



## Summary of meeting to discuss SBML interactions in NeuroML version 2.0

**Location:** European Bioinformatics Institute, Hinxton, UK  
**Date:** 19 May 2010

**Attendees:** **EBI:** Nicolas le Novère, Michele Mattioni, Sarah Keating  
**Textensor Limited:** Robert Cannon  
**Caltech:** Mike Hucka  
**Arizona State University:** Sharon Crook (remotely via EVO)  
**UCL:** Padraig Gleeson

### Motivation for meeting

It is becoming increasingly clear that models in computational neuroscience can benefit from including models of biochemical reactions, and that computational neuroscientists need to interact more closely with the systems biology community. The most productive way of interacting will involve the exchange of models between computational neuroscientists and systems biologists and therefore requires the use of common model description languages.

One of the possible extensions to NeuroML version 1.x which was highlighted was the ability for neuronal models expressed in this format to include detailed subcellular interactions (potentially influenced by membrane potential, and influencing in turn distributed or synaptic channel properties). Such reaction schemes can be expressed in languages like SBML and CellML, and NeuroML version 2.0 will feature greater interaction with model components in these languages. SBML has been identified as the language of initial study towards this goal due to support for this language in the MOOSE simulator (<http://moose.sourceforge.net>) and support for SBML components in NEURON, which has been made possible by mapping SBML files to the NMODL language (for more see [http://www.neuroml.org/neuron\\_tools.php](http://www.neuroml.org/neuron_tools.php)).

This meeting was arranged following discussions at the second annual NeuroML Development Workshop at Arizona State University on March 4th & 5th 2010 (<http://math.la.asu.edu/~cans/workshop.html>) and focussed on the technical implementation of SBML interaction in NeuroML v2.0 as well as providing a forum for getting input from the SBML developers on widening the community processes of the NeuroML initiative.

### Discussion Points

#### Summary of NeuroML & SBML status

A brief status report of NeuroML and SBML were presented. NeuroML v1.8.1 is the latest stable version of the language. Currently supported simulators include NEURON, GENESIS, MOOSE, PSICS and PyNN based simulators, with support in a number of other tools in active development (see [http://www.neuroml.org/tool\\_support.php](http://www.neuroml.org/tool_support.php)). This version of the language, together with a number of new cell and network models is described in a paper which has just been published in PLoS Computational Biology (Gleeson et al. 2010, NeuroML: A Language for Describing Data Driven Models of Neurons and Networks with a

High Degree of Biological Detail). NeuroML version 2 was discussed at the NeuroML Development Workshop at ASU recently, and the minutes of that meeting discuss in detail the range of cell and channel/synapse models which will be covered by the new version of the language (available at <http://www.neuroml.org/files/NeuroMLWorkshop2010.pdf>). Following the meeting, a number of the example NeuroML v2.0 files were updated with the suggested structures for elements in the new version of the language, and these served as the basis of discussions at this meeting.

Development of the core of SBML Level 3 is well advanced, while the individual packages are proceeding at a variety of paces. The recent SBML/BioModels.net Hackathon ([http://sbml.org/Events/Hackathons/The\\_2010\\_SBML-BioModels.net\\_Hackathon](http://sbml.org/Events/Hackathons/The_2010_SBML-BioModels.net_Hackathon)) was an opportunity for the community to get up to date with the various activities taking place, and information on a number of projects is available at the above link.

## Hooks in NeuroML version 2 to model elements in SBML

A typical scenario for the interaction between model elements in NeuroML & SBML was:

- an extracellular drug (e.g. 4-AP to block K<sup>+</sup> channels) whose concentration varied with time (as described in an SBML model)
- a K<sup>+</sup> channel whose conductance goes to zero for high 4AP (specified either in "ChannelML" v2.0 or SBML)
- a NeuroML model containing a cell with K<sup>+</sup> channels and the blocker which links the two

It was agreed that in this case the SBML file (e.g. KBlocker.xml) for the extracellular concentration should have a species **4AP\_src** (or potentially just a parameter) which can be read by the rest of the NeuroML model, but whose value is solely determined by the SBML description of its dynamics. The executing simulator should run this as a normal SBML model but make the **4AP\_src** variable accessible from outside. This model can be imported into the NeuroML description in the following way:

```
<reactionScheme id="KChannelBlocker" source="KBlocker.xml" type="sbml">
  <variable id = "4AP_nml" type = "species">
    <metadata/>
  </variable>
  <mapping intRef="4AP_nml" extRef="//species[@id='4AP_src']" direction="in"/>
</reactionScheme>
```

The **extRef** attribute points to a variable in the SBML document (using XPath or a subset thereof). The blocker reaction is placed in the extracellular medium using:

```
<extracellularProperties id = "extracellular">
  <biochemistry reactionScheme = "KChannelBlocker"/>
</extracellularProperties>
```

The Na<sup>+</sup> channel can be expressed as a "ChannelML" version 2 file with a parameter **4AP\_tgt** which can be set from outside (i.e. whose value is not determined by the channel model description), or as a "pure" SBML file with a **4AP\_tgt** parameter (which is not the variable changed in any of the rateRules). The main NeuroML model is responsible for linking all of these elements. The proposed structure for this had the following form (assuming the channel is expressed in an SBML file in KChan\_SBML.xml):

```
<reactionScheme id = "KChan_ext" source = "KChan_SBML.xml" type = "sbml">
  <mapping intRef="neuroml:chan_curr" extRef="//parameter[@id='i']" direction="in"/>
  <mapping intRef="neuroml:memb_pot" extRef="//parameter [@id='v']" direction="out"/>
  <mapping intRef="/reactionScheme[@id='KChannelBlocker']/variable[4AP_nml]"
    extRef="//parameter[@id='4AP_tgt']" direction = "out"/>
</reactionScheme>
```

i.e. the model exposes a variable for channel current based on the parameter **i** in the SBML file, the parameter **v** is taken from the NeuroML model and set in the SBML model component, as is the SBML parameter **4AP\_tgt**, which is set to be the value **4AP\_nml** from the NeuroML model.

A possible extension of this mechanism will allow manipulations of the values read from (or given to) the SBML model, e.g.

```
<mapping intRef="ca_tot" direction="in" >!--adds 2 params found in the SBML model-->
  <extRef>
    <lambda>
      <bvar>
        <ci definitionURL = "//species[@id='ca_cyt']">ca_cyt</ci>
      </bvar>
      <bvar>
        <ci definitionURL = "//species[@id='ca_mit']">ca_mit</ci>
      </bvar>
      <plus/>
      <ci>ca_cyt</ci>
      <ci>ca_mit</ci>
    </lambda>
  </extRef>
</mapping>
```

which would add 2 parameters in the SBML model and make only the sum visible in the NeuroML model. This scheme could also be useful for transforming units between the different languages.

There is much still to work out about the proper form of these elements, not least of which the handling of the **<variable>** element (is it related only to the **<reactionScheme>**, is it global, etc.). Also ensuring a format which is flexible enough for multiple instances of SBML models (e.g. one for each synapse) or potentially a simulator specific discretisation of model instances (one internal instance of SBML model for conductance on each subsection of dendrites) will be essential.

The example NeuroML files containing these elements (for example: [http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2\\_Test/web/NeuroMLFiles/FullNeuroML\\_SBML.xml?view=markup](http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2_Test/web/NeuroMLFiles/FullNeuroML_SBML.xml?view=markup)) will be updated as the scheme is worked out. It is hoped that this format will be independent of the language of the imported model component (e.g. CellML or a NineML model component).

## Usage of units

At the NeuroML Workshop in March, it was agreed that where physical quantities are required in NeuroML files (e.g. to specify a conductance density, specific capacitance, etc.) that information on the dimensions of the units used should be given each time a value is specified (as opposed to the use of SI Units & Physiological Units in a wrapping element in NeuroML v1.x). Some shortcut may be required when many instances of the same units are used, e.g. not requiring  $\mu\text{m}$  each time a length is used in a morphology file.

The format for specifying the units in the draft NeuroML version 2 files was discussed. The option of completely free specification of new units (e.g. "0.03 mS cm<sup>-2</sup>" where only Siemens and meters are known in advance) was agreed to put too much work on to the parsing application. A better option was to allow predefined strings for new unit types (in terms of a core set, e.g. mSpercm<sup>2</sup>) which could be used as needed when defining a value. The option of presenting the value and unit information together in a single attribute (e.g. "0.03 mSpercm<sup>2</sup>") was not very popular. One option presented was:

```
<channelPopulation ionChannel="NaConductance" segmentGroup="soma_group">
  <conductanceDensity value = "0.03" units = "mS_percm2"/>
</channelPopulation>
```

The string "mS\_percm<sup>2</sup>" could be defined earlier in a structure similar to the **<listOfUnitDefinitions>** in SBML. A number of the most commonly used units (e.g. for

conductance (density), specific capacitance and axial resistance, in both SI & Physiological units) could be defined in a standard NeuroML units imports file, meaning most cells would be able to use the standard strings for specifying units, while also retaining the option to override them at any point.

The example NeuroML v2.0 files will be updated with some of the above proposed schemes, and input sought on the final form from the wider community.

## Libraries for language usage (libNeuroML)

The usefulness of the libSBML package for reading and writing SBML, and checking for validity beyond basic XML Schema checks, was reemphasised. Such a package would be useful for ensuring wider support for NeuroML version 2. A basic version of such a library in C++ has been developed as part of the MOOSE project (for the latest details on this see <http://neuroml.svn.sourceforge.net/viewvc/neuroml/TestLibNeuroML/v0.2/> and for MOOSE usage see [http://moose.svn.sourceforge.net/viewvc/moose/moose/trunk/neuroML\\_IO/](http://moose.svn.sourceforge.net/viewvc/moose/moose/trunk/neuroML_IO/)). A Java based API generated from the v1.x Schema documents is also currently available (<http://sourceforge.net/projects/neuroml-api>) but this is not regularly updated.

The libSBML approach of having the core of the package defined in C++ and generating other language bindings (e.g. Java, Python) from this via SWIG is an attractive approach for libNeuroML v2.0. This should not however preclude native APIs in other languages such as Java (which would allow greater portability of the library in contrast to the SWIG generated Java), which is starting to happen in SBML. Development of such libraries will however most likely require at least one dedicated person to assume the role of lead developer. While such there is some interest for this in the labs of Upi Bhalla & Sharon Crook, there is currently no one ready to take on that role in the longer term.

A basic Java API for generating NeuroML v2.0 example files is available at: [http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2\\_Test](http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2_Test). This is mainly intended for demoing ideas for v2.0, and may or may not develop into a more stable library. It should be useful in the short term though for investigating the feasibility of the proposed structure of the new version of NeuroML.

## Annotations

The SBML community has a much longer experience of dealing with structured annotations than NeuroML, and infrastructure and guidelines is already in place to encourage good practice in annotating models with machine readable metadata. A good argument was made for the use of MIRIAM standards for annotation of model components. A unique string can be used to identify an entity in a database which is registered on the MIRIAM Resources website, e.g.

```
<cell id = "SpikingCell" metaid = "HippoCA1Cell">
  <annotation>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
      <rdf:Description rdf:about = "HippoCA1Cell">
        <bqbiol:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource = "urn:miriam:neurondb:258"/>
          </rdf:Bag>
        </bqbiol:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
  ...

```

This annotation will identify the cell as a CA1 Pyramidal cell as defined by NeuronDB. A simple transformation (e.g. via the web service at <http://www.ebi.ac.uk/miriam>) will link to the URL at <http://senselab.med.yale.edu/NeuronDB/ndbEavSum.asp?id=258> giving the definition of this cell.

This general scheme will allow links to a wide range of other databases for giving information on structures in NeuroML files, for example:

NeuroMorpho.org	Neuronal reconstruction source files
BioModels database	Original SBML model component
PubMed	Publication on (original) model
Chemical Entities of Biological Interest (ChEBI)	Database of small molecules relevant for biological processes
Gene Ontology	Controlled vocabulary to describe gene and gene product attributes in any organism
ModelDB*	Original model scripts
NeuroLex*	A dynamic lexicon of neuroscience terms

\* yet to be added to MIRIAM Resources

This general scheme could replace the set of elements like **<meta:publication>** and **<meta:neuronDBref>** currently in NeuroML. It is a good indicator that both the SBML and CellML communities are using this format for their metadata. If researches publish their models with this type of structured information, automated searches of the following type could be made possible:

*"Get me all models of cerebellar (urn:miriam:neurolex:...) cells containing potassium channel type X (urn:miriam:GO:...) along with all relevant publications"*

A similar structure to the MIRIAM suggested format for creators/history of the model could also be used in place of **<meta:authorList>**, **<meta:modelTranslator>** etc.

## Process of creating specification for version 2.0

The process for creating specification documents for the next version of NeuroML was also discussed. The SBML process of elected editors was considered, and one requirement for such a system in NeuroML would of course be more widespread involvement in the language development process (compare ~300 subscribers to the main SBML mailing list with ~40 to NeuroML's main list). It was agreed that formal editors would be useful in the long term but that a few steps prior to a draft specification for v2.0 are required before that stage (see below).

## Next Steps

The draft NeuroML v2.0 examples will be continuously updated at: [http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2\\_Test/web/NeuroMLFiles](http://neuroml.svn.sourceforge.net/viewvc/neuroml/DemoVer2.0/NMLVer2_Test/web/NeuroMLFiles). The revision which reflects the agreements and suggestions made during this meeting and discussed in this document is SVN revision 410.

As regards wider input from the community on the proposed NeuroML v2.0 examples it was agreed that input should be requested on individual sections of the language in turn. As the morphology specification is most stable at the moment, this will be the first to be sent around to the NeuroML mailing list in the next few weeks to try to stabilise the relevant elements there and get agreement on the scope of morphological descriptions in v2.0.

This will be followed by ion channel specifications, synapses and biophysical specifications, as each of these become more stable (the updated representation of network structure will evelove in line with the INCF Task force on multiscale modelling). Once there is agreement within the NeuroML community on the scope and format of each part of the language, a more formal specification document can be created for input from the wider neuroscience community (e.g. via the comp-neuro mailing list). At this stage formal NeuroML editors will have to be in place.