Emerging Standards for Computational Neuroscience

NeuroML Development Workshop: Biophysical Single Cell Modelling
March 9-10, 2009, London, UK

Supported by: Wellcome Trust, International Neuroinformatics Coordinating Facility & National Science Foundation

Meeting Organizers

Padraig Gleeson and Angus Silver, University College London, UK
Sharon Crook, Arizona State University, USA
Robert Cannon, Textensor Limited, UK

Background

The complexity of problems associated with structure and function in neuroscience requires that research from multiple groups across many disciplines be combined. In order to achieve this, there must be an infrastructure for exchanging neuronal model specifications; however, the current use of multiple formats for encoding model information has hampered model exchange. NeuroML is a model description language developed in XML that allows the specification of models of neuronal systems based on the underlying physiological and anatomical properties, independently of any one simulator, allowing for greater simulator interoperability and model exchange.

The declarative specifications produced by the NeuroML initiative are currently arranged into levels. Level 1 deals with neuroanatomical information (MorphML) and metadata. Level 2 allows for specification of cell models with realistic channels (ChannelML) and synaptic mechanisms distributed on their membranes, and Level 3 describes networks of these cells in three dimensions (NetworkML). This workshop brought together members of the computational neuroscience community including modellers, software developers and experimentalists to focus on the refinement of NeuroML related to single cell modelling including ChannelML and the biophysical description of cells.

There were presentations on the current structure of NeuroML, comparisons with the SBML initiative and detailed discussions of proposals for updating specific parts of the language, including MorphML and ChannelML. The administrative structure of the NeuroML initiative was made more formal with introduction of a NeuroML Team, and Working Groups were set up to start to formally specify the updates agreed at the meeting.

Introductory presentations

After welcoming remarks and round table introductions of workshop participants, Padraig Gleeson presented an overview of NeuroML. This presentation included a brief history of NeuroML, the current scope, the design philosophy, current tool support, and testing using detailed neuronal models currently implemented in NeuroML. Following this presentation, Nicolas Le Novere provided an introduction to SBML, including many details that might be relevant to

further development of NeuroML.

To set the stage for the workshop discussion, Sharon Crook presented a brief overview of the participant comments that were solicited prior to the meeting and the organization of the workshop based on this feedback. Erik De Schutter followed with a presentation on his thoughts regarding NeuroML. He contrasted the current state of NeuroML with the rich ecology of modelling tools on the SBML website. He expressed opinions about semantic structure problems with the current version of NeuroML, which he feels lead to difficulties with extensions. Dr. De Schutter also advocated a layered approach for NeuroML with a descriptive layer and an abstraction layer that describes the mathematical concepts. More details regarding this discussion are available on the NeuroML Sourceforge website in the email archive.

Simulator independent descriptions of neuronal morphologies *(Moderator: Sharon Crook)*

Sharon Crook provided a brief description of the current elements in MorphML and a summary of feedback on MorphML. The first topic of discussion focused on nesting in the specification. In general it was felt that 1) there is not a clear need for pluralized elements that contain multiple examples of singular elements, e.g. <segments>, 2) all dendritic structures can be represented using elements and 3) descriptions should continue to be completely independent of compartmentalization issues.

There was some discussion of the choice early in the development of MorphML to use points and diameters to represent cross-sections that are grouped into a segment, with a consensus that this is the most practical, and conceptually straight forward approach. The use of proximal and distal locations on segments was discussed, which leads to the need for specification of a “root” segment rather than assuming that the root is a soma compartment. There was agreement that a single, flexible grouping mechanism is needed. A “beyond” mechanism would provide a simple way to group all of the dendritic tree distal to a specified location. There was also agreement that the fract_along_parent attribute is useful for specifying relative locations and should be included in the element. This could work well with templates and relative positions (e.g. describing spines once and using them often throughout a morphology). Overall, workshop participants expressed a desire for continued support of bridges with the neuroanatomy community through MorphML.

The discussion moved on to the metadata structure with a brief overview of the current elements and a summary of the feedback on metadata. There was some discussion of the use of separate files for metadata and whether NeuroML should use RDF for the metadata framework. Workshop participants agreed that in the future, incorporation of brain region, cell type, and species information should be based on the use of existing ontologies, and the specification of metadata for subcellular structures should be based on previous work from the BIRN group. There was some discussion of the development of a description of minimal information required for neural models and the model specification curation process.

Recommendations about these issues will be made by the Morphology and Channel Distributions Working Group described below. Future discussions also should consider extensions to MorphML such as specifications for a mesh structure for 3D surfaces, variable morphologies within populations; neuronal structural plasticity or growth during simulations, and hemispherical ends on segments. There was agreement that the issues of describing mesh structures and variable morphologies could probably be postponed until the NeuroML workshop next year.

Simulator independent spatial descriptions of electrical properties of cells *(Moderator: Padraig Gleeson)*

This part of the discussion dealt with the part of the language, which allowed morphological descriptions of cells at Level 1 to be extended with a description of the electrical properties across the cell (e.g. specific capacitance of the membrane, channel density on different parts of the cell). An overview was given on the current elements used to describe this information, centring on the <biophysics> element, which can be added to <cell>, and the structure of the subelements. It was generally agreed that the current structure (with nested mechanism, parameter, group elements) was too verbose, and a more compact form was proposed (a channel_density element with channel, group, param and value attributes). It was agreed that the term "mechanism" should be avoided and that specification of internal concentrations, e.g. calcium pools should be treated separately.

There was also a proposal for using standardised function calls in simulators based on this structure as a basis for changing channel properties in cells (as opposed to addressing sections/compartments directly), which would allow for more portable scripts for interacting with cells. This could be helped by defining a NeuroML based API with functions for changing properties of the cells (e.g. change parameter A of channel B on group C to value D). The implementation of this would shield the simulator specific implementation of the cell (e.g. the set of compartments on GENESIS or sections on NEURON) and would carry out the operations needed to implement the method called in the API (e.g. changing the maximal conductance in all sections in "dendrite_group"). This would enable easier recompartmentalisation of cells within the different simulators.

An initial proposal for extending the new way of describing electrical properties to allow references to external pathways specifying reactions in SBML (e.g. an internal_species element with a pathway attribute) was presented. It was pointed out that this would require a standardised way to reference species in the SBML file from NeuroML and vice versa. The example of varying internal calcium and its effect on e.g. Ca^{2+}-dependent K^{+} channels would be a good first example to try to support in any potential implementation.

In general meeting participants were in support of a mechanism whereby a description of channel densities, etc. could be reused easily by multiple cell morphologies. The issue of describing non-uniform channel distributions was discussed briefly, and there was a suggestion that the specification of the functional form of these (e.g. how the maximal conductance changes as a function of path length along dendrites) could be specified in a similar way to the new form for specifying the rate equations for channels (potentially describing the function in MathML).

These issues will be discussed further and a new set of elements agreed in the Morphology and Channel Distributions Working Group described below.

Simulator independent descriptions of membrane conductances *(Moderator: Robert Cannon)*

Robert Cannon presented a brief overview of the use of declarative XML formats for representing ion channels including the initial NeuroML proposal, ChannelDB, ChannelML prior to version 1.8 and the most recent structures in 1.8. The feedback received prior to the meeting concerned mainly the scope of what could be described along with some more detailed comments on the current specification for voltage gated channels. The latter were addressed first with a review of possible ways to extend the 1.8 specification to make it more flexible in the definition of transitions for both Hodgkin-Huxley style and kinetic scheme channel models. Lyle Graham showed how channels are defined in SurfHippo using a similar transition structure to that in 1.8 and functions with a dummy argument for the voltage.

As with other parts of the meeting there was extensive discussion over the use of domain specific terms. At one extreme a channel model could be represented as a generic mechanism governed by differential equations. At the other it could use a variety of commonly used terms,

such as “sigmoid” or “linoid” transitions and set only their parameters. In that case, all such terms would require external definitions to specify the corresponding expressions. The conclusion was that the terms “Channel”, “State”, and “Transition” should be retained, but that there should be no pre-defined terms for particular types of transition equation. Instead, transitions should be expressed by referring to parameterized functions using dummy arguments for the voltage or calcium concentration as in SurfHippo. There was general consensus that channel descriptions should migrate to state based kinetic schemes, maintaining support for older forms such as HH.

Two biological constraints that could influence channel definitions were raised: first all transition are reversible, and second, channels represented by schemes with loops must satisfy microscopic reversibility. The absence of one-way state changes reflects both the semantics of the word “Transition” and the practicalities of avoiding unnecessary duplication in definitions since, for example, a thermodynamic transition model uses one set of parameters to define both forward and reverse rates. It was decided to include both forward and reverse rate specifications within each transition specification. On microscopic reversibility, it was concluded that such constraints should be external to the channel specification itself and could instead be implemented by simulators, or, eventually, by software for fitting channel models to data.

Three choices were considered for specifying functions: custom structures, MathML and inlined expressions. The benefit of MathML is that it provides independently defined terms (for example, in using inlined functions it would be necessary to document whether powers are expressed by “^”, “**” or “pow”). The disadvantage is that it is a large specification and only parts would be needed. It would therefore be necessary to specify which constructs are allowed in the NeuroML context. The conclusion was to continue the use of inlined expressions in the short term but move towards an agreed subset of MathML. It was also agreed that, since only a relatively small set of functions are commonly used for channel transition rates, these should be provided in a standard library which could then be referenced from a model to avoid duplication and the possible errors that could arise from having the same function separately defined in each model. The case of channels with tabulated rates that are interpolated at runtime (GENESIS “tabchan”) raised concerns because of the way the table resolution affects simulated channel behaviour. The decision to use functions allows for the possibility of expressing values obtained by interpolation or splines. The rest can be left to the user and the simulator.

Moving beyond voltage gated ion channels, it was noted that the same formalism will work for ligand-gated channels by allowing the ligand concentration to be a dummy argument in the functions that set transition rates. The connection between a ligand referenced in a channel model and the internal model that controls the ligand concentration requires knowledge of the channel position and, possibly the 3-d structure of the cell. For compartmental multi-shell calcium models, the channel just needs access to the concentration in the outer shell, but, with the development of mesh-based reaction-diffusion simulators, finer spatial scales will become important. In general, a channel should be situated in a membrane separating two volumes and needs consistent definitions of inside and outside.

Although the general case involves 3-d reaction diffusion, many current models use the same simplified forms for internal calcium dynamics. It could therefore be of interest to support simple phenomenological models for ligand concentrations. For more general reaction-diffusion schemes the reactions could be defined with SBML or CellML, adding diffusion constants for each species. Avrama Blackwell's NeuroRD software uses an alternative specification which includes diffusion constants but is more compact than SBML and intended to be human readable and writable. As such it is closer to the current style of NeuroML than SBML is. Although spatial reaction simulators exist, SBML does not yet support geometry specification because of the lack of commonality between simulators. Whatever solution is adopted, it will require structures for linking channel models to concentrations from the reaction system. This is a specific instance of a more general requirement within NeuroML for a flexible standardized way to connect concepts between model specifications.

ChannelML at present includes structures for specifying integrate and fire mechanisms. These do not fit within the kinetic scheme picture, or indeed even a differential equation scheme because of the discontinuous change to membrane potential. It was agreed to move integrate and fire processes elsewhere, but the need to be able to express other membrane processes such as pumps remains. These could be represented as reaction schemes or as state variables and differential equations. In either case, a mechanism is required for specifying how the state of such a component is used in other parts of the model. For ion channels this is currently implicit, via the specification of the permeant ion and the conductances of the states but should be made more explicit, either in the documentation or in some more formal structure. It was agreed to pursue ChannelML development in two stages: first with channel models as described above, then with the development of more general and extensible structures to accommodate pumps and other membrane processes. A Channels Working Group will be set up and will work further on these issues.

Simulator independent descriptions of synaptic mechanisms *(Moderator: Angus Silver)*

The current version of NeuroML supports a number of synaptic mechanisms, including fixed (double exponential/alpha/AMPA, voltage dependent NMDA), plastic (short-term depressing/facilitating synaptic plasticity, spike timing dependent plasticity) and electrical (at gap junctions). These are based on a set of widely used mechanisms allowing setting of the parameters for fixed existing models, rather than providing a framework for creating more flexible models. There is a need for a language that supports more biologically accurate models including concepts such as stochastic descriptions of release and postsynaptic desensitization. Angus presented an overview of the physiology behind such synaptic mechanisms. It was generally agreed that pre and postsynaptic elements of the synapse should be made explicit. The possibility of describing synaptic models of plasticity in a kinetic state based framework was discussed but it was noted that modelling of synaptic behaviour is at an early stage and converting existing models into a common state based framework is unexplored territory and possibly more suited to a dedicated research project. However, the option of a synaptic model with elements of behaviour described by an SBML model was acknowledged as being attractive. It was noted that detailed models of plasticity will raise a number of the same issues as distributed membrane conductances and their interaction with subcellular species. There was a suggestion that the concept of an input event/spike should be formally defined and the behaviour of the synapse described in terms of this. This may also allow for various generalizations of the synapse concept in the future (possibly passing more information with the event), but this may depend on support for such concepts in the simulators.

The question was raised whether synaptic mechanisms should remain in ChannelML or move to SynapseML and there was a consensus on moving to SynapseML, as many of the concepts which will need to be supported are unique to synapses (input events, pre/post synaptic division). If common elements are needed between this and ChannelML (e.g. MathML function calls, SBML support), some importation chain for files can be used.

The distinction was made between an explicit list of synaptic locations and a concept of synaptic density. This concept is not supported in the language at the moment, but may need consideration in the future (would be linked to NetworkML descriptions).

These issues will be discussed further and a new set of elements for describing synapse mechanisms agreed in the Synapses Working Group described below.

Implementation issues *(Moderator: Padraig Gleeson)*

This part of the meeting focussed on issues related to implementation support for this and future versions of NeuroML. The three issues originally on the agenda were: support for

"recommended" implementation specific information (e.g. table sizes for rate equations, etc.); practical issues of interaction with SBML; possibility of common libraries/API for reading & writing NeuroML. The discussions ranged over a number of these and related issues.

On the question of what aspects the simulators should always decide for themselves as opposed to being given in the NeuroML file (e.g. spatial discretisation), there was general agreement that a minimum of information should be given on these aspects as the simulator itself should be depended on to judge the best discretisation (note that these concepts can be very simulator dependent, and so it shouldn't be up to a simulator independent standard to give recommendations to all possible simulators). What is important is that there should be criteria worked out for determining whether a model is behaving "the same" on multiple simulators. There were also discussions on reproducing the same simulation results, use of random number seeds, etc. across multiple runs on same simulator. It was pointed out that some information on seeds, discretisation etc. can be included in the non standard, free form notes in a NeuroML file, with a view to enabling exact reproduction of the model on a given simulator.

There were also questions on how one can know a model translated to NeuroML (say from an NMODL file) still represents the same model. This may just require careful testing of the model until people start using NeuroML as their primary model development format. It was also suggested that tools/simulators should be able to export NeuroML and then import back and produce same results. As part of enabling comparison of models between simulators we may need a suite of tools for comparing waveforms (implementing metrics such as the phase plane method used in Neurofitter). While this is not the core task of the NeuroML initiative, as more models become converted, a set of recommended tools and protocols could be decided on for comparison of models between simulators (cf. SBML Test Suite).

Interaction with SBML: This was also discussed in a number of previous sessions. Nicolas Le Novere agreed with defining pieces of models in NeuroML and SBML with sharing of variable names and model names. It was pointed out that any practical solution for NeuroML/SBML interoperability will depend heavily on SBML support in existing neuronal simulators. Both MOOSE and NEURON are working on greater SBML support.

Options for common API: the use of libSBML is a key reason for the large number of applications, which can read and write SBML in a consistent way. A "libNeuroML" would take care of many things in a similar way for NeuroML. Many lessons could be learned from the implementation of libSBML, including their use of C++ as the core of the implementation, and use of swig for bindings to other languages, e.g. Python, Perl, etc. Time constraints will determine whether such a package could be developed for NeuroML v1.x, but it would be good to incorporate such an API into the planning for v2.0. Some prototyping for such a package can take place already, potentially based on existing parsers which have been developed (e.g. parser for Level 1-2 cells in NEURON, parsers in development for MOOSE, parser for NetworkML in neuroConstruct distribution).

Another point raised was that some people felt uncomfortable with the current use of Levels, e.g. the numbering not reflecting an increasing spatial scale. Also, an application claiming "I support MorphML" or "I support ChannelML" may be more natural than "I support Level X". With a restructuring of some areas of the language for v2.0 (e.g. splitting SynapseML from ChannelML), the naming/use of Levels can be reassessed.

Need for ProtocolML

During the course of the workshop, there was a discussion concerning the need for "ProtocolML", which would provide a standard for describing the elements needed to go from a model description to a simulation description (e.g. simulation duration, inputs to apply, time step, etc.). This could be useful for testing models with different simulators. A language is being

developed in the SBML field (SED-ML: http://www.ebi.ac.uk/compneur-srv/sed-ml), which can take a description in SBML (the model) and add the extra data required for running it under various conditions (the simulation). This could potentially be used for running NeuroML models (or a simpler NeuroML specific version developed). Hugo Cornelis pointed out that Neurospaces has something like this also. This may help NeuroML to describe simulation protocols for reproduction, for figures in papers, etc.

neuroConstruct will put some extra metadata at the top of the file (e.g. simulation duration, time step, what to plot) when a project is exported to NeuroML Level 3 to support reloading the project, and this information could potentially be reused by other applications loading the file. Such a solution may be adequate until a more concrete simulation description language is formalized.

Future plans

Presently the core NeuroML team will consist of Angus Silver, Sharon Crook, Padraig Gleeson, and Robert Cannon with plans for additional funded members starting in the fall of 2009. These will have overall responsibility for implementing and testing the NeuroML specifications, maintaining the website and validator, organizing workshops and other events, and obtaining funding specifically for coordinating the further development of NeuroML.

Meeting participants agreed that specific, more focused Working Groups should continue discussing the issues raised during this workshop and create a draft of new proposed NeuroML structures by July 2009. These drafts will be distributed to the wider community for comments and debate prior to the CNS09 meeting in Berlin. After community feedback is incorporated into these documents, they will become the core of NeuroML version 2.0. Beginning in fall 2009, coordinated updates will be made to the schemas, simulator mappings, tools, website, documentation and core example models in order to transition to NeuroML version 2.0.

Morphology and Channel Distributions Working Group: Padraig Gleeson, Sharon Crook, Robert Cannon, Hugo Cornelis, Nathan Lepora, Henrik Lindén, Alfredo Rodriguez

Channels Working Group: Robert Cannon, Padraig Gleeson, Avrama Blackwell, Lyle Graham

Synapses Working Group: Angus Silver, Guy Billings, Andrew Davison, Michele Mattioni

Workshop Participants

Giorgio Ascoli	Krasnow Institute for Advanced Study, USA
Dave Beeman	University of Colorado, USA
Sandra Berger	Arizona State University, USA
Guy Billings	University College London, UK
Avrama Blackwell	George Mason University, USA
Robert Cannon	Textensor Limited, UK
Hugo Cornelis	University of Texas Health Science Center, San Antonio, USA
Sharon Crook	Arizona State University, USA
Erik De Schutter	Okinawa Institute of Science and Technology, Japan
Andrew Davison	Centre National de la Recherche Scientifique, Gif sur Yvette, France
Padraig Gleeson	University College London, UK
Lyle Graham	Centre National de la Recherche Scientifique, Paris, France
Sean Hill	EPFL, Switzerland
Michael Hines	Yale University, Connecticut, USA
Nicolas Le Novere	European Bioinformatics Institute, UK
Nathan Lepora	University of Sheffield, UK
Michele Mattioni	European Bioinformatics Institute, UK
Henrik Lindén	Norwegian University of Life Sciences, Norway
Ivan Raikov	Okinawa Institute of Science and Technology, Japan
Subhasis Ray	National Center for Biological Science, India
Arnd Roth	University College London, UK
Angus Silver	University College London, UK
Koen Vervaeke	University College London, UK