IEICE Proceeding Series

# Multilevel modeling platform and its application for modeling in neuroscience

Yoshiyuki Asai, Hideki Oka, Alessandro E. P. Villa, Hiroaki Kitano

Vol. 1 pp. 296-299
Publication Date: 2014/03/17
Online ISSN: 2188-5079

2012 International Symposium on Nonlinear Theory and its Applications
NOLTA2012, Palma, Majorca, Spain, October 22-26, 2012

# Multilevel modeling platform and its application for modeling in neuroscience

Yoshiyuki Asai[1], Hideki Oka[2], Alessandro E. P. Villa[3] and Hiroaki Kitano[1,4]

1. Open Biology Unit, Okinawa Institute of Science and Technology Graduate University,
1919-1 Tancha, Onna-son, Okinawa, Japan
2. Neuroheuristic Research Group, HEC-ISI, University of Lausanne,
CH-1015, Lausanne, Switzerland
3. RIKEN, Brain Science Institute, Neuroinformatics Japan Center,
2-1, Hirosawa, Wako, Saitama, Japan
4. The Systems Biology Institute, 5-6-9, Shirokanedai, Minato, Tokyo, Japan
Email: yoshiyuki.asai@oist.jp, czoka@brain.riken.jp, avilla@neuroheuristic.org, kitano@sbi.jp

**Abstract**—Recently models of physiological systems including single neuron or neural network models are getting larger in size and more complicated in accuracy. In addition, there are a wide range of models. To enhance sharing and reusing a whole or a part of a model is an effective way to promote the computational neuroscience. A platform for model sharing, and for building multilevel models of physiological systems have been developed and presented in this article with a use case of a neural network modeling. On the platform, a model is represented as an aggregate of modules. We applied our new platform to implement the neural network model, and demonstrated the platform functions, especially focusing on a feature to create large scale models.

## 1. Introduction

In past decades, based on the huge amount of data provided by the reductionism science, modeling-based science in systems biology[1] and integrated physiology has been progressing rapidly. Models are getting bigger and bigger in the size, and more and more complicated and detailed in the structure. Neuroscience is not exception. Simulation studies of brain science tends to use bigger neural networks and more detailed and complicated single neuron models. It is almost impossible to build such models without inter-research-group collaborations, not only between 'wet' and 'dry' research groups, but also 'dry' and 'dry' research groups. For promoting effective collaboration to build large-scale models, it is very important to develop tools to support such activities.

There are a couple of efforts to develop such technology such as SBML[2], CellML[3] and PHML. These are XML based descriptive language formats to describe dynamics of physiological systems. PHML is a rather recently developed language with an application PhysioDesigner on which users can build a mathematical model of multilevel physiological systems with graphical user interface. On PhysioDesigner and PHML, every physiological entity included in a model is represented as a module. Hence a model is an aggregate of modules. It is easy to share and reuse a whole or parts of model because of the modular expression.

In a series of our previous work[4, 5, 6], a multiple layers neural network characterized by diverging/converging projections between successive layers activated by an external spatio-temporal input pattern in presence of stochastic background activities was considered. We have reported the properties and performance of spike information transmission in the network depending on neuron model type, inputed information type and background activity level. The models were rather simple and can be more detailed and bigger in size for further investigation. Taking the network model as an example, the implementation of the network model on PhysioDesigner is demonstrated in this article.

Simulations of PHML models can be conducted by a simulator Flint which is developed along with PhysioDesigner. It is also available at `http://physiodesigner.org`. Details of Flint will be shown somewhere else.

## 2. Model building platform: PhysioDesigner

### 2.1. Overview

PhysioDesigner is a software that supports spatiotemporal multiple level modeling based on modules to create computable models of physiological systems. The software is opened and available at `http://physiodesigner.org`. PhysioDesigner was rebranded from *insilico*IDE (`http://physiome.jp`) in 2011[7, 8]. Models build on PhysioDesigner are written in PHML format, which is an XML based specification to describe hierarchy of systems in comprehensive biological models explicitly, which is a successor language of ISML (`http://physiome.jp`)[9].

In PHML, each of biological and physiological elements involved in a model is called a module. Structural and functional relationships among modules are defined by edges. A group of several modules can be defined as a module at one level higher concept. By this recursive definition of the modules, a hierarchical structure of physiological systems is expressed in a model.

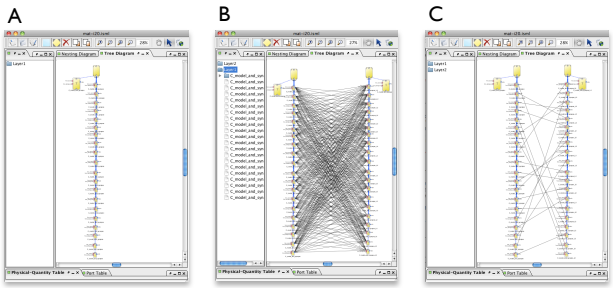


Figure 1: A. 20 instances of a template were created, and aligned in line. The template and instances are located under a module named "Layer1". B. The module "Layer1" was copied to create the second layer. Then instances in Layer1 and Layer2 were linked by functional edges with 1:15 correspondence, i.e. every instance in Layer1 projects to randomly selected 15 instances in Layer2. Hence some of instances in Layer2 receive inputs from multiple instances in Layer1. C. As a special case, it is also possible to link instances with one to one correspondence (without any overlap in Layer2) by automated method.

Each module is quantitatively characterized by several physical-quantities, such as, states defining the system's dynamics, and variable and static parameters. Definition of the dynamics such as ordinary/partial differential equations, or functions of physical-quantities are explicitly described by mathematical equations using physical-quantities. A definition of a functional relationship between any two modules are represented by functional edges linking an out-port of a module to an in-port of other module. Each of functional edges carries a numerical information of a physical-quantity associated to the out-port in the sender module. The module can utilize a value received via an in-port in definitions of physical-quantities.

A concept to make a kind of package of a physiological function has been introduced to PHML, which is called *capsulation*, in order to enhance the model sharing even a part of a model. By the capsulation, several modules acting together as a certain physiological function are encapsulated by a capsule module. All connections to (from) the encapsulated modules from (to) outside of the capsule must go through the capsule to secure the independence of the encapsulated modules. By this isolation of modules, it becomes easier to reuse the encapsulated modules in other part of the model or in other models.

### 2.2. Template and instance framework

One of distinguished features of PhysioDesigner is a template/instance framework to assist to create large size models. For example, to create a neural network model consisting of the same type of neurons, instead of multiplying modules representing a neuron, it is possible to define a template of a neuron model, and create a lot of instances

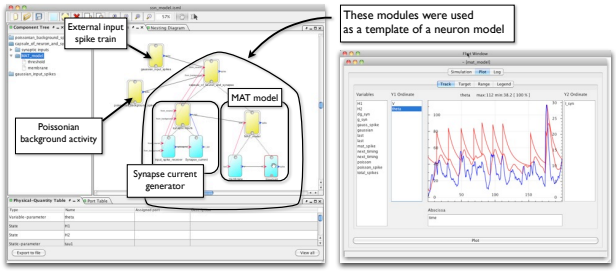


Figure 2: Left panel shows an implementation of a single MAT model, synaptic current generator, and input spike train generators on PhysioDesigner. Right panel shows a simulation result of the model on a simulator Flint. Red and blue curves show the adaptive threshold and membrane potential, respectively.

according to the template. Then if parameters in the template were changed, the changes have an effect to the all instances. Additionally, if some instances need to be given a specific characteristics, initial values of dynamic parameters and constants can be modified individually.

The number of instances can be big. And often we need to make links between instances of a template and instances of the other template. But because of the number of instances, it is not practical to define edges one by one. An automated method to create links among instances is required, and has been implemented on PhysioDesigner (Fig. 1).

## 3. Implementation of a neural network model

### 3.1. Single neuron model

In this article, we adopted a multiple-timescale adaptive threshold (MAT) neuron model[10] to simulate the dynamics of *regular spiking neurons*, whose membrane potential dynamics follows a non-resetting leaky integrator,

$$\tau_m \frac{dV}{dt} = -V(t) + R\ A\ I_{ext}(t)\,, \qquad (1)$$

where $\tau_m, V, R$ and $A$ are the membrane time constant, membrane potential, membrane resistance, and scaling factor, respectively. A spike is generated when $V(t) \geq \theta(t)$, $\theta(t) = \omega + H_1(t) + H_2(t)$ , $\frac{dH_1}{dt} = -H_1/\tau_1$ , $\frac{dH_2}{dt} = -H_2/\tau_2$ , where $\omega$ is the resting value. $H_1$ and $H_2$ are components of the fast and slow threshold dynamics (characterized by decaying time constants $\tau_1$ and $\tau_2$, respectively) which have a discrete jump when $V(t) \geq \theta(t)$, $H_1 = H_1 + \alpha_1$ , $H_2 = H_2 + \alpha_2$ .Parameters were set to values $\tau_m = 5$ ms, $R = 50$ MΩ, $A = 0.106$, $\omega = 19$ mV, $\tau_1 = 10$ ms, $\tau_2 = 200$ ms, $\alpha_1 = 37$ mV, $\alpha_2 = 2$ mV. The implementation of this neuron model and simulation result are shown in Fig. 2. Numerical integration was done by the Euler method with 0.002 *ms* time steps for all cases.

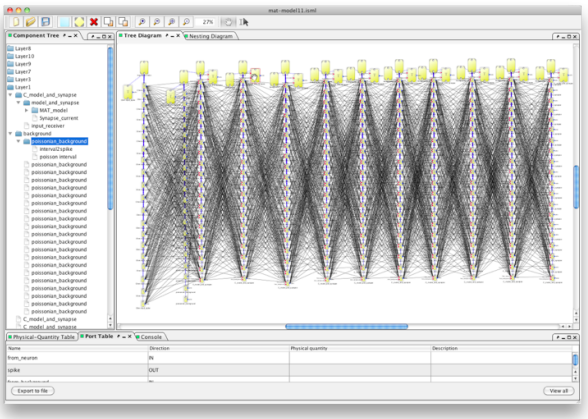

Figure 3: Left panel shows an implementation of a single MAT model, synaptic current generator, and input spike train generators on PhysioDesigner. Right panel shows a simulation result of the model on a simulator Flint. Red and blue curves show the adaptive threshold and membrane potential, respectively.

### 3.2. Neural network model

We consider a diverging/converging neural network composed of ten layers. Each layer includes 20 neurons. All neurons in a network are identical and are MAT models and receive background activity represented by an independent Poissonian spike train with a mean firing rate of 425 $spikes/s$. Each neuron of Layer 1 receives an external input represented by 15 spike trains derived from a dynamical systems described below. From 2nd to downward layers each neuron receives afferences from 15 neurons randomly selected among those of the immediately upstream layer. All connections were hardwired, and no synaptic plasticity was taken into account.

A synaptic current $I$ was defined as $I = -A \sum_k g_{syn}(t - t_k)$ ,where $A$ is a constant intensity ($A = 0.9$ for all simulations), and $t_k$ represents time when the $k$-th spike arrives to the neuron. $g_{syn}$ is the post synaptic conductance represented by $g_{syn}(t) = \frac{C_0}{\tilde{\tau}} e^{-t/\tilde{\tau}}$ ,where $\tilde{\tau}$ is a time constant given by 2 $ms$. $C_0$ is a coefficient used to normalize the maximum amplitude of $g_{syn}(t)$ to 1. It is necessary for a post-synaptic neuron to integrate several arriving synaptic currents for a spike generation.

Inter-spike-intervals of input spike train given to the first layer of the network were generated based on Chen's equations, which exhibits chaotic dynamics with certain parameter values.

$$\frac{dx}{dt} = a(y - x)\ ,\ \frac{dy}{dt} = (c - a)x - xz + cy\ ,\ \frac{dz}{dt} = xy - bz$$

where $a = 35.0, b = 3.0, c = 28.0$, and $x(0) = y(0) = z(0) = 3.0$ for initial conditions. We considered a Poincaré map where the Poincaré section was defined by $\frac{dx}{dt} = 0$, and

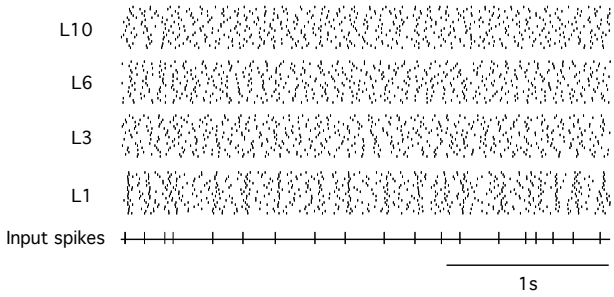


Figure 4: A raster plot of neural activity at layer 1, 3, 6 and 10, labeled by L1, L3, L6 and L10, respectively. The bottom shows the input spike train fed to the first layer of the network. In each layer, there are twenty rows corresponding to twenty neurons.

the sequence of $z(t)$ on the section was traced. $z(t)$ was used as the inter-spike-interval of the input spike train.

### 3.3. Implementation on PhysioDesigner

In a single layer, there are 20 neurons and 20 background activity generators which were modeled as instances. Every neuron instance has to receive an independent background activity, hence they were connected by one-to-one. Neuron instances in a layer to those in a next layer were linked by one to fifteen correspondence. Whole network model is shown in Fig. 3.

The simulation was done by Flint. The raster plot of spikes of neurons in Layers 1, 3, 6 and 10 were displayed in Fig. 4.

## 4. Discussion

Taking a diverging/converging neural network as an example, we demonstrated one of functions of PhysioDesigner to support users to build large scale models which can be applicable to neuroscience as well as other physiological field. There are a lot of other tools to support modeling in neuroscience, such as Neuron (`http://www.neuron.yale.edu/neuron/`), Genesis (`http://www.genesis-sim.org/GENESIS/`) and more. Besides the function presented here, PhysioDesigner equips other functions to support multilevel modeling. In total, PhysioDesigner can provide complementary functions to those existing applications. And moreover if users want to integrate other physiological systems such as muscles, circulation, and so on, PhysioDesigner can still support such modeling comprehensively.

One unique function of PhysioDesigner is to enable users to build SBML-PHML hybrid models. SBML (the systems biology markup language) [2] is an XML format for computer models of biological processes, such as metabolism, cell signaling, and more. PHML is good at

representing a functional network and hierarchical structure using its modular representation. Combining SBML and PHML, it is possible to extend the capability of both languages complementarily to construct models including multiple levels of physiological phenomena from biochemical processes to electrophysiological dynamics.

Integration of morphometric data with mathematical models is another function of PhysioDesigner worth to be mentioned. Morphology information can be used to define, for example, a domain in which partial differential equations (PDEs) such as Poisson equation for the electric field. These techniques can be used in calculation of EEG, for example, considering explicitly modeled hypothetical neural activity. Simple models integrating morphology and PDEs can be dealt with on PhysioDesigner already, and details will be presented somewhere else.

To providing people a chance to share or reuse analysis tools is also important. There is an effort called OpenAdap.net [11] to develop a web-based community in which people can share tools for data analyses. Recently another new effort called Garuda alliance [12] (`http://www.garuda-alliance.org`) has initiated. Garuda platform aims at providing seamless linkages among existing tools so that users can work on multiple tools as like on one platform. PhysioDesigner is an alliance member of Garuda project.

We hope PhysioDesigner and relevant technologies can be contributory to the development of physiology and neuroscience.

**Acknowledgment**

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".

## References

[1] Kitano, H.: Computational systems biology. Nature **420**(6912) (11 2002) 206–10

[2] Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., Cuellar, A.A., Dronov, S., Gilles, E.D., Ginkel, M., Gor, V., Goryanin, I.I., Hedley, W.J., Hodgman, T.C., Hofmeyr, J.H.H., Hunter, P.J., Juty, N.S., Kasberger, J.L., Kremling, A., Kummer, U., Le Novère, N., Loew, L.M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E.D., Nakayama, Y., Nelson, M.R., Nielsen, P.F., Sakurada, T., Schaff, J.C., Shapiro, B.E., Shimizu, T.S., Spence, H.D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J., Forum, S.: The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics **19**(4) (3 2003) 524–31

[3] Lloyd, C.M., Halstead, M.D.B., Nielsen, P.F.: Cellml: its future, present and past. Prog Biophys Mol Biol **85**(2-3) (2004) 433–50

[4] Asai, Y., Yokoi, T., Villa, A.E.P.: Detection of a dynamical system attractor from spike train analysis. Lecture Notes in Computer Sciences **4131** (2006) 623–631

[5] Asai, Y., Guha, A., Villa, A.E.P.: Deterministic neural dynamics transmitted through neural networks. Neural Networks **21** (2008) 799–809

[6] Asai, Y., Villa, A.E.P.: Transmission of distributed deterministic temporal information through a diverging/converging three-layers neural network. Lecture Notes in Computer Sciences **6532** (2010) 145–154

[7] Nomura, T.: Toward integration of biological and physiological functions at multiple levels. Frontiers in Systems Physiology **1**(164) (2010)

[8] Asai, Y., Oka, H., Abe, T., Okita, M., Hagihara, K., Nomura, T., Kitano, H.: An open platform toward large-scale multilevel modeling and simulation of physiological systems. 11th Annual International Symposium on Applications and the Internet, SAINT 2011, Conference Proceedings (2011) 250–255

[9] Asai, Y., Suzuki, Y., Kido, Y., Oka, H., Heien, E., Nakanishi, M., Urai, T., Hagihara, K., Kurachi, Y., Nomura, T.: Specifications of insilicoml 1.0: a multilevel biophysical model description language. J Physiol Sci **58**(7) (12 2008) 447–58

[10] Kobayashi, R., Tsubo, Y., Shinomoto, S.: Made-to-order spiking neuron model equipped with a multi-timescale adaptive threshold. Front Comput Neurosci **3** (2009) doi:10.3389/neuro.10.009.2009

[11] Villa, A., Iglesias, J.: Openadap.net: Evolvable information processing environment. Lecture Notes in Artificial Intelligence **4578** (2007) 227–236

[12] Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y.Y., Kitano, H.: Software for systems biology: from tools to integrated platforms. Nat Rev Genet **12**(12) (12 2011) 821–832