35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

A versatile platform for multilevel modeling of physiological systems: Template/instance framework for large-scale modeling and simulation

Yoshiyuki Asai, Takeshi Abe, Hideki Oka, Masao Okita, Tomohiro Okuyama,
Ken-ichi Hagihara, Samik Ghosh, Yukiko Matsuoka, Yoshihisa Kurachi, Hrioaki Kitano

Abstract— **Building multilevel models of physiological systems is a significant and effective method for integrating a huge amount of bio-physiological data and knowledge obtained by earlier experiments and simulations. Since such models tend to be large in size and complicated in structure, appropriate software frameworks for supporting modeling activities are required. A software platform, PhysioDesigner, has been developed, which supports the process of creating multilevel models. Models developed on PhysioDesigner are established in an XML format called PHML. Every physiological entity in a model is represented as a module, and hence a model constitutes an aggregation of modules. When the number of entities of which the model is comprised is large, it is difficult to manage the entities manually, and some semiautomatic assistive functions are necessary. In this article, which focuses particularly on recently developed features of the platform for building large-scale models utilizing a template/instance framework and morphological information, the PhysioDesigner platform is introduced.**

I. Introduction

In recent biological and physiological research, computable mathematical models have become increasingly important for integrating the huge amount of knowledge and data obtained from experiments and simulations, and for applying simulation results to medicine[1]. There is an apparent trend for models to be increasingly large in size, as well as increasingly complicated and detailed in structure. It is almost impossible to build such models without collaboration among 'wet' and 'dry' research groups. To promote effective collaborations to build large-scale models, it is also important to consolidate fundamental tools to support such activities.

Model sharing and model reuse, which are crucial for the above-mentioned multidisciplinary collaborations, must be encouraged by using such tools. There have been several pioneering efforts to develop technologies in that direction, such as SBML[2], CellML[3] and PHML, among others.

This work was supported in part by MEXT G-COE program "in silico medicine" at Osaka University, and Grant-in-Aid for Scientific Research on Innovative Areas "Integrative Multi-level Systems Biology."

Y. Asai, T. Abe are with Okinawa Institute of Science and Technology Graduate University 1919-1 Onna-son, Okinawa, Japan. (email: [yoshiyuki.asai, takeshi.abe]@oist.jp). H. Oka is with Neuroinformatics Japan Center. RIKEN Brain Science Institute. 2-1 Wako, Saitama, Japan. (email: czoka@brain.riken.jp). M. Okita, T. Okuyama, K. Hagihara are with Graduate School of Information Science and Technology, Osaka University. 1-5 Suita, Osaka, Japan. (email: [hagihara, okita, t-okuyam]@ist.osaka-u.ac.jp). Y. Kurachi is with Graduate School of Medicine, Osaka University. 2-2 Suita, Osaka (email: ykurachi@pharma2.med.osaka-u.ac.jp). S. Gohsh, Y. Matsuoka, Hiroaki Kitano are with The Systems Biology Institute. 5-6-9 Minato, Tokyo, Japan. (email: [kitano, yukiko, samik]@sbi.jp)

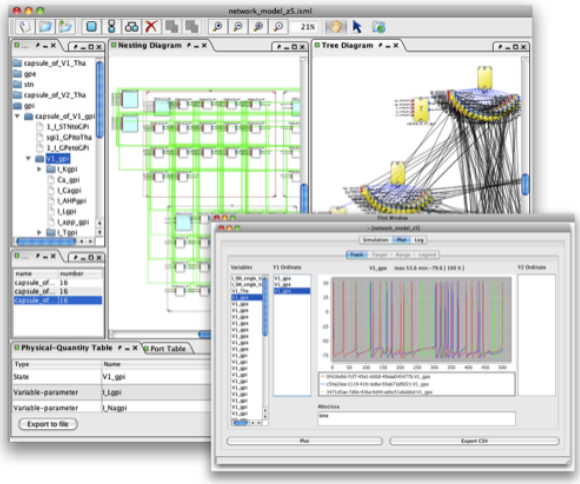

Fig. 1. Snapshot of PhysioDesigner and Flint.

These are XML-based descriptive language formats to describe the dynamics of biological and physiological systems. The main purpose of the development of these languages was to establish a common communication foundation to enhance the exchange of models among collaborators.

The development of the PHML language was initiated relatively late compared with the others listed above, in parallel with PhysioDesigner (Fig. 1) on which users can build mathematical models of multilevel physiological systems with a graphical user interface. Development of PhysioDesigner started in 2011, with the inheritance of all features from *insilico*IDE[4]–[6] available at physiome.jp[7]. Models built on PhysioDesigner are written in PHML, which is good at describing the hierarchical structure of physiological systems explicitly. PHML is a successor language of *insilico*ML[8].

Physiological structures are commonly composed of numerous similar sub-structures, such as muscle tissue as an aggregate of many muscle cells and neural networks consisting of many similar neurons. To model such physiological systems, PhysioDesigner employs a template/instance framework to deal with these repetitive structures systematically. Focusing on the template/instance framework, PhysioDesigner is introduced here as an extension of previous reports.

II. PhysioDesigner Overview

PhysioDesigner is software that supports the creation of computable models of physiological systems with mul-

978-1-4577-0216-7/13/$26.00 ©2013 IEEE

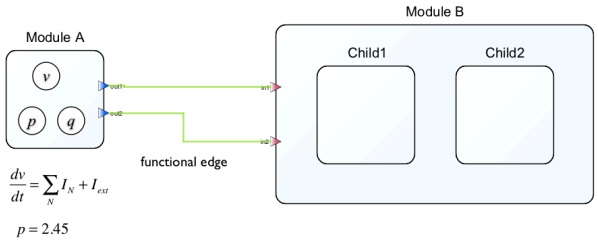

Fig. 2. Scheme of module, physical quantity and edge. Modules are fundamental elements to construct a model in PHML. Each module is quantitatively characterized by physical quantities. Relationships among modules are explicitly represented by edges. Modules can be defined in (or below) a module, forming a nesting (or tree) structure representing the hierarchical structure seen in physiological systems.

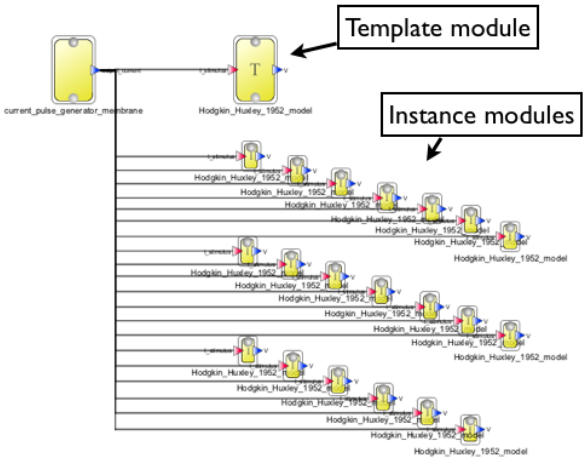

Fig. 3. An example of a template and instances in PhysioDesigner. When a functional edge is linked to the template from another module, the functional edge is considered to project to all instances.

tiple spatiotemporal levels. The software is available at http://physiodesigner.org. The current version, as of January 2013, is 1.0 beta3.

In PHML, each of the biological and physiological elements involved in a model is represented as a module (Fig. 2). Multiple modules can be defined at one level below a module; this lower level represents physiological entities that are more precise in spatial scale and more detailed in logical scale. By this nested representation of modules, hierarchical structures of physiological systems are explicitly expressed in a model. Structural and functional relationships among modules are represented by edges.

Each module is quantitatively characterized by several physical quantities, such as states defining the system's dynamics, and variable and static parameters. The dynamics, such as ordinary/partial differential equations, or functions of physical quantities are defined by mathematical equations using physical quantities. To define physical quantities in a module, it is often necessary to refer to the values of physical quantities defined in the other modules. The value of a physical quantity can be exported from a module through an out-port. Then, a functional edge linking the out-port and an in-port of a destination module carries numerical information. The value arriving at the in-port can be used to define a value of a physical quantity in the destination module.

The concept of making a kind of package of a physiological function has been introduced to PHML, which is called *capsulation*, in order to enhance the sharing and reuse of model, or even a part of the model. Capsulation is an operation that involves the encapsulation of an arbitrary number of modules acting together as a certain physiological function by a capsule module. All edge connections to (from) encapsulated modules from (to) the outside of the capsule must pass through the capsule module once in order to secure the independence of the encapsulated modules. Namely, the capsule module acts as an interface or gateway of all modules in the capsule. By this isolation of modules, it becomes easier to reuse the encapsulated modules in another part of the model or in other models.

Simulations of PHML models are conducted by the simulator Flint, which is being developed concurrently with PhysioDesigner. Flint was rebranded from *insilico*Sim[9], [10], and is also available at http://physiodesigner.org. Flint is described in detail elsewhere[6], [11].

III. Template/Instance Framework

A. Concept and Basic Usage

PhysioDesigner is equipped with a template and instance framework to model physiological systems involving many similar sub-structures. To use template/instance in a model, first, users need to define a template module. Any subset of an arbitrary number of modules under one capsule module can be defined as a template. Once users define a template model, it is possible to create multiple instances according to the template. Instances are not simply 'copies' of a template. All properties of instances are inherited from the template, but each instance itself does not have concrete definitions of physical quantities, such as states and parameters. Hence, once a user changes properties in the template, the changes are immediately applied to all instances. Users do not need to go through all instances to make the same changes for each of them. For example, let us assume that there is a template of a regular-spiking neuron model, and 100 instances forming a neural network. If some parameters are changed in the template so that the template behaves like a fast-spiking neuron, then the neural network composed of the instances is no longer a regular-spiking network, but instead represents a network of fast-spiking neurons.

Of course, sometimes we would like to assign a particular characteristic to several specific instances among many. In such a case, it is possible to modify values of static-parameter-type physical quantities and initial values of state-type physical quantities in individual instances. However, it is forbidden to modify, for example, equations of state-type physical quantities because, if the definition of the dynamics of an instance is different from the template, the instance can no longer be considered as an 'instance' of the template.

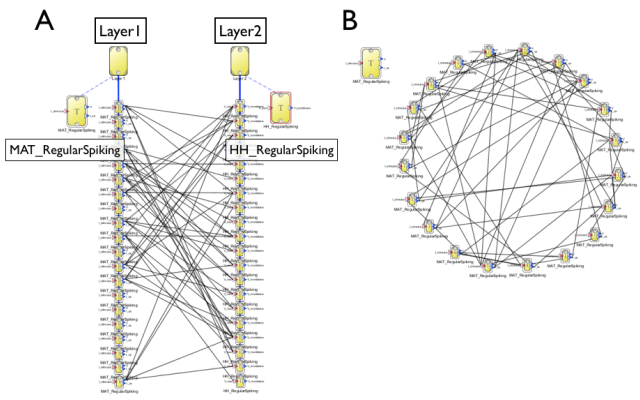

Fig. 4. Examples of semi-automatic edge connections among instance modules. A. Two-layered diverging/converging neural network model. Each layer includes one template ("MAT_RegularSpiking" and "HH_RegularSpiking") and 20 instances. B. Twenty instances of a single template connected to each other. These edge connections were generated semi-automatically.

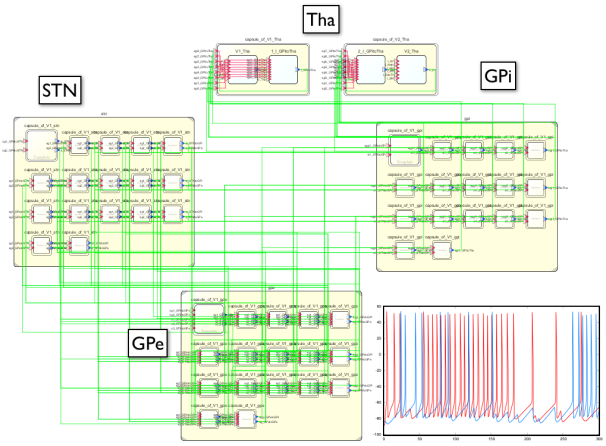

Fig. 5. A model of basal ganglia[12]. The lower right panel shows an example of a simulation result computed by Flint. Symbols are as follows. STN: subthalamic nucleus, GPi: globus pallidus internus, GPe: globus pallidus externus, Tha: thalamus.

B. Semi-automatic Ways to Integrate Template/Instance

Since all instances follow the configuration of their template, functional edges connected to the template are considered to be connected to all instances as well. This is convenient when all instances need to receive some information from other modules. Users do not need to link edges with one instance after another (Fig. 3).

In a case in which there are not many instances, it is possible to link edges to/from instances by manual operation, as users can do for typical modules. However, if there are a lot of instances, such as millions or more, it is practically impossible to do this. To overcome this difficulty, PhysioDesigner is equipped with a method to link edges among instances semi-automatically.

One way is to create edges from randomly selected instances in a set of instances of a template to instances selected randomly from another set of instances of the other template. Let us assume the creation of a layered neural network model as an example. In Fig. 4A, there are two modules named "Layer 1" and "Layer 2", which include a different template. Each of the layers includes 20 instances of their respective template.

To link edges among instances from one layer to another semi-automatically in PhysioDesigner, users need to define the number of instances from which edges start, and the number of instances to which one instance projects. In addition, edge details (e.g. which in-port and out-port should be used) need to be defined. Figure 4A shows a model resulting from the random selection of 10 instances from Layer 1 and the definition of each instance in Layer 1 as having edges projecting to 5 instances randomly selected from Layer 2. In this case, overlap of projections of edges at the target side is allowed. Hence, some instances in Layer 2 receive multiple edges. If projection overlap is forbidden and all instances in Layer 1 are involved and each of them projects only one edge to Layer 2, then instances in the two layers are linked in a one-to-one manner.

Similarly it is also possible to link edges among instances of a single template. Figure 4B shows an example model that includes only instances of one template. Using these functions, it is possible to create, for example, a neural network model including several nerve nuclei. Figure 5 shows one such model proposed by Rubin and Terman, 2004[12]. There are complicated connections within and between nuclei.

C. Object-based Modeling with Template/Instance

A template/instance framework can be a powerful method to create, for example, a whole-organ model, because usually a certain number of cells of the same type are involved in the formation of organs, which can be modeled by instances. For such modeling, we also need to consider a morphology. PhysioDesigner can utilize morphometric data with voxel-based volume representation and template/instances for the creation of such a model. A voxel-based volume model is an aggregate of volume elements, representing values on a regular grid in three-dimensional space. By replacing each voxel by an instance and defining how an instance links to adjacent instances (in 6 or 26 directions), a computable model is created on the basis of a voxel-based volume object. For now, all voxels in one object are replaced by instances of a single template, that is, the resultant model is homogeneous in terms of the property of the volume element. However, it is possible to define not only direct connections between two adjacent instances but also indirect connections, that is, an arbitrary number of instances of other templates can be placed between two adjacent volume elements.

Figure 6 shows the workflow for creating an object-based model with template/instances. In this example, the object represents a cardiac chamber. The example starts from an object with a three-dimensional volume mesh, which is another format that is also frequently used to represent morphological data. At first, PhysioDesigner converts it into a voxel-based volume representation. Then, every voxel is replaced by an instance of a template representing the Luo-Rudy cardiac cell model[13]. In this case, two adjacent

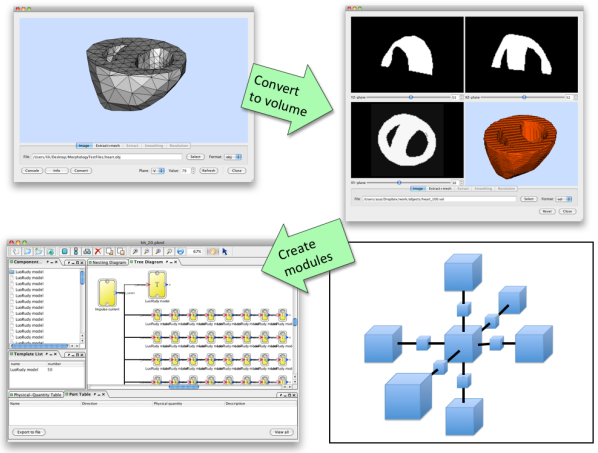

Fig. 6. An example of object-based modeling with template/instances. A three-dimensional mesh object is converted once into a voxel-based volume representation. Then, every voxel in the three-dimensional object is replaced by an instance. To define connections between two adjacent voxels, it is possible to insert instances of other templates in-between, as shown in the lower right panel.

cardiac cell instances are not directly connected by edges. A gap junction instance is sandwiched between them, as shown in the lower right panel in Fig. 6.

IV. Conclusion

A versatile platform, PhysioDesigner, which provides an environment for multilevel modeling of physiological systems, has been demonstrated focusing on one of its features, called the template/instance framework. With functions to link edges semi-automatically among instances, the template/instance framework can facilitate large-scale modeling. Moreover, the use of this platform enables the utilization of morphometric information as a skeletal structure to create a computable model with instances. PhysioDesigner is also capable to integrate SBML models which describe subcellular phenomena. Integration of SBML and morphology to PHML with template/instance framework could be a novel way to create a large-scale multilevel models. A function to integrate SBML is described in detail elsewhere.

One of the problems envisioned to arise when models become huge is a shortage of computing resources for simulations. To solve this problem, a development for a simulator Flint, which fully supports the template/instance framework, is ongoing in order for it to be executable on K Computer, the next-generation supercomputer. The execution of Flint on K Computer will provide a solution for the problems associated with the scaling-up of models, in terms of computational power.

We provided an example of how to create a ventricular model with morphometric data and template/instance in Fig. 6. A limitation of the current function to create a model with morphology and template/instances is that all voxels in the voxel-based volume object are replaced by instances of a single template. Practically, ventricular wall, for example, should be modeled as a three-layered structure. If users prepare these three layers separately as different three-dimensional objects, creation of a heterogeneous model can still be achieved using existing approaches. However, the procedure to make a model becomes more troublesome. Including dealing with this issue, further functions to utilize morphometric data for the creation of computable models on PhysioDesigner are under development.

Acknowledgment

The authors would like to thank Prof. T. Nomura and Mr. H. Shakuda, Osaka University, for implementing the model in Fig. 5 on PhysioDesigner.

References

[1] H. Kitano, "Computational systems biology." *Nature*, vol. 420, no. 6912, pp. 206–10, 11 2002.

[2] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, J. Wang, and S. Forum, "The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models." *Bioinformatics*, vol. 19, no. 4, pp. 524–31, 3 2003.

[3] C. M. Lloyd, M. D. B. Halstead, and P. F. Nielsen, "Cellml: its future, present and past." *Prog Biophys Mol Biol*, vol. 85, no. 2-3, pp. 433–50, 2004.

[4] Y. Suzuki, Y. Asai, T. Kawazu, M. Nakanishi, Y. Yaniguchi, E. Heien, K. Hagihara, Y. Kurachi, and T. Nomura, "A platform for in silico modeling of physiological systems ii. cellml compatibility and other extended capabilities." *Conf Proc IEEE Eng Med Biol Soc*, vol. 2008, pp. 573–6, 2008.

[5] Y. Suzuki, Y. Asai, H. Oka, E. Heien, T. Urai, T. Okamoto, Y. Yumikura, K. Tominaga, Y. Kido, M. Nakanishi, K. Hagihara, Y. Kurachi, and T. Nomura, "A platform for in silico modeling of physiological systems iii." *Conf Proc IEEE Eng Med Biol Soc*, vol. 2009, pp. 2803–6, 2009.

[6] Y. Asai, H. Oka, T. Abe, M. Okita, K. Hagihara, T. Nomura, and H. Kitano, "An open platform toward large-scale multilevel modeling and simulation of physiological systems," *Conf Proc 11th IEEE/IPSJ International Symposium on Applications and the Internet, SAINT 2011*, pp. 250–255, 2011.

[7] T. Nomura, "Toward integration of biological and physiological functions at multiple levels," *Frontiers in Systems Physiology*, vol. 1, no. 164, 2010.

[8] Y. Asai, Y. Suzuki, Y. Kido, H. Oka, E. Heien, M. Nakanishi, T. Urai, K. Hagihara, Y. Kurachi, and T. Nomura, "Specifications of insilicoml 1.0: a multilevel biophysical model description language." *J Physiol Sci*, vol. 58, no. 7, pp. 447–58, 12 2008.

[9] E. M. Heien, Y. Asai, T. Nomura, and K. Hagihara, "Optimization techniques for parallel biophysical simulations generated by *insilico*IDE," *IPSJ Online Transactions*, vol. 2, pp. 149–161, 2009.

[10] E. M. Heien, M. Okita, Y. Asai, T. Nomura, and K. Hagihara., "*insilico*sim: an extendable engine for parallel heterogeneous biophysical simulations." *In Proceedings 3rd International Conference. Simulation Tools and Techniques (SIMUTools '10)*, pp. 78:1–78:10, 2010.

[11] Y. Asai, T. Abe, M. Okita, T. Okuyama, N. Yoshioka, S. Yokoyama, M. Nagaku, K. Hagihara, and H. Kitano, "Multilevel modeling of physiological systems and simulation platform: Physiodesigner, flint and flint k3 service," *Conf Proc 12th IEEE/IPSJ International Symposium on Applications and the Internet, SAINT 2012*, pp. 215–219, 2012.

[12] J. E. Rubin and D. Terman, "High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model." *J Comput Neurosci*, vol. 16, no. 3, pp. 211–35, 2004.

[13] C. H. Luo and Y. Rudy, "A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction." *Circ Res*, vol. 68, no. 6, pp. 1501–26, 6 1991.