XML representation of neuron for multi-scale modelling

Dilawar Singh

October 1, 2013

Abstract

This document specify an XML format for representing neurons for the purpose of multi-scale modelling. Many issues related to multi-scale modelling are discussed in a meeting held in Bangalore in 2009 [**?**].

1 Domains

A **domain** is like a compartment of the cell. A cell is made up of domains. A cell also has a electrical circuit model. A **chemical domain** is a compartment of cell with a well-defined chemical model. It may be possible to classify these domains according to the type of chemical reactions are taking place in them. We can describe each class in XML and use their instances with co-ordinates. An **electrical domain** on the other hand describes the electrical circuit representation of a compartment of the cell. A chemical domain always has a equivalent electrical model or electrical domain.

1.1 Representation of domains

Chemical domains are named `cd_<number>` and electrical domain are named `ed_<number>`. Each cell is divided into two or more chemical domains and equivalent circuit of the cell is divided into two or more electrical domains.

Synaptic activity in chemical domain A chemical domain if it contains synaptic activity, should have an optional attribute i.e. `synaptic_site = ‘‘yes’’`. How would [*ligand*] change with synaptic-activity? A model for such an activity should be embedded in xml.

```
<cell id=‘‘1‘‘ ... >
    <morphology>
       <domains>
          <domain id=‘‘cd_1’’ synaptic_site=‘‘no’’>
            <species>list of species </species>
            <reaction id=‘‘reactionA’’>
              <reactants> .. </reactants>
              <products> .. </products>
              <rate_coeff> 20 </rate_coeff>
            </reaction>
            <reaction>
            .
            </reaction>
          </domain>
          <domain id=‘‘cd_2’’ synaptic_site=‘‘yes’’>
            <!-- specify chemical domain here with information about synaptic site-->
          </domain>

          <!-- Electrical domains here -->
          <domain id=ed_1 >
            <circuit>
               <!-- spice type netlist or matrix representation or graph -->
               <input> list of points which are input to this domain. </input>
               <output> list of points which are output of this circuit. </output>
            </circuit>
            <coordinates>
               <!-- coordinates of cell body which this circuit is an equivalent
                 representation. Does it map over to some chemical domain? -->
            </coordinates>
          </domain>
       </domains>

       <mapping>
         <!-- see section 1.2 for mapping between the domains -->
       </mapping>
   </morphology>
</cell>
```

1.2 Mapping between chemical and electrical domain

Moving borders of compartment If the borders of the compartment are to move, we need to think of standard geometries which can be thought of "enclosing" the compartment.

Any change in chemical composition of a chemical domain (Say `cd_1`) can change the electrical properties of an electrical domain (Say `ed_3`) (and vice versa?). A user-defined mapping has to be provided in the model. [1].

Such a mapping can be encoded by a graph by the application. XML based graph representation such as **graphml** are available and can be used by application to serialize the graph. Directly writing the mapping in graphml format will be less user-friendly than having a more readable XML element to describe such mapping.

How the change in one domain changes properties of other domain are to be described by `<mapping>` element. It looks like the following:

```
<mapping id=‘‘1’’ from=‘‘cd_1’’ to=‘‘ed_3’’>
    <relation from=‘‘cd_1’’ to=‘‘ed_3’’>
        <lhs> concCa </lhs>
        <rhs> some_XML_equation_describing_relation_between_[Ca]_and_Ica </rhs>
    </relation>
</mapping>
```

Format of rhs Element rhs should be generic enough to describe mathematical relations. How about using latex math syntax? Some work is needed to parse and build an equation out of it but it should be straightforward.

The library python-networkx has good support for graphml format. Each relation is represented by an edge between two nodes (nodes). We can attach python objects to edges and nodes. For instance a functor which computes how variation in I_{ca} changes $[Ca]$ can be attached to the edge. Graphs are great for non-planner relationships and great many algorithms already exits to analyse topology.

2 Adaptors

Adaptors has to be employed if there is unit-conversion problem. We recommend that one should stick to the standard units making XML representation unit agnostic. Writing adaptor needs reference documentation of **pymoose**.

[1] http://graphml.graphdrawing.org/

3 An example of a neuron

Let say we have a simple neuron. It has two dendrites connected to soma and one axon. Each dendrite and axon has 2 chemical domains.

References