OpenCell Development
(the COR-like view)

Alan Garny
alan.garny@dpag.ox.ac.uk
EDITING OF CELLML FILES CAN BE DONE USING:
- The initial conditions/constants view;
Introduction

- Editing of CellML files can be done using:
  - The initial conditions/constants view;
  - The complete model structure view;
Edi$ng
of
CellML
files
can
be
done
using:

- The initial conditions/constants view;
- The complete model structure view;
- The XML view; or
**Introduction**

Editing of CellML files can be done using:
- The initial conditions/constants view;
- The complete model structure view;
- The XML view; or
- The equations view.
EDITING OF CELLML FILES CAN BE DONE USING:

- The initial conditions/constituents view;
- The complete model structure view;
- The XML view; or
- The equations view.

However, this may not always be the fastest and/or obvious way to edit a CellML file.

Another possible approach is the one taken by COR which relies on a proprietary language.
COR-LIKE VIEW IN OPENCELL

```xml
<model name="hodgkin_huxley_squid_axon_1952"
cmeta:id="hodgkin_huxley_squid_axon_1952"
xmlns=http://www.cellml.org/cellml/1.0#
xmns:cellml=http://www.cellml.org/cellml/1.0#
xmns:cmeta=http://www.cellml.org/metadata/1.0#>
...
<units name="millisecond">
  <unit prefix="milli" units="second" />
</units>
...
<component name="sodium_channel">
  <variable name="i_Na" public_interface="out"
    units="microA_per_cm2" />
  ...
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply id="E_Na_calculation"><eq />
      <ci>E_Na</ci>
      <apply><plus />
        <ci>E_R</ci>
        <cn cellml:units="millivolt">115.0</cn>
      </apply>
    </apply>
  </math>
  ...
</component>
...
</model>
```

CellML DOM API
def model hodgkin_huxley_squid_axon_1952 as
    def unit millisecond from
        unit second {pref: "milli"};
    enddef;
    ...
    def comp sodium_channel as
        var i_Na: microA_per_cm2 {pub: out};
    enddef;
    ...
    E_Na = E_R+115.0{unit: "millivolt"};
    i_Na = g_Na^pow(m, 3.0{unit: "dimensionless"})*h*(V-E_Na);
enddef;
    ...
enddef;
COR-LIKE VIEW IN OPENCELL

```xml
<model name="hodgkin_huxley_squid_axon_1952"
  xmlns="http://www.cellml.org/cellml/1.0#">
  <units name="millisecond">
    <unit prefix="milli" units="second" />
  </units>
  <component name="sodium_channel">
    <variable name="i_Na" public_interface="out" units="microA_per_cm2" />
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply id="E_Na_calculation"><eq />
        <ci>E_Na</ci>
        <apply><plus />
          <ci>E_R</ci>
          <cn cellml:units="millivolt">115.0</cn>
        </apply>
      </apply>
      <apply id="i_Na_calculation"><eq />
        <ci>i_Na</ci>
        <apply><times />
          <ci>g_Na</ci>
          <apply><power />
            <ci>m</ci>
            <cn cellml:units="dimensionless">3.0</cn>
          </apply>
          <ci>h</ci>
          <apply><minus />
            <ci>V</ci>
            <ci>E_Na</ci>
          </apply>
        </apply>
      </apply>
    </math>
  </component>
</model>
```

```python
def model hodgkin_huxley_squid_axon_1952 as
  def unit millisecond from
    unit second {pref: "milli"};
  enddef;
  ...
  def comp sodium_channel as
    var i_Na: microA_per_cm2 {pub: out};
    ...
    E_Na = E_R+115.0{unit: "millivolt"};
    i_Na = g_Na*pow(m, 3.0{unit: "dimensionless"})*h*(V-E_Na);
  enddef;
  ...
enddef;
```
Requirements

- The proprietary language used in COR and in OpenCell should, ideally, be the same.
- The order of a CellML file’s contents should be reflected in the COR-like view.
- The overall behaviour of the COR-like view should be the same as that of the XML view.
The editor used in the COR-like view is very limited (also the case for the XML view).

Differences between the COR-like view and COR:

- Support for reactions and imports;
- Equations are enclosed within a `def math as ... enddef` statement;
- Referencing state variables is done using `d(my_var)/dt` rather than `ode(my_var, time)`;
- Referencing constants is done using `my_const{unit: "my_unit"}` rather than `my_const{my_unit}`;
- Some mathematical functions don't (yet) have the same name (e.g. `pow` rather than `power`);
- `E` rather than `e` for the exponent (?!);
- Nested piecewise statements are supported;
- Groups can be of more than one type; and
- Overall, still far too many unnecessary spaces and parentheses.

Editing of metadata is not currently supported.
CONCLUSION

- A COR-like view has been added to OpenCell.
- It will, however, only be available in OpenCell 0.8.
- Further work includes:
  - Thorough user testing of the COR-like view;
  - ‘Fix’ remaining syntactical issues (e.g. `power` rather than `pow`);
  - Support metadata editing (somehow!); and
  - Use a better editor!

www.opencellproject.org

Randal Britten  
r.britten@auckland.ac.nz

Alan Garny  
alan.garny@dpag.ox.ac.uk

Justin Marsh  
j.marsh@auckland.ac.nz

Andrew Miller  
a.miller@auckland.ac.nz