R.M. Jakobovits, K.P. Hinshaw, J.F. Brinkley, Motivation and organizational principles for anatomical knowledge representation: the digital anatomist symbolic knowledge base, Journal of the American Medical Informatics Association (1998).
[24] D.L. Cook, J.L.V. Mejino, C. Rosse, Evolution of a foundational model of physiology: symbolic representation for functional bioinformatics, in: Proceedings, MedInfo, 2004, pp. 336–340.
[25] D.L. Cook, J.L.V. Mejino, M.L. Neal, J.H. Gennari, Bridging biological ontologies and biosimulation: the ontology of physics for biology, in: In Proceedings, American Medical Informatics Association Fall Symposium, 2008, pp. 136–140.
[26] P.J. Hunter, T.K. Borg, Integration from proteins to organs: the physiome project., Nature Reviews, Molecular Cell Biology (2003).
[27] Cardiopulmonary mechanics with gas exchange. Available from: http://www.physiome.org/jsim/models/webmodel/NSR/CardiopulmonMechGasBloodExch/, 2011.
[28] A.I. Hern andez, V. Le Rolle, A. Defontaine, G. Carrault, A multiformalism and multiresolution modelling environment: application to the cardiovascular system and its regulation, Physical and Engineering Sciences 2009 (1908) 4923–4940.
[29] A. Guyton, T. Coleman, H. Granger, Circulation: overall regulation, Annual Review of Physiology (1972).
[30] PLplot. Available from: http://plplot.sourceforge.net/, 2011.
[31] D. Noble, The rise of computational biology, Nature Reviews Molecular Cell Biology (2002).

Please cite this article in press as: E.Z. Erson, M.C. Çavuşoğlu, Design of a framework for modeling, integration and simulation of physiological models, Comput. Methods Programs Biomed. (2012), doi:10.1016/j.cmpb.2011.11.010