

OpenCell

Status and plans

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



VARIOUS VERSIONS OF OPENCELL

- Several versions of OpenCell (formerly known as PCEnv) over the past few years. Some key features include:
 - **PCEnv 0.4:** graphical rendering of mathematics, units, etc., and C code export;

VARIOUS VERSIONS OF OPENCELL

- See PC inc
-

Physiome CellML Environment

File Tools View Help

Models

View Change tree

Name	Type	Value	Units
cheng_model_2000	Mathematics	12	
	Equation		
	Mathematics	13	
	Equation		
	Mathematics	14	
	Equation		
	Mathematics	15	
	Equation		
V	V	0.09314	first_order_rate_constant
mass	mass	0.005	kilogram
F_total	F_total		newton
time	time		millisecond

Equation $T_f = \text{piecewise}(\text{case } df_eff_dt \geq 0.0 \{ \text{units}="dimensionless" \} \text{ then } T_f$

$$T_f = \begin{cases} T_{f1} \times L^{2.0} + T_{f2} \times f_{env} & \text{if } df_eff_dt \geq 0.0 \\ \frac{T_{f3} + T_{f4} \times Af}{L} & \text{otherwise} \end{cases}$$

Model cheng_model_2000 loaded.

in as
tures
units,

VARIOUS VERSIONS OF OPENCELL

- Several versions of OpenCell (formerly known as PCEnv) over the past few years. Some key features include:
 - **PCEnv 0.4:** graphical rendering of mathematics, units, etc., and C code export;
 - **PCEnv 0.5:** visual validator, experimental DAE solver;

VARIOUS VERSIONS OF OPENCELL

- See
- PC
- inc
-
-

Physiome CellML Environment

File Tools View Help

Type Value Units

Type	Value	Units
L		dimensionless
V		first_order_rate_constant
Mathematics	5	
Equation		
Af		
F0		
F_CE		
F_total		
L_eff		
S		
Y		

Equation

$$FV = \text{piecewise}(\text{case } V \leq 0.0 \{ \text{units} = \text{"dimensionless"} \} \text{ then } (V_{\text{ma}} \dots$$

Warning: Expected all arguments to MathML apply to have the same units

Warning: Expected arguments to MathML apply to have dimensionless units

Warning: Expected all arguments to MathML apply to have the same units

Warning: MathML equals element has inconsistent units between the sides

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Error: Connection of two variables which have dimensionally inconsistent units

Error: Mapping variable_1 has public interface of in but variable_2 also has public interface of in

Dismiss

Model cheng_model_2000 loaded.

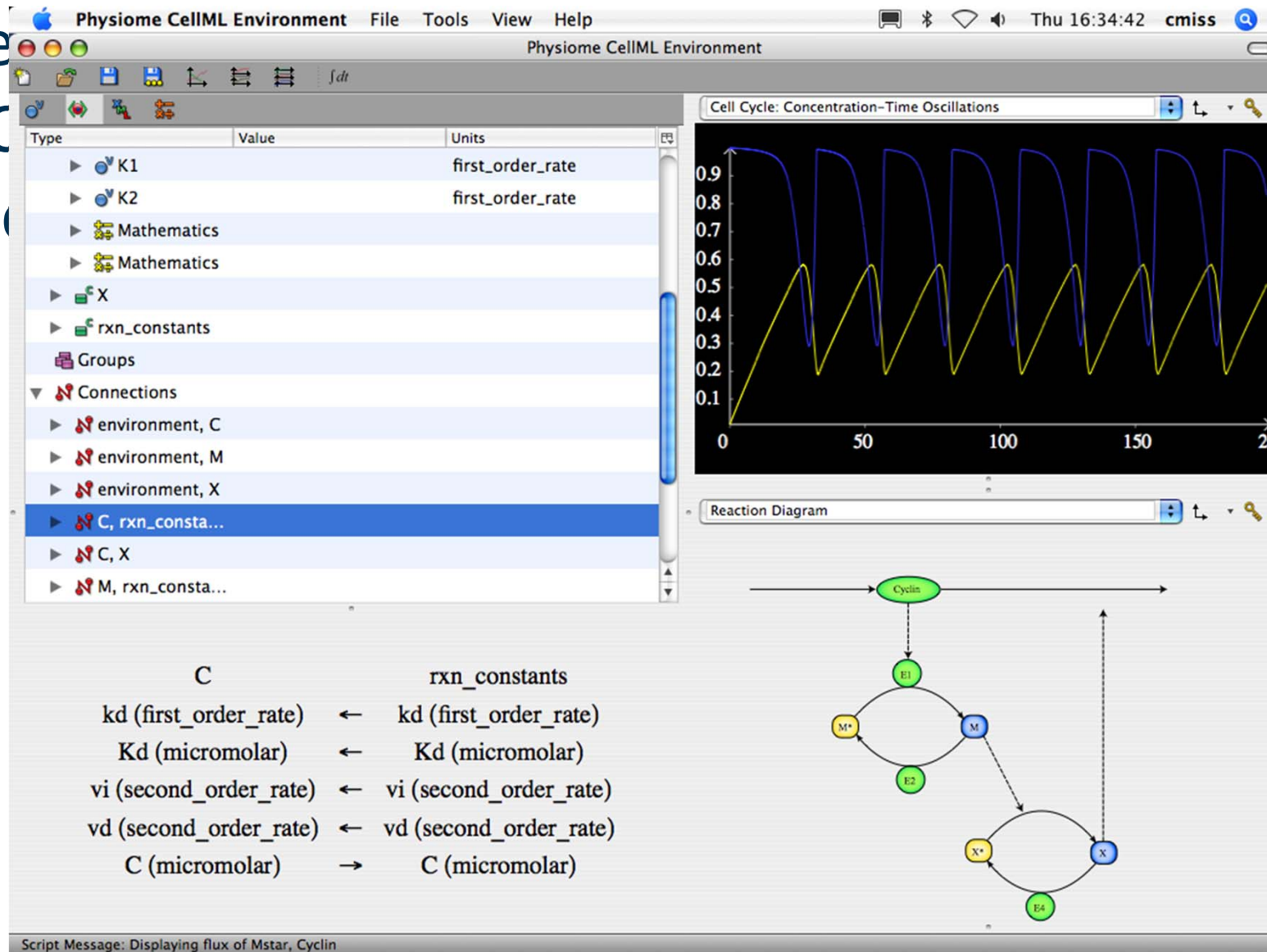
own as
tures
units,
er;

VARIOUS VERSIONS OF OPENCELL

- Several versions of OpenCell (formerly known as PCEnv) over the past few years. Some key features include:
 - **PCEnv 0.4:** graphical rendering of mathematics, units, etc., and C code export;
 - **PCEnv 0.5:** visual validator, experimental DAE solver;
 - **PCEnv 0.6:** MATLAB and Python code export in addition to C (incl. DAE models), as well as Mac OS X support;

VARIOUS VERSIONS OF OPENCELL

- See PC in
-
-
-



on as
tures
units,
r;
dition
t;

VARIOUS VERSIONS OF OPENCELL

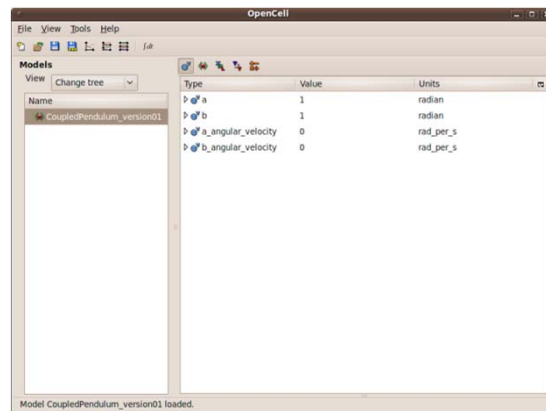
- Several versions of OpenCell (formerly known as PCEnv) over the past few years. Some key features include:
 - **PCEnv 0.4:** graphical rendering of mathematics, units, etc., and C code export;
 - **PCEnv 0.5:** visual validator, experimental DAE solver;
 - **PCEnv 0.6:** MATLAB and Python code export in addition to C (incl. DAE models), as well as Mac OS X support;
 - **OpenCell 0.7:** support for drag and drop editing of connections and graph traces, multiline math input, experimental Fortran 77 code export; and

VARIOUS VERSIONS OF OPENCELL

- Several versions of OpenCell (formerly known as PCEnv) over the past few years. Some key features include:
 - **PCEnv 0.4:** graphical rendering of mathematics, units, etc., and C code export;
 - **PCEnv 0.5:** visual validator, experimental DAE solver;
 - **PCEnv 0.6:** MATLAB and Python code export in addition to C (incl. DAE models), as well as Mac OS X support;
 - **OpenCell 0.7:** support for drag and drop editing of connections and graph traces, multiline math input, experimental Fortran 77 code export; and
 - **OpenCell 0.8RC1:** support for IDA as an integrator, copy and paste of parts of models from the tree views.

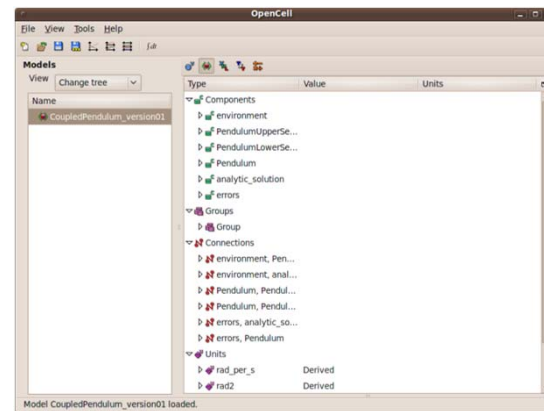
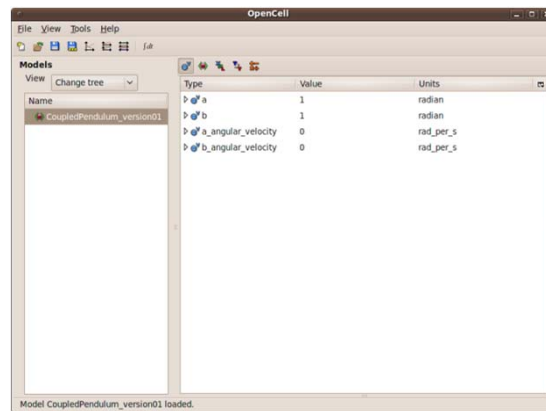
EDITING IN OPENCCELL

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



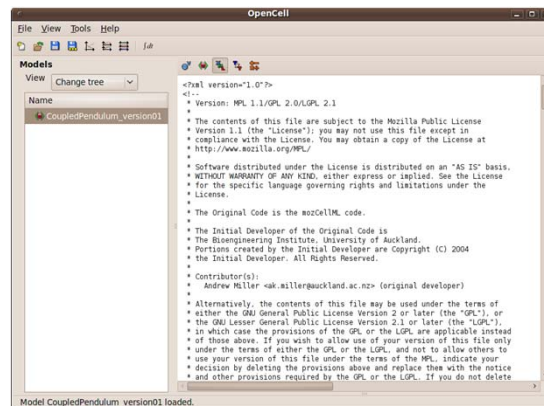
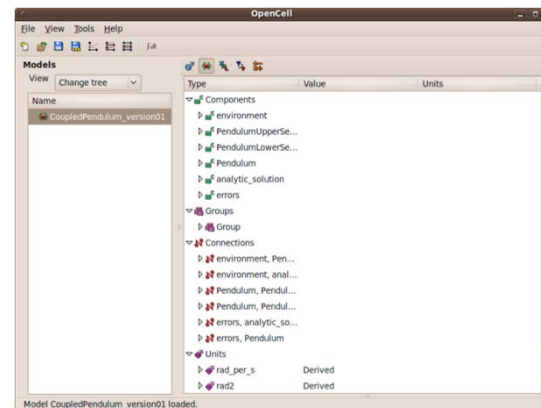
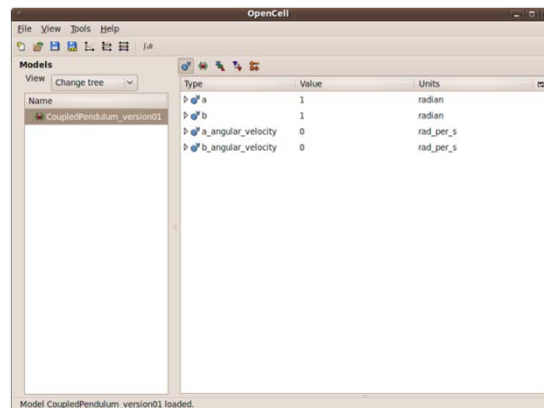
EDITING IN OPENCELL

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



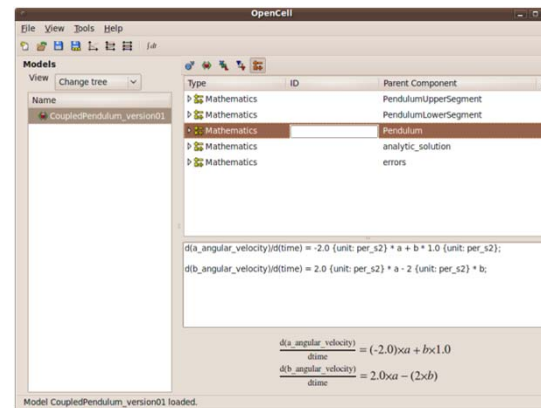
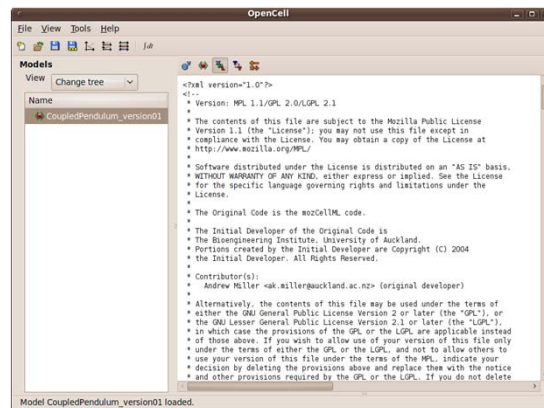
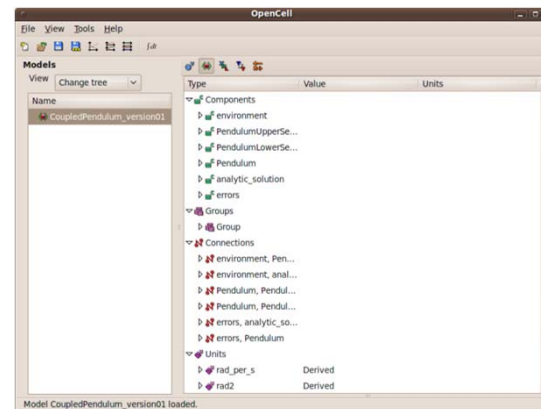
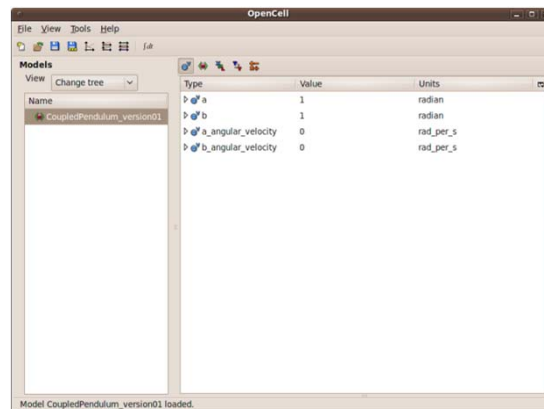
EDITING IN OPENCELL

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



EDITING IN OPENCELL

- 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 IN OPENCELL

- 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 most obvious way to edit a CellML file.
- Another possible approach is that of COR which relies on a proprietary language.

```
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_millisecond} * exp((-V - 48 {millivolt}) / 15 {millivolt});
  beta_m = 120 {per_millisecond} * exp((V + 8 {millivolt}) / 5 {millivolt});
  ode(m, time) = alpha_m * (1 {dimensionless} - m) - beta_m * m;
enddef;

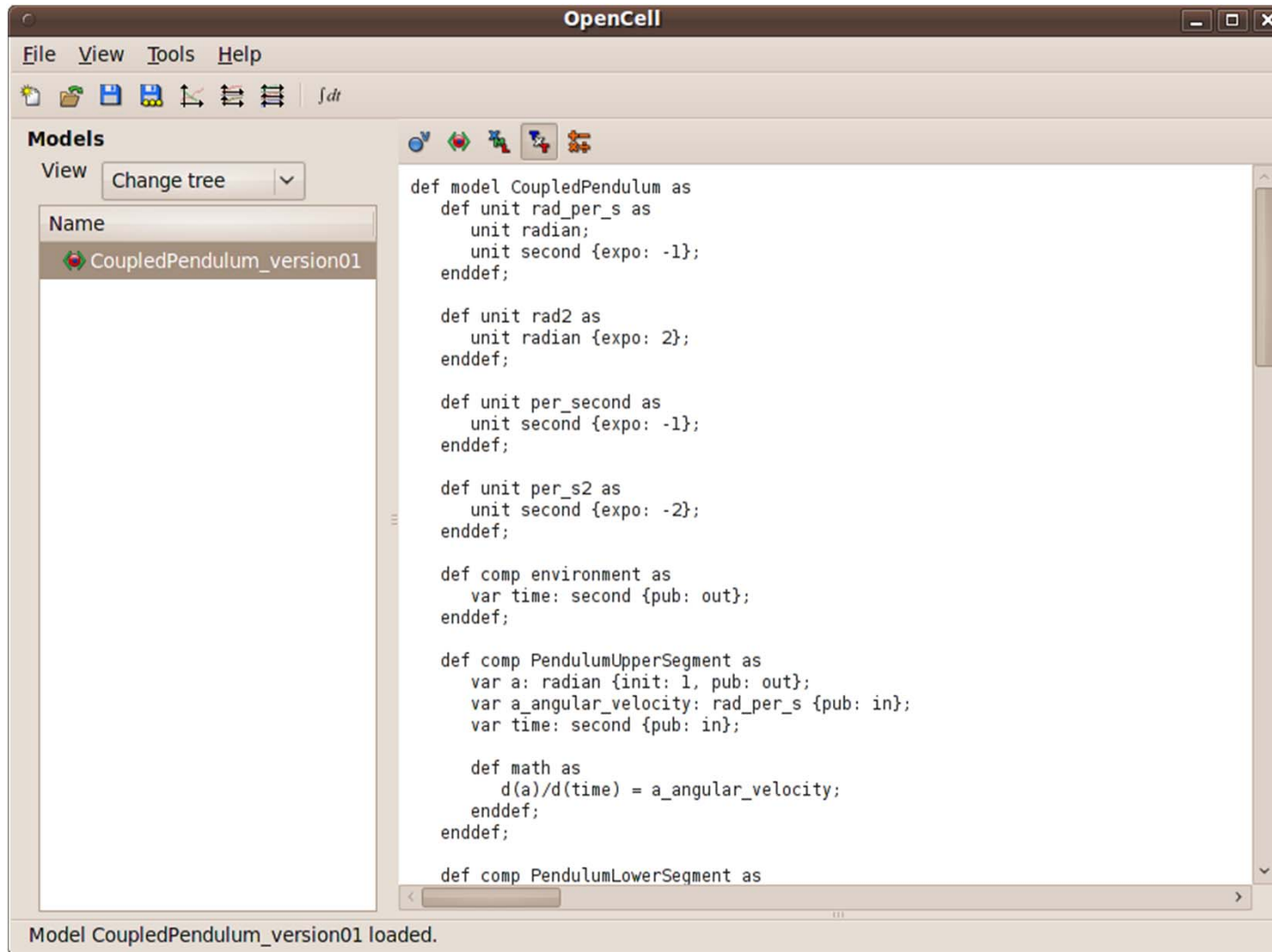
def comp sodium_channel_h_gate as
  var h: dimensionless {init: 0.8, pub: out};
  var alpha_h: per_second;
  var beta_h: per_second;
  var V: millivolt {pub: in};
  var time: second {pub: in};

  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};
```

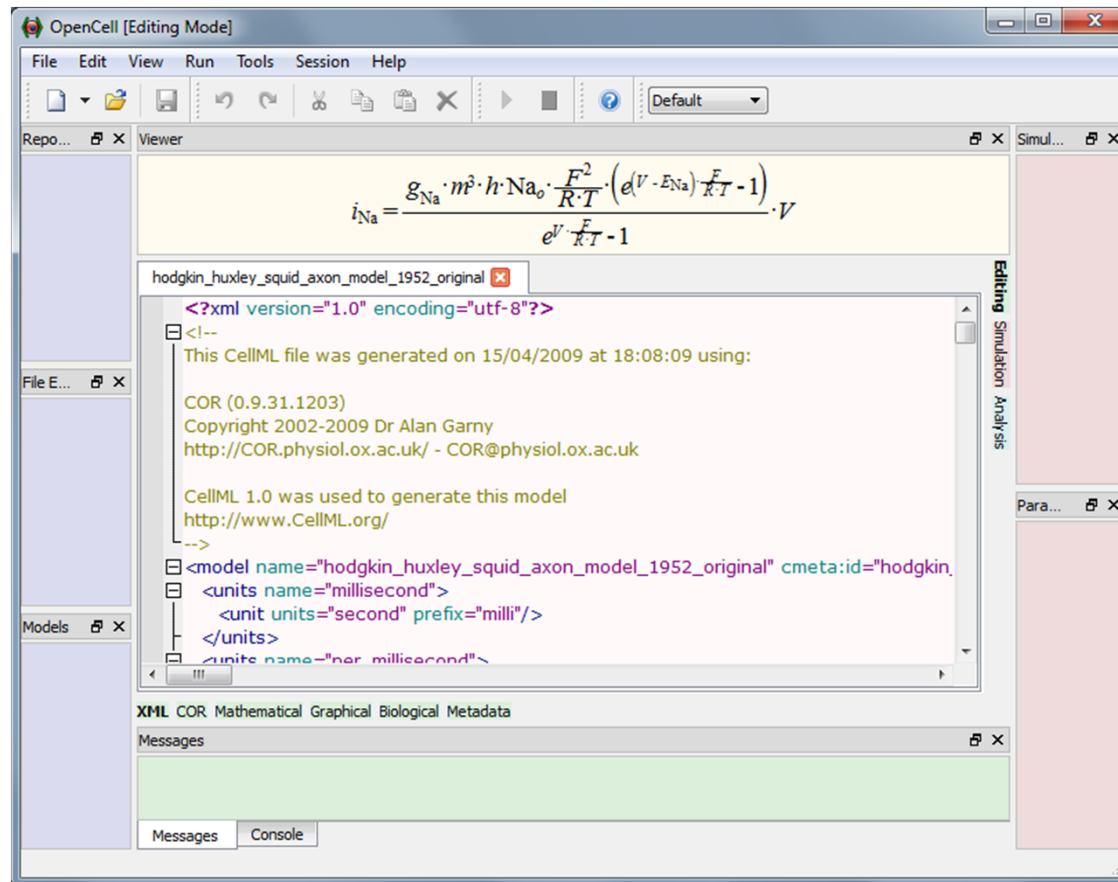
COR-LIKE VIEW IN OPENCELL

- Not 100% compatible with COR, but CellML 1.1 capable.



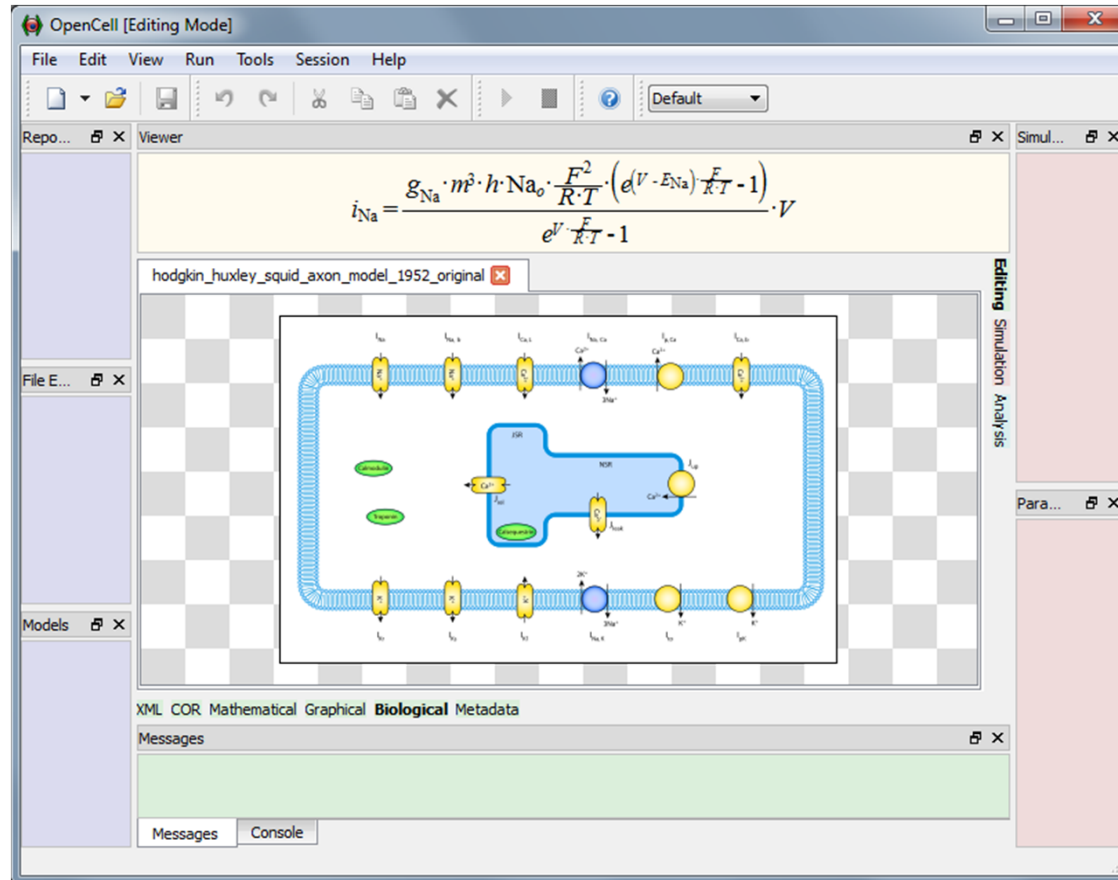
PLAN FOR A NEW OPENCELL

- OpenCell currently relies on the Mozilla XULRunner Framework, making it difficult to develop OpenCell further.
- New OpenCell to be developed using Qt/C++.



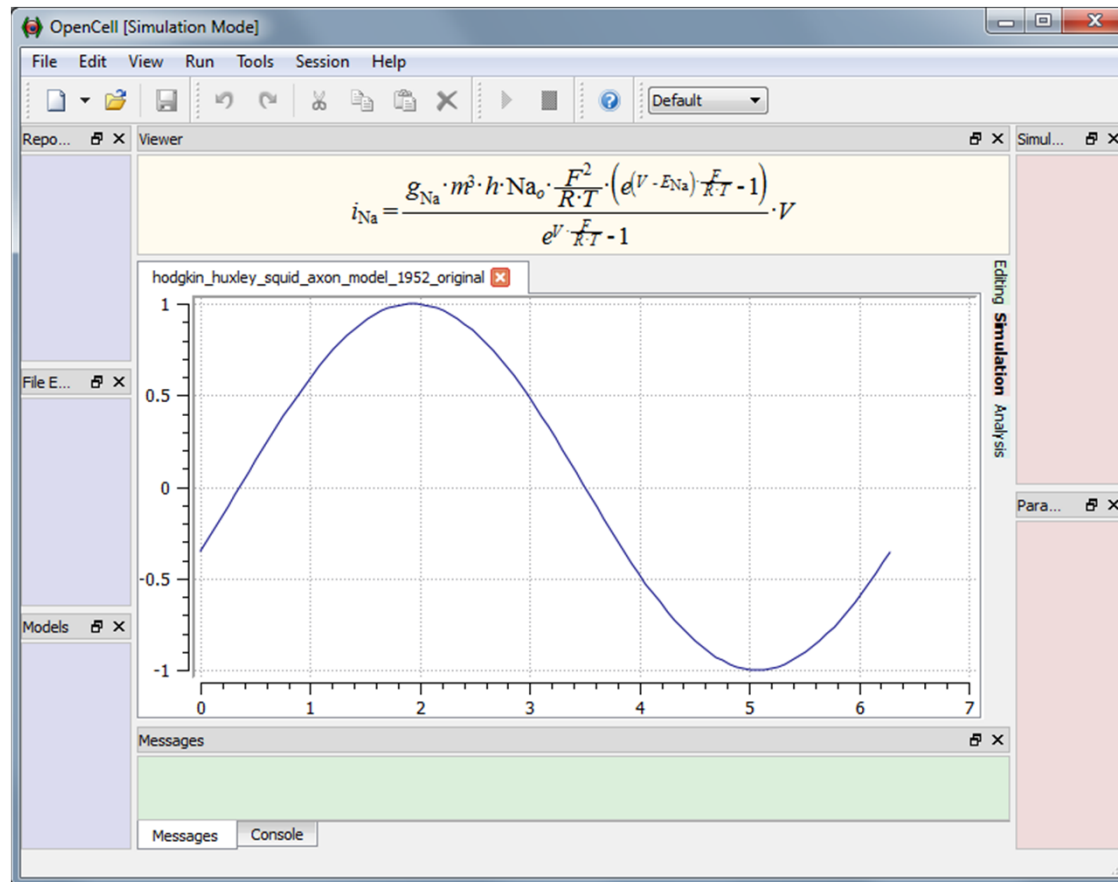
PLAN FOR A NEW OPENCELL

- OpenCell currently relies on the Mozilla XULRunner Framework, making it difficult to develop OpenCell further.
- New OpenCell to be developed using Qt/C++.



PLAN FOR A NEW OPENCELL

- OpenCell currently relies on the Mozilla XULRunner Framework, making it difficult to develop OpenCell further.
- New OpenCell to be developed using Qt/C++.



PLAN FOR A NEW OPENCELL

- OpenCell currently relies on the Mozilla XULRunner Framework, making it difficult to develop OpenCell further.
- New OpenCell to be developed using Qt/C++.
- Aiming at a working version by the end of next year.
- Initial focus will be on combining the current versions of OpenCell and COR.
- Next, the focus will be on metadata, ontologies, etc.

CONCLUSION

- OpenCell 0.8 is soon to be released (OpenCell 0.8RC1 is currently available for download).
- There might be an OpenCell 0.9 (with the COR-like view).
- OpenCell, as we know it, is soon to enter maintenance mode.
- A new OpenCell is to be developed (led by Oxford), using the existing CellML 1.0/1.1 API (led by Auckland).
- A first public release is expected by the end of next year.

www.opencell.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

ACKNOWLEDGEMENTS

