Extended Summary:

Simulator-independent representation of ionic conductance models with ChannelDB

David Beeman and James M. Bower

Simulation packages such as GENESIS (Bower and Beeman, 1968), NEURON (Hines and Carnevale, 1997), and other systems that were developed for structurally realistic neural modeling provide the means to exchange and collaborate on the development of models by providing standard environments for the construction of simulations. However, each simulator has its own scripting language for the construction of simulations, and these are sufficiently different to make the exchange of models or model components between simulators very difficult. This is because the simulation scripts are largely procedural programs that tell the simulator how to construct a model, using the tools and basic model components of the simulator, rather than a purely declarative description of the model.

In order to facilitate the exchange of neuronal model descriptions in a simulator-independent format, we developed a declarative XML-based format that has become the basis for the NeuroML model description language (Goddard, et al., 2001). The Modelers Workspace (MWS) project (Hucka, et al., 2002) is a collection of software tools to facilitate the collaborative construction of models via the WWW, using NeuroML as the format for the description of models and data. ChannelDB, a database of ionic conductance models, is an implementation of one of the first core components of the MWS. At present, ChannelDB is implemented as a stand-alone module, with its own graphical user interface to the database. After further development, the ChannelDB GUI will be merged into the MWS.

This paper describes the representation used in ChannelDB, and is intended to generate discussion and collaboration for the further development and extension of this representation for a wider variety of conductance models and simulators. The listing below gives an example of a NeuroML representation of a simple Hodgkin-Huxley "squid-like" potassium conductance. This is a declarative representation that describes everything about the model, not a procedural description of how to implement the model.

```
<neuroml class="DBChannel" description="Hodgkin-Huxley squid K channel"
    author="Dave Beeman"
      keywords="Hodgkin-Huxley potassium squid delayed rectifier"
      uniqueID="10262778758662F22@dogstar.colorado.edu"
      notes="An implemention of the GENESIS K_squid_hh channel" Erest="-0.07V">
   <channels>
      <channel name="K_squid_hh" class="HHChannel" permeantSpecie="K"
           Erev="0.09V" Gmax="360.0S/m^2" ivlaw="ohmic">
        <gates>
           <gate name="X" class="HHVGate" timeUnit="sec" voltageUnit="V"
```

```
          vmin="-0.1" vmax="0.05"
           instantCalculation="false" useState="false" power="4">
        <forwardRate class="ParameterizedHHRate" A="-600.0"
             B="-10000.0" C="-1.0" D="1.0"
             E="0.060" F="-0.01"/>
        <backwardRate class="ParameterizedHHRate" A="125.0" B="0.0"
            C="0.0" D="1.0" E="0.07" F="-0.08"/>
      </gate>
    </gates>
     <log author="Dave Beeman" date="Jul 9, 2002 11:11:15 PM"
         literatureReference="A.L. Hodgkin and A.F. Huxley, J.
          Physiol. (Lond) 117, pp 500-544 (1952)">
      <logEntries>
      </logEntries>
    </log>
  </channel>
 </channels>
</neuroml>
```

In this example, the K_squid_hh channel is derived from the HHChannel class, which has certain properties such as a reversal potential Erev, and a conductance density, given in units of Siemen per square meter. It also possesses a voltage activated Hodgkin-Huxley gate, derived from the HHVGate class. The gate contains objects representing the forward and backward rate variables. In this case, the equations for the rate variables are represented with a parameterized form, derived from the ParameterizedHHRate class. A TabulatedHHRate class is provided to represent the rates with with tabulated values, and an EquationHHRate is used to represent the rate with an equation. Other types of conductances, such as calcium-dependent channels, make use of classes for concentration-dependent gates, concentration pools, and current sources.

The NeuroML development kit parser (from http://www.neuroml.org) is used to convert between model representations in the form of Java objects, and the NeuroML representation. NeuroML description files are generated by creating Java objects that represent the channel description. In our implementation, the NeuroML representations are stored in a searchable database, which is imlemented with MySQL. However, there is no requirement to use any particular database implmentation or user interface, nor to even use a database at all.

After a model is selected from the database, the NeuroML parser is used to convert the representation back into Java objects. It is then a relatively simple task to use Java string manipulation tools to convert the representation into a simulation script for the desired simulator. This has been done for GENESIS, and efforts are underway to extend this to NEURON. ChannelDB with all source code and documentation may be downloaded from http://www.modelersworkspace.org.

References:

Bower, J. M. and Beeman, D. (1998). The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System, second edition. Springer-Verlag, New York.

Goddard, N., Hucka, M., Howell, F., Cornelis, H., Shankar, K. and Beeman, D. (2001) Towards NeuroML: Model Description Methods for Collaborative Modelling in Neuroscience, Philosophical Transactions of the Royal Society B. 356:1209-1228.

Hines, M. and Carnevale, N. T. (1997). The NEURON Simulation Environment. Neural Computation 9:1179-1209.

Hucka, M., Shankar, K., Beeman, D., and Bower, J. M. (2002). The Modeler's Workspace: Making model-based studies of the nervous system more accessible. In Computational Neuroanatomy: Principles and Methods, G. Ascoli (Ed.), Humana Press Inc.