Report from the 2010 NeuroML Development Workshop

Workshop Attendees

Murat Alp	University of Oregon, USA
Guy Billings	University College London, UK
Avrama Blackwell	George Mason University, USA
Upi Bhalla	National Center for Biological Science, India
Ted Carnevale	Yale University, USA
Robert Cannon	Textensor Limited, UK
Hugo Cornelis	University of Texas Health Science Center, USA
Sharon Crook	Arizona State University, USA
Andrew Davison	Centre National de la Recherche Scientifique, France
Suzanne Dietrich	Arizona State University, USA
Padraig Gleeson	University College London, UK
Joe Graham	Arizona State University, USA
Marco Herrera-Valdez	Arizona State University, USA
Mike Hucka	CalTech, USA
Stephen Larson	University of California San Diego, USA
Henrik Lindén	Norwegian University of Life Sciences, Norway
Michele Mattioni	European Bioinformatics Institute, UK
Alfredo Rodriguez	Mount Sinai School of Medicine, USA
Angus Silver	University College London, UK

Introduction

The second NeuroML Development Workshop was held March 4-5, 2010 at Arizona State University. The main goals of the workshop were to engage in discussion that would lead to further development of version 2.0 of NeuroML and to explore the future role of NeuroML in the computational neuroscience community. After a review of the work to date and presentations on related projects including presentations by team members from SBML, the Whole Brain Catalog, NIF, and the INCF Multiscale Modeling

Program, the group focused on a discussion of the overall aims and structure of NeuroML 2.0. This was followed by breakout sessions that focused on subdomains of NeuroML--specifically, morphologies and distributions of biophysical properties, channels, and synapses. The final discussion of the workshop involved the entire group coming back together to discuss future plans for the implementation of version 2.0. These discussions are summarized below with contributions from Guy Billings, Upi Bhalla, Robert Cannon, Sharon Crook, and Padraig Gleeson.

Overview: Overall Aims and Structure of NeuroML 2.0

There was a great deal of discussion concerning the overall modular structure of NeuroML and which modules should be configured to stand alone (MorphML, ChannelML, NetworkML). Most contributors did not feel strongly about whether NeuroML is defined in one schema or several schemas as long as modularity is maintained. As in earlier discussions, it was decided to adopt camelCase for naming, and that Levels add confusion to the overall structure of the schemas. Levels will be removed in 2.0. However, many agreed that "compliance scenarios" would be helpful for software developers, where different software might be compliant with morphology descriptions only, single cell morphology and biophysics, or networks for example. A lively discussion followed on the topic of units, with input from Mike Hucka concerning the problems units have caused to the SBML community. This resulted in a decision to support “freeform” units only with all units explicitly provided in the XML.

A discussion of the overall aims of NeuroML led to the suggestion that a mission statement that explicitly defines the vision of NeuroML would be helpful.

Next Steps

It was decided that many of the more technical issues that were discussed will be addressed after the NeuroML 2.0 description language develops further. Sharon Crook will draft a mission statement, and the workshop participants will provide comments and edits after the workshop.

Overview: Specifications for Morphologies and Distributions of Biophysical Properties in NeuroML 2.0

The discussion began with an overview of the discussion and decisions from last year’s workshop. In particular, there was agreement to maintain the use of segments, to simplify the schema by removing cables which many developers find confusing and are not necessary, and to provide ways to group together parts of a morphology through groups of segments. It was suggested that all group information should appear after the segment definitions if possible, and that groups be defined explicitly or using higher-level constructs such as along a path between two points (path <from> and <to>) or by defining a subtree that is distal to a particular point (subtree <from>).

There was further discussion of how to define specific types of morphological groups such as apical dendrites, basal dendrites, axons, etc. Several options include having predefined names for common types or linking to ontologies that define these types. We

suggest adding tags or rdf for metadata that provide NeuroLex ontology ids to groups. We propose to begin with simple tags, and when a tag is present, one should assume it indicates “is a”. If more complicated semantic information is needed, we can use rdf in a way that is similar to SBML.

A lively discussion of fractAlong led to many suggestions of usage and alternatives such as distAlong, etc. It was decided that fractional distances prevent many possible mistakes, make shrinkage correction easier, and prevent overshoots. Therefore, fractAlong will be maintained but it will be based on segments rather than cables.

Discussion of features for fiducials led to the decision to maintain these components of NeuroML. There needs to be consideration of how to link NeuroML to descriptions of mesh-based structures as well.

The discussion of biophysical properties led to a separation of membraneProperties, intracellularProperties, and extracellularProperties that are applied to all segments when a group is not defined, or to user-defined groups of segments, or to individual explicitly listed segments. The problem of how to define different extracellular areas was also discussed as well as the placement of information about the reversal potential, which is technically a membrane property. It will also be important to provide a way to allow similar biophysics in varying morphologies in an efficient way. Using the same names for groups across cells should allow for easy linking of channel distributions to multiple cells.

Next Steps

Stephen Larson will provide input on the development of links to ontologies based on what has been done in the Whole Brain Catalog. He will also provide some input on possible interfaces between NeuroML and mesh-based structures. The specification committee will consider issues relating to biophysical properties that were not discussed at the workshop. The demo examples of NeuroML version 2.0 files will be updated with the new format.

Overview: Channel Specifications in NeuroML 2.0

The key requirements identified for the extension and revision of the NeuroML channel specification for version 2.0 were:

1. modifying the scope to cover gap junctions and channels gated by voltage and/or ligands, but to exclude some of the existing channel mechanisms that will instead be addressed by the synapse specification. It should include “instantaneous” channels where relative state occupancy is expressed algebraically, rather than by time dependent differential equations.
2. generalizing channel specifications to reuse a small number of base components, thereby reducing the number of main elements. This includes replacing parameterized transition rates with references to user-defined functions and using kinetic schemes for gating mechanisms.
3. separation of the state representation, which yields an open fraction or open probability for a channel, and the electrical properties to do with conductance laws and pore permeation.
4. a means to express how a channel interacts with other parts of a model such

that, for example, the membrane potential or calcium concentration used in a gating expression can be identified with the right quantities on an extended cell model.

The first part of the discussion addressed suggestions in the ChannelML 2.0 proposal document in which it had been suggested that kinetic schemes should be used for all channel models (with multiple schemes, and multiple instances of a given scheme for models based on the HH equations). One concern with this proposal was the extent to which it associates channel models with specific kinetic mechanisms. For example, a mechanism in which discrete gates are represented by kinetic schemes can only have an integer number of instances of a gate, whereas if the HH formalism is used purely as an abstract mathematical system, then there is no reason to preclude non integer gate powers. The consensus was that we should provide elements to express this distinction so that models can specify which case is intended, rather than try to make one structure fit both interpretations. Another concern was the extent to which the definition of structures for kinetic schemes within NeuroML duplicates capabilities in SBML, which is also able to express kinetic schemes (albeit not with a specific structure, but instead by defining pseudo-species for each state and reaction rates for their concentrations). Opinion was divided on the pros and cons of representing channels in SBML, with a decision that the proponents of each approach should pursue their ideas further, develop representations for a range of example models, and report on their experiences.

Point 4 above was identified as a general requirement for NeuroML with two proposed solutions: NeuroML could define a set of reserved words (v, Ca_in, Ca_ext, Na_in etc) that could be used without further explanation in models; or quantities that are exposed by a model component could be linked to an external ontology such as SBO to define their role. Note that these are not mutually exclusive: the reserved words could be linked to an ontology in any case. The preference was to reserve a small set of terms as above initially, with the possibility that future versions could require ontology references and demote these words from being reserved to being loaded as a convenient set of ontology references.

A possible problem for any high level channel representation, including the kinetic scheme based one, is that they are not easily mapped to the internal representations used by most simulators (current exceptions are channels built with Neuron's channel builder, and PSICS). This makes it unlikely that simulators will be able to export existing channel models in this form without manual editing. It will also require simulators to support a higher level internal channel representation (which should be relatively easy via libNeuroML) before round-tripping becomes possible. Given the limited number of channel models in use, a requirement for one-off manual intervention was considered acceptable in view of the benefits offered by higher level representations.

Other points discussed were:

- Is there is a need for functions that return vectors rather than single quantities? The proposal document suggested this for transitions where the forward and reverse rates are both functions of the same set of parameters. It is inelegant to duplicate the parameters in the specification because it suggests they can be meaningfully changed independently. Others found this less inelegant than having the two options of either supplying separate function and parameter sets for the forward and reverse rates, or supplying them both together with a function that returns two values. This should become clearer with the development of

more concrete examples.

- Can channel conductance, e.g. as specified with the Nernst or GHK equations, be expressed purely by a generic function and dependency mechanism, or whether it requires special elements to express these relations? Again, development of examples should clarify the situation.

Extending the range of fully worked example models was widely felt to be the most productive way forward. These use cases should include:

1. Mechanistic state scheme representing Hodgkin Huxley model
2. Hodgkin Huxley model as an abstract entity with fractional powers
3. Calcium channel with Nernst potential
4. Mixed channel with GHK equations for current fraction carried by Calcium
5. Calcium-dependent Potassium channel
6. Mg-block and other voltage-dependent blocks
7. Regular gap junction
8. Rectifying gap junction
9. Synaptic channels
10. NMDA receptor, which includes numbers 4 and 6

Next Steps

Robert Cannon and Padraig Gleeson will develop examples of how the use cases models could be represented using kinetic schemes and functions for the electrical properties.

Upi Bhalla's group will look into expressing the same structures with SBML.

LibNeuroML will provide a range of features useful to simulator developers who wish to support NeuroML channel formats including: reading and writing models; checking of correctness; utility functions, including converting between MathML and something writable; unit handling; an object model for lossless storage of the input model; version handling.

Overview: Synapse Specifications in NeuroML 2.0

At present, NeuroML provides a range of "pre-packaged" synapse types closely matched to those used in existing models for Neuron and Genesis. The goal for version two is to extend the range of models that can be expressed by introducing a more flexible and modular scheme. The existing types could either be preserved as they are, or represented as parameterized instances of models defined in the new system.

Discussion focused around two main topics: a possible structure for a more modular and flexible synapse specification; and the wide range of synaptic phenomena that might or might not be expressible in such a structure.

The existing proposal is to use the same structures as for ion channels on the post-synaptic side for expressing the receptors and corresponding conductances, and to define new components to express the presynaptic state, plasticity rules and input

transformations. The components for the presynaptic state and plasticity rules need to be able to express a wide range of possible behaviors and could include kinetic scheme elements as for the channel specification, state variables governed by differential equations or reaction networks expressed SBML. Input transformations serve to receive events or monitor continuous quantities such as the membrane potential or ambient neurotransmitter, and generate outputs that go to the post-synaptic unit and plasticity rules. Wherever possible, postsynaptic units should refer to elements from the channel specification rather than duplicating their structure.

The main benefits of such a scheme come from its modularity, providing “mix and match” pre- and post-synaptic elements, its coverage of exiting models, and from making all the components of a model explicit, so that the states, variables and dependencies necessary to implement a model are present in its specification.

Two main areas of concerns were raised with this proposal. First, as with any system that allows a wide range of models to be expressed, it may become difficult to provide efficient implementations, or even any implementation, of some models. The second concern focused on the limitations of what can be expressed leading into the second half of the discussion. Models discussed that might present a problem for this scheme included:

- broadcast neurotransmitter affecting dendritically distributed ligand gated channels. This would poses a problem for schemes that require a single pre-and post- unit in each synapse.
- hormonal gating of ligand gated channels.
- synapses involving a post-synaptic reaction network that might most conveniently be expressed in SBML (and presumably couldn't be expressed with the NeuroML ion channel components alone).

As with the discussion of channels, it was felt that the most productive way to advance the specification at this stage is develop concrete example specifications covering a wide range of use cases. These could include:

1. Single-alpha-function event-driven channel
2. Dual-alpha-function event-driven channel
3. 'Poisson' channel with user-defined firing probability
4. 'Poisson' channel with state-variable-controlled firing probability
5. Ligand-gated receptor with Nernst potential
6. Ligand-gated receptor with GHK calculation for fraction of Ca current
7. NMDA receptor
8. Dendro-dendritic synapse
9. Bidirectional dendro-dendritic synapse
10. Hebb synapse
11. Unity-convergence synapses
12. High-convergence synapses
13. Multi-site synapses: can they simply be composites?

14. Alpha-function event-driven channel with history dependence
15. Alpha-function event-driven channel with independent release probability
16. Alpha-function event-driven channel with history-dependent release probability
17. Co-transmitter
18. Modulator
19. Bi-directional dendro-dendritic synapse with plasticity on one side
20. Metabotropic receptor that opens a voltage-gated channel
21. Metabotropic receptor that modulates a voltage-gated channel
22. Stochastic variants: single channel-level descriptions
23. 3-D diffusion for the mechanism of vesicle release
24. Spillover of transmitter
25. Non-punctate release from glia
26. State-based models.

Next Steps

Members of the group will refine the proposal for modular structures, look into ways to interface with SBML, and develop sample model specifications for cases from the above list. Robert Cannon and Padraig Gleeson will work on an initial set of synapse definitions for further consideration by the group.

Overview: Future Development and Tools

One of the most important necessities for future uptake of NeuroML is the development of libNeuroML to aid in tool development. libNeuroML for v1.x of NeuroML is currently under development by Upi Bhalla's lab. The key requirements for libNeuroML are:

- read and write capabilities
- validation (in part from schema but perhaps with additional checks)
- utility functions (e.g. converting from MathML to string format)
- unit checking
- annotation support (which could help with round tripping)
- version conversion

Additional areas for future consideration are language support and issues of full instantiation.

Throughout the development of NeuroML, there has been a clear separation between the model description and the specification of simulation control and other procedural specifications. SED-ML was mentioned repeatedly during the workshop as an area of future investigation for specification of computational "protocols".

Further discussion focused on the need for a wiki-based approach for the NeuroML website that would provide more accessible documentation and a general rationale for decisions. There was some discussion of how to improve the volume of discussion on

the mailing list, eventually elect editors for NeuroML, and formalize the process for users to request or propose extensions to NeuroML.

Next Steps

In the short term, Upi Bhalla's group will continue the development of a working prototype for libNeuroML based on NeuroML v1.x; however, a plan is needed for development of libNeuroML for version 2.0. Padraig Gleeson is developing a proposal for a formalized process for extensions to NeuroML, which will be discussed via the NeuroML mailing list. We need a volunteer to approach the SED-ML developers for discussions of overlap between these languages or additions to SED-ML in support of the needs of the computational neuroscience community. Sharon Crook will investigate additions to the website and a possible reorganization of the mailing lists. Specification committees will be updated with a goal of voting for editors as the mailing list subscription numbers grow.