

# OpenCell Development

*(the COR-like view)*

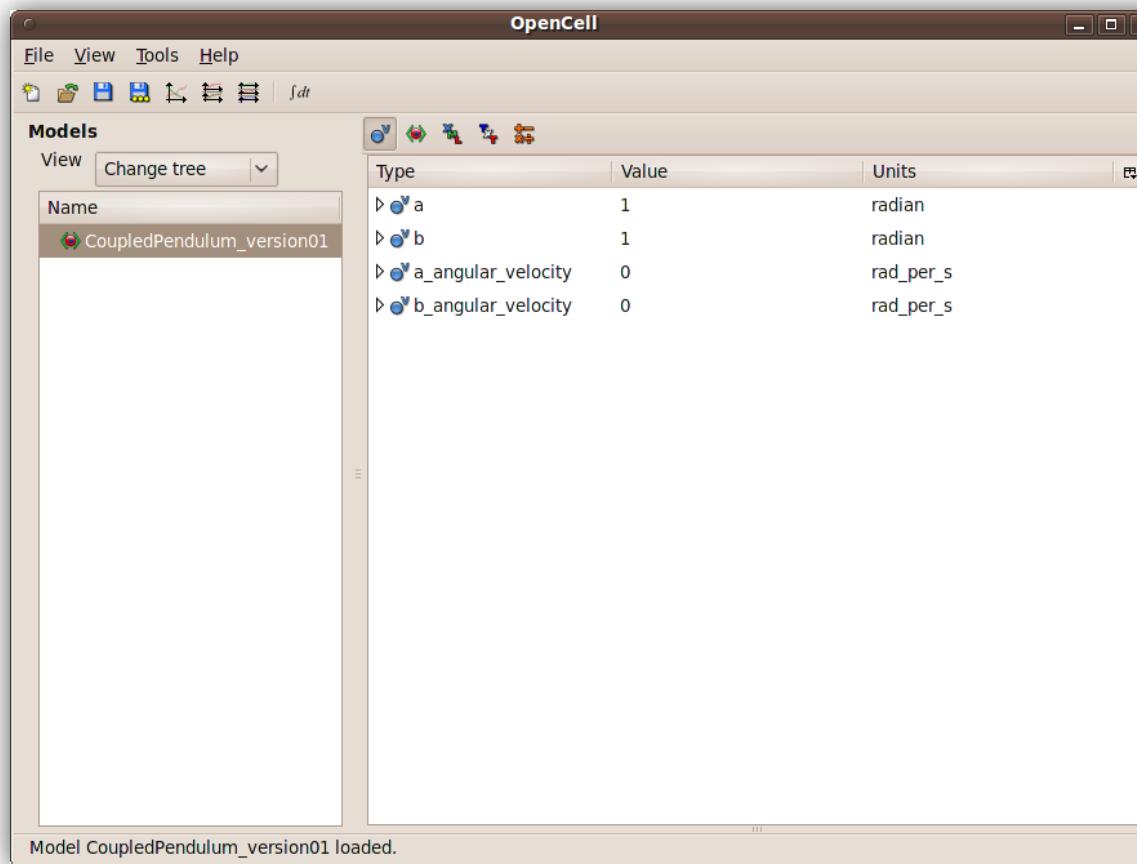
*Alan Garny*  
[alan.garny@dpag.ox.ac.uk](mailto:alan.garny@dpag.ox.ac.uk)



# INTRODUCTION

---

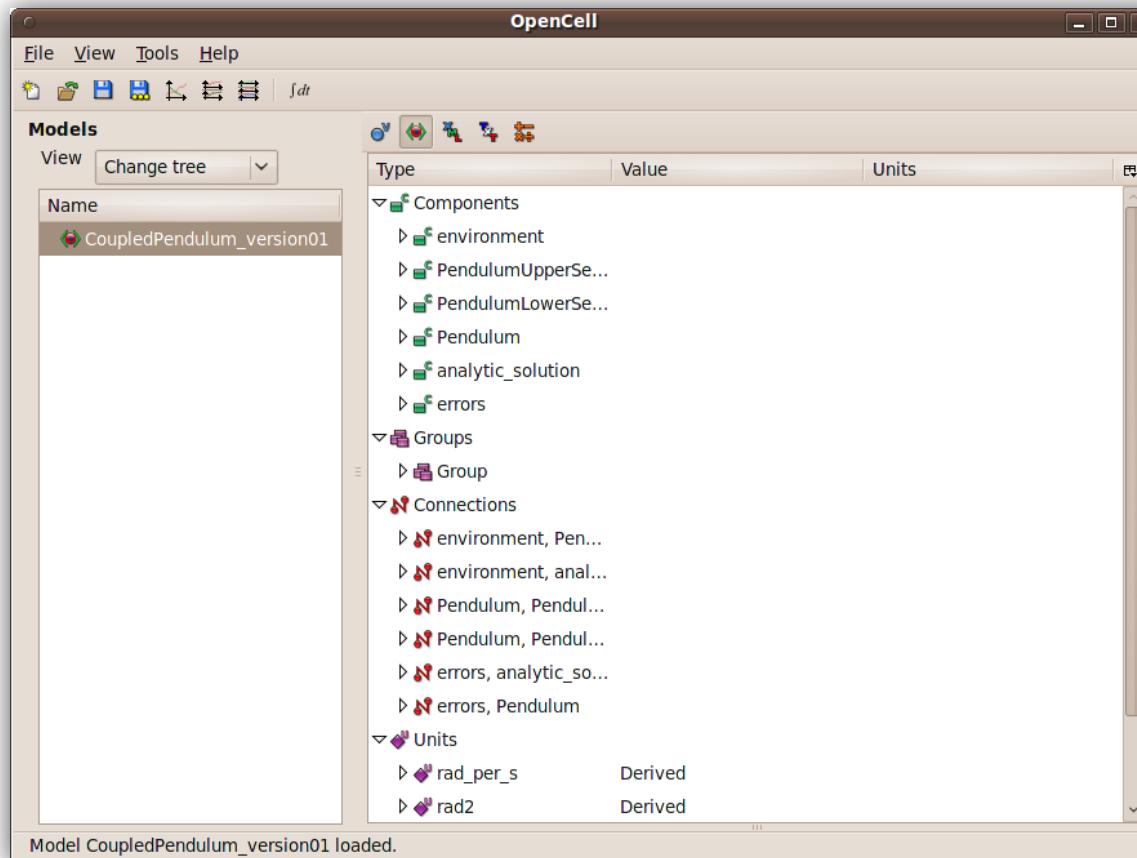
- Editing of CellML files can be done using:
  - The initial conditions/constants view;



# INTRODUCTION

---

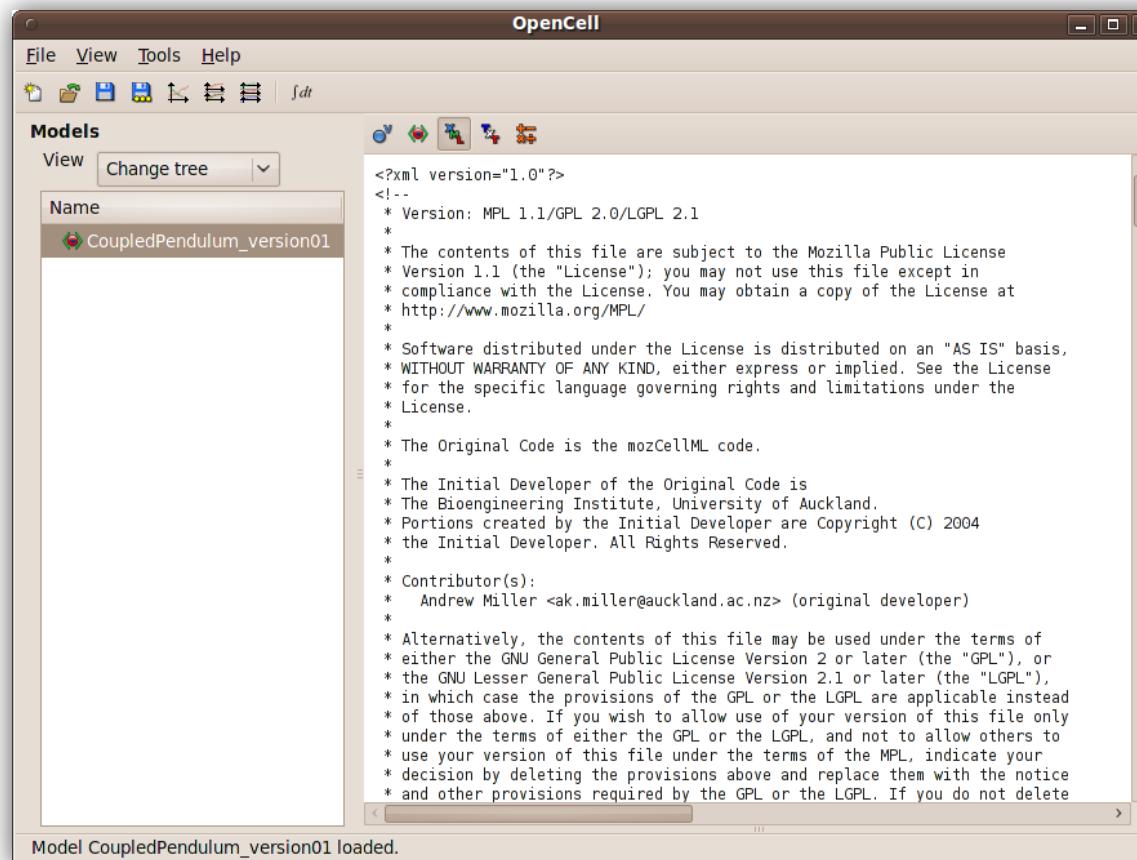
- Editing of CellML files can be done using:
  - The initial conditions/constants view;
  - The complete model structure view;



# INTRODUCTION

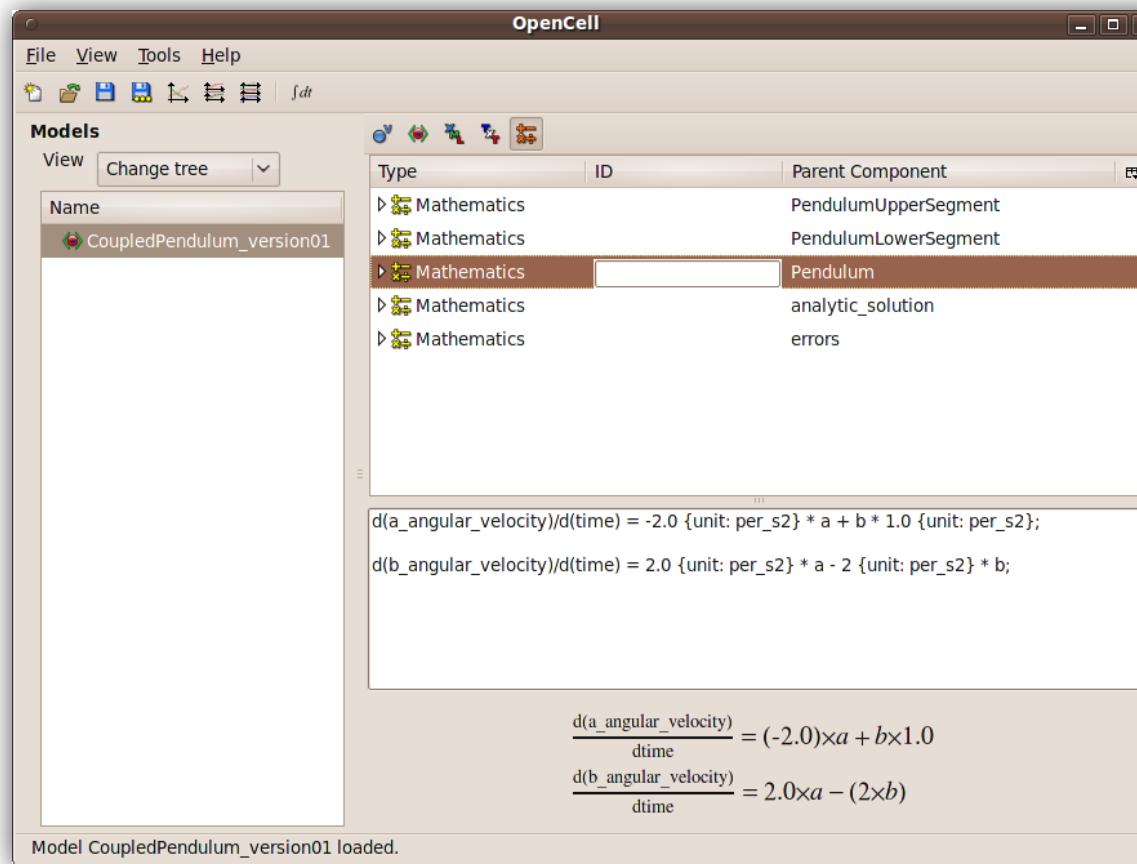
---

- Editing 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.



# INTRODUCTION

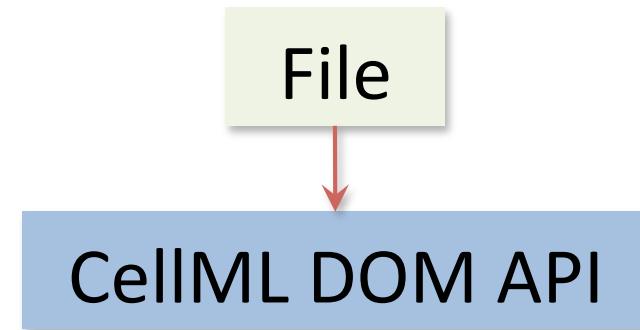
---

- 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.
- 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.

The screenshot shows the COR 0.9 software interface in Editorial Mode. At the top, there's a menu bar with File, Edit, View, Run, Tools, Help. Below the menu is a toolbar with various icons. The main window has a title bar "COR 0.9 [Editorial Mode] - \Documents and Settings\Administrator\Desktop\Models\noble\_1962\_version05.cellml". The left side of the window displays a mathematical equation for the sodium channel  $m$  gate:
$$a_m = \frac{100(-V-48)}{e^{\frac{-V-48}{15}} - 1}$$
Below the equation is the corresponding CellML code for the sodium channel  $m$  gate:def comp sodium\_channel\_m\_gate as  
var m: dimensionless {init: 0.01, pub: out};  
var alpha\_m: per\_second;  
var beta\_m: per\_second;  
var V: millivolt {pub: in};  
var time: second {pub: in};  
  
alpha\_m = 100\*(per\_millivolt\_per\_second)\*(-V-48{millivolt})/(exp((-V-48{millivolt})/15));  
beta\_m = 120\*(per\_millivolt\_per\_second)\*(V+8{millivolt})/(exp((V+8{millivolt})/5));  
ode(m, time) = alpha\_m\*(1{dimensionless}-m)-beta\_m\*m;  
enddef;  
  
def comp sodium\_channel\_h\_gate as  
var h: dimensionless {init: 0.8, pub: out};  
var alpha\_h: per\_second;  
var beta\_h: per\_second;  
var V: millivolt {pub: in};  
var time: second {pub: in};  
  
alpha\_h = 170\*(per\_second)\*exp((-V-90{millivolt})/20{millivolt});  
beta\_h = 1000\*(per\_second)/(1{dimensionless}-exp((-V-42{millivolt})/10{millivolt}));  
ode(h, time) = alpha\_h\*(1{dimensionless}-h)-beta\_h\*h;  
enddef;  
  
def comp potassium\_channel as  
var i\_K: nanoA {pub: out};

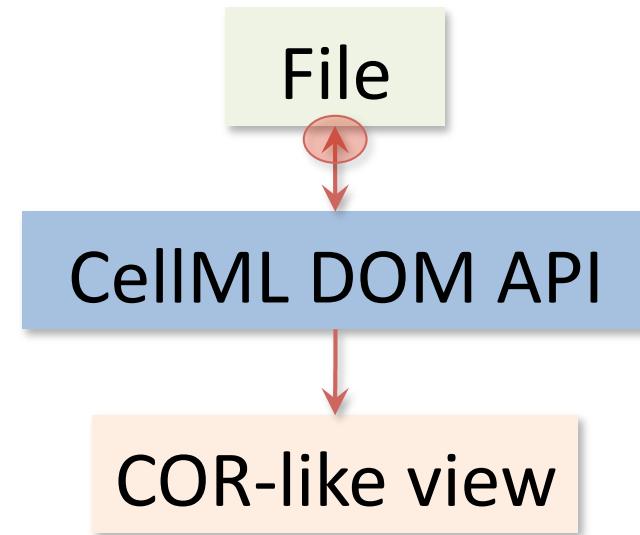
# COR-LIKE VIEW IN OPENCELL

```
<model name="hodgkin_huxley_squid_axon_1952"
  cmeta:id="hodgkin_huxley_squid_axon_1952"
  xmlns="http://www.cellml.org/cellml/1.0#"
  xmlns:cellml="http://www.cellml.org/cellml/1.0#"
  xmlns:cmeta="http://www.cellml.org/metadata/1.0#">
...
<units name="millisecond">
  <unit prefix="milli" units="second" />
</units>
...
<component name="sodium_channel1">
  <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>
```



# COR-LIKE VIEW IN OPENCCELL

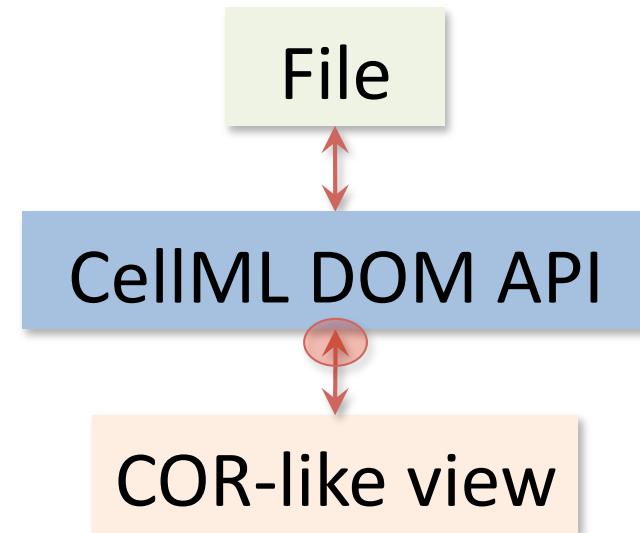
```
<model name="hodgkin_huxley_squid_axon_1952"
  cmeta:id="hodgkin_huxley_squid_axon_1952"
  xmlns="http://www.cellml.org/cellml/1.0#"
  xmlns:cellml="http://www.cellml.org/cellml/1.0#"
  xmlns:cmeta="http://www.cellml.org/metadata/1.0#">
...
<units name="millisecond">
  <unit prefix="milli" units="second" />
</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>
  <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>
```



```
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;
```

# COR-LIKE VIEW IN OPENCCELL

```
<model name="hodgkin_huxley_squid_axon_1952"
  cmeta:id="hodgkin_huxley_squid_axon_1952"
  xmlns="http://www.cellml.org/cellml/1.0#"
  xmlns:cellml="http://www.cellml.org/cellml/1.0#"
  xmlns: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>
  <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>
```



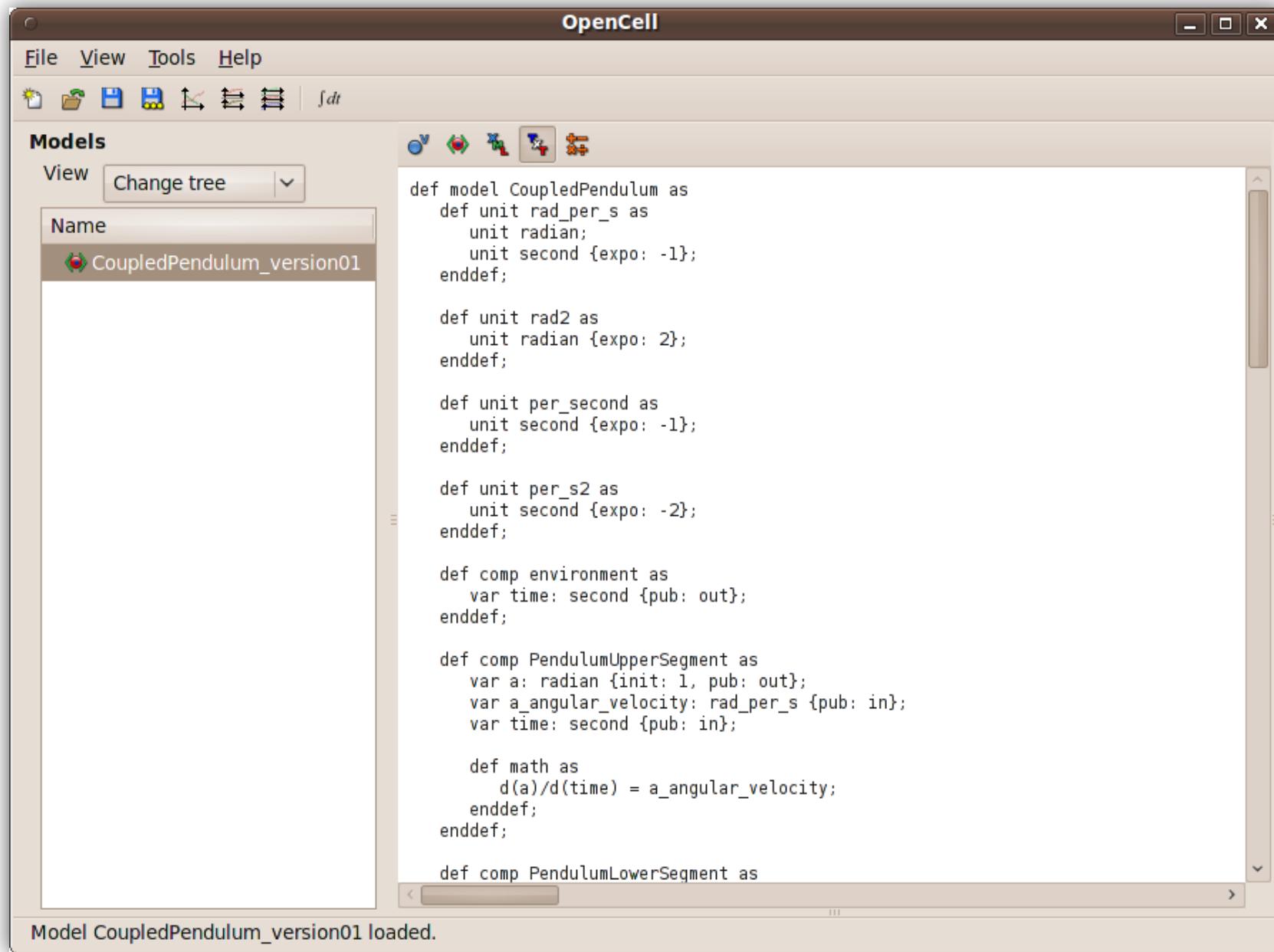
```
def model hodgkin_huxley_squid_axon_1952 as
  def unit millisecond from
    unit second {pref: "milli"};
  enddef;
  ...
  def comp sodium_channel as
    var i_Na: microA_per_cm2 {pub: out};
    ...
    E_Na = E_R+115.0{unit: "millivolt"};
    i_Na = g_Na*pow(m, 3.0{unit: "dimensionless"})*h*(V-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

# COR-LIKE VIEW IN OPENCCELL



# KNOWN DIFFERENCES/LIMITATIONS

- 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:

```
i_Stim = piecewise(
    case time >= 10{def group as
        type containment;
    then
        -20{unit: microA_per_cm2};
    else
        0{unit: microA_per_cm2}; enddef;
);
i_Stim = sel
    case time >= 10{def group as containment for
        type containment;
    then
        ...;
    else
        -20{microA_per_cm2}; enddef;
}
otherwise:
    0{microA_per_cm2};
endsel;
```

COR

COR

COR

- 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](http://www.opencellproject.org)



Randal Britten  
[r.britten@auckland.ac.nz](mailto:r.britten@auckland.ac.nz)



Alan Garry  
[alan.garry@dpag.ox.ac.uk](mailto:alan.garry@dpag.ox.ac.uk)



Justin Marsh  
[j.marsh@auckland.ac.nz](mailto:j.marsh@auckland.ac.nz)



Andrew Miller  
[a.miller@auckland.ac.nz](mailto:a.miller@auckland.ac.nz)