

# Executable Symbolic Models of Neural Processes

Sriram M Iyengar<sup>1</sup>, Carolyn Talcott<sup>2</sup>, Riccardo Mozzachiodi<sup>3</sup>, Enrico Cataldo<sup>4</sup>, Douglas A Baxter<sup>3§</sup>

<sup>1</sup>School of Health Information Sciences, The University of Texas Health Science Center at Houston, USA

<sup>2</sup>Computer Science Laboratory, SRI Intl., Menlo Park, CA, USA

<sup>3</sup>Department of Neurobiology and Anatomy, The University of Texas Health Science Center at Houston, USA

<sup>4</sup>Department of Biology, General Physiology Unit, Via San Zeno 31, University of Pisa, Italy

§Corresponding author

Email addresses:

SMI: [msriram@uth.tmc.edu](mailto:msriram@uth.tmc.edu)

CT: [clt@cs.sri.com](mailto:clt@cs.sri.com)

RM: [riccardo.mozzachiodi@uth.tmc.edu](mailto:riccardo.mozzachiodi@uth.tmc.edu)

EC: [ecataldo@biologia.unipi.it](mailto:ecataldo@biologia.unipi.it)

DAB: [douglas.baxter@uth.tmc.edu](mailto:douglas.baxter@uth.tmc.edu)

## Abstract

### Background

Neuroscience is experiencing explosive growth in detailed high-quality experimental information on neural processes underlying learning, memory and behavior. There is a need for computational models that can manage this outpouring of information, derive knowledge from information, and to generate novel, testable hypotheses. Many current models of neural processes utilize a framework of differential equations. These models tend to exhibit high sensitivity to system parameters requiring accurate measurements of these parameters. Such data are frequently unavailable, leading to difficult solution stability, robustness, and validity problems. Further, the models do not scale easily since they rapidly become intractable as the number of cells incorporated increases. The situation is analogous to that in the biochemical pathway modeling. There, a complementary approach based on a computing formalism called *Pathway Logic*, implemented by the high-level rewriting-logic language Maude. This approach is being successfully used to model and simulate biochemical signaling pathways.

### Results

We describe an application of *Pathway Logic*, using the rewriting-logic specification system Maude, to model a neural circuit involved in feeding behavior of a marine mollusk. This approach has potential advantages of scalability and robustness with regard to system parameters, allows us to easily represent properties of neuron as well as different types of inter-cellular connections and to reproduce key features of known behavior. Also, the model enables easy modification of the properties characterizing a neuron and observation of the effects on its behavior, singly and as a part of more complex circuits. The ability to carry out such *in silico* experiments is one of the benefits of this proposed approach.

## Conclusions

This approach yields expressive models capable of simulating known neural circuit behaviors and performing *in silico* experiments including knock-outs, ‘what-if’s and others. More complete models including representations of additional neuron properties, might be capable of providing new understanding of more complex behavior and testable hypotheses regarding the effects of modifying circuit elements.

## Background

The last several years have witnessed exciting and explosive growth in the amount of detailed high-quality experimental information on neural processes underlying behavior. Concurrently, computational neuroscience has also experienced a surge of interest and activity in the formulation of models of increasing complexity. These twin developments present opportunities as well as challenges to neuroinformatics. Among current models of neural processes, a lot of them utilize a framework of differential equations such as the Hodgkin-Huxley (H-H) equations [1], integrate-and-fire methods and artificial neural networks [2]. These models and in particular the H-H models, are affected by high sensitivity to system parameters, which are frequently unavailable, implying difficult solution stability, robustness, and validity problems. In addition, these models do not scale easily, rapidly become intractable as the number of cells incorporated increases. The situation is analogous to that in the modeling of biochemical pathways. There, a complementary approach based on a computing formalism called *Pathway Logic* (PL) [3-7], implemented by the high-level rewriting-logic language Maude [8, 9], is being successfully used to model and simulate biochemical signaling pathways. By representing knowledge at an abstract, symbolic level this approach has enabled development of models capable of performing sophisticated queries about signaling pathways, while not being overly constrained by lack of low-level details.

The aim of the present research is to apply such approaches to neural processes underlying behavioral plasticity. By behavioral plasticity we mean the changes in neuronal properties that result as a consequence of learning and memory. This methodology is not intended to replace the existing techniques based on differential equations, but to complement these by representing knowledge at a different level of abstraction. PL models are concerned with, among other things, the overall logic of the neuronal signals and the variations in qualitative features of neural components across the entire circuit. Further, this abstraction can be easily extended to relate the behavior of the circuit and circuit components to visible high-level behavior. The models are intended to be qualitative and executable, i.e., starting from a given system state, complex concurrent state changes can be deduced based upon existing knowledge. This capability enables the model to answer “what if questions”, perform *in silico* neural circuit modification experiments, and symbolically execute theories over time, formally analyzing all states reachable from an arbitrary initial state. In this paper, we present the first steps in the development of PL models of identified neural circuits, important in learning and memory, in the marine mollusk *Aplysia*.

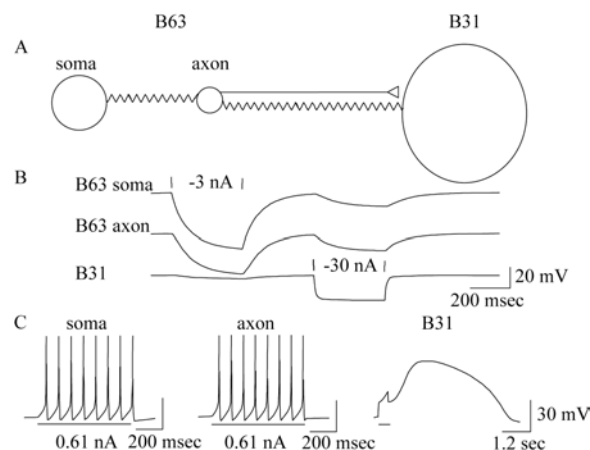
## Neurons and Neural Circuits

Neurons are highly specialized eukaryotic biological cells capable of communicating with each other by means of electrical (also called *gap junctions*). and chemical (inhibitory and excitatory) signaling. In general, all possess a cell body, called the *soma*, from which emerge several highly branched structures called *dendrites* as well as a single long process called the *axon*, which ends in several branches, the *synaptic terminals*. The synaptic terminals of the transmitting (*presynaptic*) neuron communicate chemical signals to the dendrites of nearby

receiving (*postsynaptic*) neurons by releasing specialized molecules, the *neurotransmitters*, released by the presynaptic neuron after a sufficient transmembrane voltage depolarization reaches the synaptic terminal. The buildup of membrane depolarization at the synapse occurs after a nerve impulse, or *action potential*, travels down the cell body and the axon in a series of *depolarizations* and *repolarizations* caused by bidirectional transfer of sodium and potassium ions between the cytoplasm of the neuron and the extracellular space. Further details can be found in [10].

The ability of the nervous system to generate behaviors arises from the organization of neurons into networks whose functional capabilities emerge from interactions among the intrinsic neuronal biophysical properties, the pattern of the inter-neural synaptic connections and the physiological properties of these synaptic connections. By virtue of their relatively simple nervous systems, often with large identifiable neurons that are amenable to detailed study, invertebrates are frequent candidates for cellular analyses of neural circuits and their relationship to behavior [11,12]. One useful animal model system is the marine mollusk *Aplysia*. Several of its behaviors, including feeding [13,14], can be modified by learning and many features of the neural circuits that underlie these behaviors are known and several neural correlates of learning have been identified.

Previously, Susswein et al. [15] developed a model that contained two neurons, B31/B32 and B63 (Fig. 1) and simulations were performed with the neurosimulator SNNAP (Simulator for Neural Networks and Action Potentials) [16].



**Figure 1 - Features of the two-cell model**

A: B63 was modeled with separate soma and axon compartments. The B63 axon was electrically coupled to B31. In addition, the B63 axon excited B31 via a moderately facilitating chemical synapse. B: coupling conductance between B63 and B31. C: response properties of B63 soma and axon and of B31, in the absence of connections between them.

Several configurations of the model were examined. Simulations indicated that a configuration in which electrical and fast chemical coupling occurred in combination with a plateau-like potential gave rise to a circuit that was robust to changes in parameter values and stochastic fluctuations, that closely mimicked empirical observations.

### PL Models of Biochemical Pathways

The situation described above is analogous to that currently in the fields of biological pathways including signal transduction and metabolic pathways. Quantitative models in these fields, such as the Michaelis-Menten [17] equations are similarly affected by sensitivity of

parameters, hampered by lack of detailed knowledge of these parameters and intractability when models become large in the sense of including significant numbers of biomolecular reactions. These issues have prompted biologists to collaborate with computer scientists to develop new knowledge representation and modeling techniques. In particular, PL, described further below, has been successfully applied in the modeling of signal transduction pathways [5].

From an epistemological point of view, PL represents knowledge at a higher degree of abstraction. It is symbolic, i.e., it represents the different entities of interest such as cells, proteins, functional domains, cytoskeleton, and others, explicitly as symbols capable of being processed by software. A benefit of this representation is that the subsequent models are executable, enabling *in silico* simulations of complex, biologically meaningful experiments, such as, knock-outs, in which, for example, the effect of removing a particular biomolecule from a signal transduction pathway is investigated. PL is based on a branch of computer science called Formal Methods, which derive from the mathematical concept of formal systems and logics [18]. Formal methods enable *model checking*, a method for formally verifying hardware and software systems in which the system specifications are expressed as formulas in the temporal logic, and efficient symbolic algorithms are used to traverse the state space reachable from an initial state of a system model and check whether the specification holds or not. Extremely large state-spaces can be examined in a matter of minutes.

A highly desirable feature of any modeling methodology is scalability, which can be informally defined as the capability of the technique to include large numbers of the entities of interest and yet perform computations within a reasonable amount of time and with reasonable accuracy. In this respect, it has long been observed that modeling based on ordinary or partial differential equations typically become computationally intractable when the number of differential equations to be integrated simultaneously is in the high tens. Approximate solutions based on finite element methods are typically invoked to solve such equation sets. However, the limitations of such modeling methods become apparent when one notes that a single cubic centimeter of the human brain can contain 50 million neurons [10]. By contrast, qualitative approaches such the PL models can directly handle much larger systems. For example, PL networks of several thousand nodes and edges have been generated and analyzed in tens of minutes on standard laptop hardware, without substantial effort to optimize the processing (see <http://pl.csl.sri.com/>). The formal nature of the models means that abstraction interpretation techniques used to analyze complex software system can be applied to make query/analysis of even larger networks feasible. Biological systems have been modeled using computational formalisms such as Petri Nets [19], pi-calculus [20], Statecharts [21], interaction networks[22]. To the best of our knowledge none of these qualitative symbolic methods has been used to model neural signaling circuits so far.

## Results

### Pathway Logic Modeling in Neuroscience

To illustrate the proposed modeling approach a Maude model of a simplified version of the *Aplysia* B63-B31/32 two neuron system (Fig. 1) studied in [15] was developed. The purpose of developing this model was to demonstrate that Maude and PL are capable of representing and expressing neuro-physiological objects and the behavior of neural circuits. In this approach, each neuron is represented as an interactive object with attributes whose values correspond to neuron properties and state. For example the Maude specification

```
[B31 : Neuron | thresh: 4, decay: 5, dplevel: 0, ttl: 0,
      out: false, in: 0 ]
```

represents a B31-like neuron with depolarization threshold 4 (all numbers represent relative quantities in an abstract unit system) and time for decay from plateau to resting depolarization level equal to 5. The attributes `dplevel`, `ttl`, `out`, and `in` represent the current state of the neuron as the system evolves: `dplevel` is the depolarization level, initially 0; `ttl` is the remaining decay time in the case that the neuron has switched to a plateau state; `'out'` is a boolean flag indicating whether or not the neuron is transmitting; and `'in'` is the strength of the input signal.

Neurons communicate with each other via chemical and electrical synaptic connections and these are also represented as objects. For example, the Maude specifications

```
[SC | from: B63 to: B31]
[EC | from: B63 to: B31]
```

represent chemical and electrical synaptic connections from a neuron named B63 to one named B31. Transmission of signals along these different routes is represented by two Maude operators, `xmit`, and `fired`. For example, the latter is specified as

```
op fired : NeuronId Nat -> Msg.
```

Here, `NeuronId` and `Msg` are Maude sorts and `Nat` is a Maude specification of the nonnegative integers.

It is convenient to further refine the specification of signal transmission along an electrical connection by specifying an additional Maude operator called `transmit`. This is expressed as an equation

```
eq transmit(nid,j,b, [EC|from: nid, to: nid1 ]
             [nid1 : nc |atts, in: j1] conf)
   = [EC|from: nid, to: nid1 ] transmit(nid,j,b,
             [nid1 : nc |atts, in: (j + j1) ] conf).
```

Ignoring details, the semantics of this operation is that the neuron `nid` adds the integer `j` into the incoming signal of neuron `nid1` when the two are connected by means of an electrical connection. Different types of neurons, in terms of their functional roles, are represented by a Maude sort specification called `NeuronClass`. The Maude `op` specification then denotes particular types of neurons. For example, the construct

```
op SG: - > NeuronClass . specifies a neuron acting as a signal
generator.
```

Using these constructs, the two neuron model in Fig. 1A can be represented symbolically as the Maude specification:

```
op inc : -> NCConf .
eq inc = B63 B31 [EC | from: B31 to: B63]
           [EC | from: B63 to: B31] [SC | from: B63 to: B31].
```

In typical Maude syntax, the `op` statement specifies as a constant of data type `NCCConf` (i.e., Neuron configuration). The subsequent equation defines `inc` to be a configuration consisting of a B31 neuron, a B63 neuron (B31 and B63 are similarly defined by equations), an electrical connection from B31 to B62, an electrical connection from B63 to B31 and a synaptic connection from B63 to B31. Next, rewrite rules are used to describe system dynamics, i.e., how each neuron evolves as a function of its state and input signals. Since many questions of interest in the neural context relate to the temporal behavior of neural signaling, our model incorporates `tick`, and `tock`, two mechanisms for describing the passage of time. The Maude `op tick` takes a neuron as argument, and represents the passage of a unit of time in some abstract time system for that neuron. In that sense it is like the system clock of a computer system. The `tock op` takes as argument a configuration of neurons and connections between them and propagates ticks through the configuration.

An example of a rewrite rule is given below. The following rule expresses the behavior of a neuron in plateau state where there is positive decay time remaining  $s(t)$  stands for the successor of a number  $t$ . If  $t$  is an integer then  $s(t) = t + 1$ . The statement to the right of the string ``****'` is a comment line in Maude and is ignored by the Maude processor.

```
rl[sig.read.on]: **** ignore signal decrement decay
tick(nid) [ nid : Neuron | thresh: i, decay: d,
dplevel: l, ttl: s t, out: true, in: j ]
=> [nid : Neuron | thresh: i, decay: d, dplevel: l,
    ttl: t, out: true, in: 0 ] xmit(nid,j,true).
```

The rule says that when a unit of time passes, a neuron with state matching the expression above the arrow, `=>`, processes its input signal and updates its state according to the neuron expression below the arrow. The expression `xmit(nid,j,true)` represents electrical transmission by the neuron. Then, the remaining decay time is decremented, the neuron ignores its input, and continues to transmit.

When the model neural system executes, each neuron first processes its input. Then, using the connection objects, signals are propagated from pre- to postsynaptic neurons, and signals arriving at each neuron are combined to determine a net input signal. One way to use the model is to introduce an input signal object programmed to fire according to a given pattern specified by time units on, time units off and observe the sequence of outputs of the signal and neuron objects. This is accomplished in our PL model using a Maude `op` called `icr`.

```
eq icr(k,l,j) = { tock(inc sg(Cmd,k,l,j)
    [EC|from: Cmd, to: B63]) ; [B63 : nil] [B31 : nil]
    [Cmd : nil] }.
```

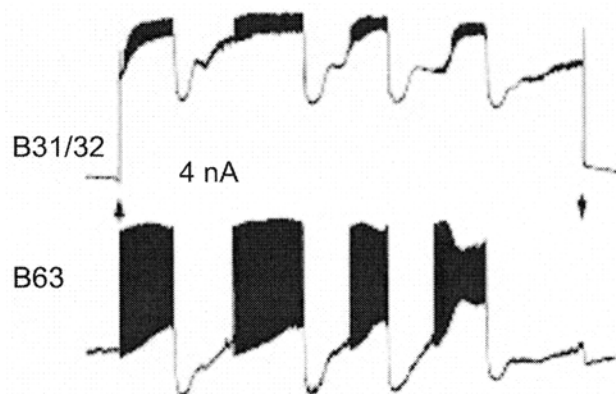
This operator takes as argument three integers representing the input signal pattern, the on-time, off-time (i.e., duration of time the neuron is off), and input signal strength. The equation implementing `icr` uses the `tock` operator to pass these parameters into a specified neuron configuration and signal generator that generates command signals according to these parameters. Note that the input stimulus (via the neuron denoted `Cmd`), is applied to the B63 neuron in the clause `[EC|from: Cmd, to: B63]`. Another `op` called `idr`, in which B63 is replaced by B31 in the same clause, represents stimulating the B31 neuron. The

behavior of the two neuron circuit can then be studied by varying input parameters to `icr` and `idr`, and running the model using Maude's rewrite command.

The following are results of an example execution of the model. A system driven by an input signal that is on for a long time can be modeled by executing the Maude rewrite on `icr(100,1,1)`. This results in alternating burst and resting states. This behavior corresponds respectively to Figure 2, which is adapted from [15].

```
[Cmd : 1 1 1 1 1 1 1 1 1 1 1 1 1]
[B63 : 0 1 1 1 1 1 0 1 1 1 1 1 0]
[B31 : 0 1 1 1 1 1 0 1 1 1 1 1 0]
```

In addition, one of the experimental results observed in [15] is that there is a strong coupling between B63 and B31 in the sense that stimulating either gave rise to similar behavior from both. Our PL model was able to reproduce this since identical results were obtained (not shown here) when rewriting the op `idr(100,1,1)` instead of `icr(100,1,1)`. It can be seen that our approach allows us to easily represent properties of neuron cells that govern their behavior such as thresholds and plateau potentials, as well as different types of inter-cellular connections, and to reproduce key features of known behavior.



**Figure 2 - Rhythmic bursting in B31/32 and B63 neurons**

Continuous stimulation of the B31/32 neuron results in rhythmic bursting in B31/32 and B63 neurons. Similar behavior results when B63 is stimulated. From [15].

Also, the model enables easy modification of the properties characterizing a neuron and observation of the effects on its behavior, singly and as a part of more complex circuits. The ability to carry out such *in silico* experiments is one of the benefits of this proposed approach. We conjecture, therefore, that more complete models which include representations of additional neuron properties, will be capable of providing new understanding of more complex behavior as well as testable hypotheses regarding the effects of modifying circuit elements.

## Conclusions

Results from the PL model for the small neural pathway described above have encouraged us to believe that such tools can be a very useful new addition to neuroinformatics. Currently, neuroinformatics focuses primarily on anatomical features of the brain, which are related to

the Human Brain Project (HBP) (see <http://www.nimh.nih.gov/neuroinformatics/>) and to a lesser extent on databases of computational models (see <http://senselab.med.yale.edu/senselab/modeldb/>). The approach that is outlined in the present paper represents a paradigm shift in neuroinformatics. PL provides a method for encoding, with abstract and qualitative methods, the dynamical properties of neurons, synapses and neural circuits. Among the key benefits we anticipate is that the new modeling approach we propose here will guide hypothesis generation and, ideally, enable detailed quantitative models in a smaller, restricted parameter space. In this way, our approach will benefit classical quantitative modeling in neuroscience by enhancing their computational tractability and robustness. Another feature of PL models is the usability: the PL models of signal transduction are curated by biologists. This is possible because the representation is very stylized, and the difficulty is in inferring the elements of rules from experimental results which requires biological insights. When the neuron modeling techniques mature, we expect neuroscientists to be able to develop their own models in a similar manner. In fact it will probably be simpler than developing signaling models. Moreover, the method is easily scalable from single cells to small identified neural circuits to complete sub-systems. PL models of signal transduction with several thousand species and reactions have been already developed and the analyses scale quite well. Such a simulation environment will be a valuable tool in the quest for understanding the ways in which the nervous system functions and the ways in which pathologies affect neural function. More complete models including representations of additional neuron properties, might be capable of providing new understanding of more complex behavior and testable hypotheses regarding the effects of modifying circuit elements, complementing, for example, previously developed H-H models of relatively complex neural networks [23]. Of course the representation makes abstractions and there is still much work to be done to refine the attributes of neurons and abstractly model signal integration and other features adequately. The future work includes also studying the semantic consequences of these abstractions using techniques such as qualitative reasoning. Finally, it will be important to organize the results of these models in order to make these available in a structured and accessible way (along the line of the semantic web approach [e.g., 26,27]). As the modeling technology matures we envision that the semantic web technology would be useful to develop ‘workflow’ scenarios to analyze behavior of proposed neuron circuits under a variety of conditions and parameter setting.

## Methods

### The Pathway Logic Model

In this section we describe a methodology of formal modeling called Pathway Logic (PL). PL [3-7] is an approach to the modeling and analysis of molecular and cellular processes based on *rewriting logic* [24]. PL models reflect the ways that biologists think about problems using informal models. They are curated from the literature, and written and analyzed using Maude, a rewriting-logic-based system [8,9]. So far, PL has been successful in developing models of mammalian signal transduction networks [3-7]. Importantly, a PL model itself also serves as a database containing heterogeneous knowledge of facts, processes, and pathways at multiple levels of abstraction.

### Rewriting Logic and Maude

Rewriting logic [24], a logical formalism widely used for modeling and reasoning about concurrent and interactive systems [25], is based on two simple ideas: 1) states of a system are represented as elements of an algebraic data type; and 2) the behavior of a system, i.e., the dynamical nature of the system, is given by local transitions between states described by



*rewrite rules*. Algebraic data types are specified by declaring sorts (names of the data types), subsort relations (one data type may be a subset of another), and operations (naming functions and specifying argument and result types), and by giving equations that define the functions computed by the operations. Terms  $t$  that denote elements of the data types can be variables (denoting some unspecified element), constants, or the application of an operation to a tuple of argument terms,  $f(t_1, \dots, t_n)$ . In its simplest form, a rewrite rule has the form  $t \Rightarrow t'$  where  $t$  and  $t'$  represent a local part of the system state. This rule says that when the system has a subcomponent matching  $t$ , that subcomponent can be rewritten to  $t'$ , possibly concurrently with changes described by rules matching other parts of the system state. The process of application of rewrite rules can be thought of as computation or deduction. Rewriting logic allows reasoning about complex changes that could occur given the possible transitions specified by a particular symbolic model. The Maude system, which executes on commonly available platforms such as Macintosh and Linux, is a very efficient implementation of rewriting logic with powerful deduction capabilities automating much of the reasoning. There are many kinds of computational analysis that can be performed using Maude. Execution can be used to find some pathway leading from an initial state. Search can be used to find all pathways leading from an initial state to a state satisfying a desired property. The Maude model-checker can be used to answer more complex questions about possible pathways leading from a given initial state. Maude also provides efficient support for *reflection*. This allows models to be treated as *ndata* and makes it easy to program operations that transform models to and from representations suitable for input to other logical systems for visualization or analysis. See <http://maude.cs.uiuc.edu/>.

### **Pathway Logic Basics**

In the PL approach, the state of a biological system such as the epidermal growth factor receptor (EGFR) network [4,5], the tumor necrosis factor receptor [6] pathway, or a neural signaling pathway is represented as a term in an equational theory and rewrite rules are used to describe a local change that could occur when an instance of the left-hand side of a rule exists. PL models are structured in four layers: (1) sorts and operations, (2) components, (3) rules, and (4) queries. The ‘sorts and operations’ layer declares the main sorts and subsort relations, i.e., data types and relations between data types. The *components* layer specifies particular entities (proteins, neurons) and introduces additional sorts for grouping entities in families. The *rules* layer contains rewrite rules specifying individual steps of a process. For example, these correspond to reactions in traditional metabolic and interaction databases. The *queries* layer specifies initial states and properties of interest. A feature of PL models that enables the execution of *in silico* experiments is that the models can be executed in Maude to simulate a biological signaling process over time. They can also be formally analyzed to reason about properties of the states reachable from an initial state. The complete symbolic exploration of all reaction interactions can provide useful insights. An example of an *in silico* experiment is to observe whether a desired end state can be reached if specific intermediate states are ‘knocked-out’ from the network.

## Competing interest

The authors declare that they have no competing interests.

## Authors' contributions

Each of the co-authors contributed equally to this project.

## Acknowledgements

This project was supported by NIH grants R01RR11626 and P01NS38310.

## References

1. Baxter DA, Canavier CC, Byrne JH: **Dynamical properties of excitable Membranes.** In *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience*. Edited by Byrne JH, Roberts J. New York: Academic Press 2003, 161-196.
2. Abbott LF: **Single neuron dynamics: an introduction.** In: *Neural Modeling and Neural Networks*. Edited by Ventriglia F. New York: Pergamon Press 1994, 57-78.
3. Eker S, Laderoute K, Lincoln P, Sriram MG, Talcott C: **Representing and simulating protein functional domains in signal transduction using MAUDE.** In: *First International Workshop on Computational Methods in Systems Biology (CSMB03)*, 2003, 164-165.
4. Eker S, Knapp M, Laderoute K, Lincoln P, Meseguer J, Sonmez K: **Pathway logic: Symbolic analysis of biological signaling.** In: *Proceedings of the Pacific Symposium on Biocomputing*. 2002, 400-412.
5. Eker S, Knapp M, Laderoute K, Lincoln P, Talcott C: **Pathway logic: Executable models of biological networks.** In: *Fourth International Workshop on Rewriting Logic and Its Applications*, In: *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002, volume 71.
6. Sriram, MG: **Modeling protein functional domains in Signal Transduction using Maude.** *Brief Bioinform* 2003, **4**: 236-245.
7. Talcott C, Eker S, Knapp M, Lincoln P, Laderoute K: **Pathway logic modeling of protein functional domains in signal transduction.** In: *Proceedings of the Pacific Symposium on Biocomputing*, 2004.
8. Clavel M, Duran F, Eker S, Lincoln P, Marti-Oliet N, Meseguer J, Talcott C: **Maude 2.0 Manual.** 2003. <http://maude.cs.uiuc.edu>.
9. Clavel M, Duran F, Eker S, Lincoln P, Marti-Oliet N, Meseguer J, Talcott C: **The Maude 2.0 system.** In: *Rewriting Techniques and Applications (RTA'03)*, Lecture Notes in Computer Science. Springer-Verlag, 2003, 76-87.
10. **Rediscovering Biology - Molecular to Global Perspective – Neurobiology** [http://www.learner.org/channel/courses/biology/support/10\\_neuro.pdf](http://www.learner.org/channel/courses/biology/support/10_neuro.pdf).
11. Pearson KG: **Common principles of motor control in vertebrates and invertebrates.** *Annu Rev Neurosci* 1993, **16**: 265-297.
12. Nargeot R, Baxter DA, Byrne JH: **In vitro analog of operant conditioning in *Aplysia*. II. Modifications of the functional dynamics of an identified neuron contribute to motor pattern selection.** *J Neurosci* 1999, **19**: 2261-2272.

13. Botzer D, Markovich S Susswein AJ: **Multiple memory processes following training that a food is inedible in *Aplysia***. *Learn Mem* 1998, 5: 204-19.
14. Lechner HA, Baxter DA, Byrne JH: **Classical conditioning of feeding in *Aplysia*: II. Neurophysiological correlates**. *J Neurosci* 2000, 20: 3377-3386.
15. Susswein AJ, Hurwitz I, Thorne R, Byrne JH, Baxter DA: **Mechanisms underlying fictive feeding in *Aplysia*: coupling between a large neuron with plateau potentials and a spiking neuron**. *J Neurophysiol* 2002, 87: 2307-2323.
16. Hayes RD, Byrne JH, Baxter DA. **Neurosimulation: tools and resources**. In *The Handbook of Brain Theory and Neural Networks (Second Edition)*. Edited by Arbib M A, The MIT Press, Cambridge: 2002, 776-780.
17. Bhagavan, NV **Medical Biochemistry**, pp 88 – 91 Academic Press, San Diego 2002.
18. Clarke EM, Grumberg O, Peled DA: **Model Checking**. MIT Press 2000.
19. Peterson JL: **Petri Nets: Properties, analysis, and applications**. Prentice-Hall, 1981.
20. Priami C , Regev A , Shapiro E , Silverman W: **Application of a stochastic name-passing calculus to representation and simulation of molecular processes**. *Info Proc Let* 2001, 80: 25-31.
21. Harel D: **Statecharts: A visual formalism for complex systems**. *Sci Comput Prog* 1987, 8: 231–274.
22. Chabrier-Rivier N, Fages F, Soliman S: **Modelling and querying interaction networks in the biochemical abstract machine biocham**. *J Biol Phys Chem* 2004, 4: 64-73.
23. Cataldo E, Byrne JH, Baxter DA: **Computational model of a central pattern generator**. *Lect Notes Bioinform* 2006, 4210: 242-256.
24. Marti-Oliet N, Meseguer J: **Rewriting logic as a logical and semantic framework**. In: *Handbook of Philosophical Logic*. Edited by Gabbay D, Kluwer Academic Publishers, 1997.
25. Marti-Oliet N, Meseguer J, Palomino M: **Rewriting logic and applications bibliography**. *Theor Comput Sci* 2002, 285: 121-154.
26. Ruttenberg A, Clark T, Bug W, Samwald M, Bodenreider O, et al.: **Advancing translational research with the Semantic Web**. *BMC Bioinform* 2007, 8 (Suppl 3):S2: doi: 10.1186/147-2105-8-S3-S2.
27. Luciano JS, Stevens RD: e-Science and biological pathway semantics. *BMC Bioinform* 2007, 8 (Suppl 3):S2: doi: 10.1186/147-2105-8-S3-S3.