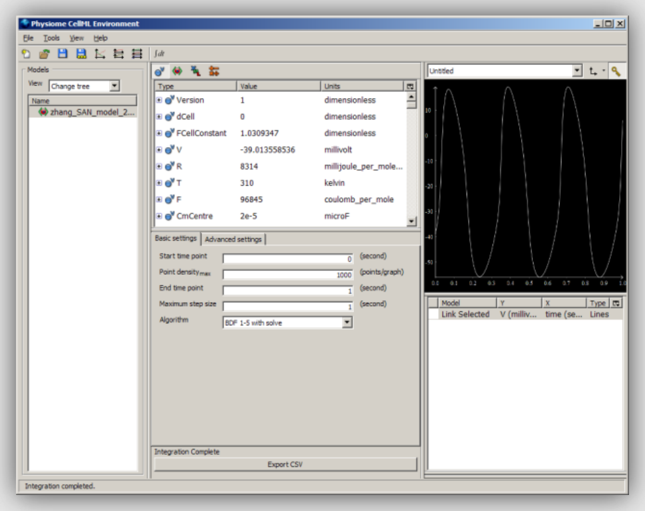
OpenCOR

Alan Garny

EMBL-EBI

Background

- The two main environments for editing and simulating CellML files are COR (http://cor.physiol.ox.ac.uk/) and OpenCell (http://www.opencell.org/).



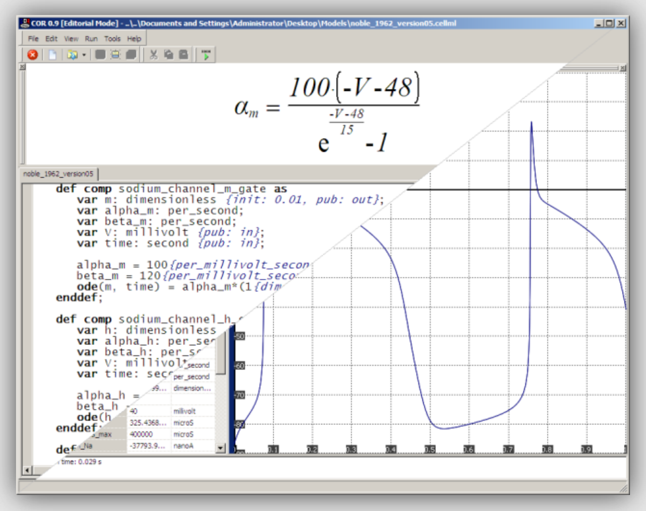
Type	Value	Units
Version	1	dimensionless
dCell	0	dimensionless
FCellConstant	1.0309347	dimensionless
V	-39.013558536	millivolt
R	8314	millijoule_per_mole...
T	310	kelvin
F	96845	coulomb_per_mole
CmCentre	2e-5	microF

Start time point	0	(second)
Point density$_{max}$	1000	(points/graph)
End time point	1	(second)
Maximum step size	1	(second)
Algorithm	BDF 1-5 with solve	

Model	Y	X	Type
Link Selected	V (milliv...	time (se...	Lines

Integration Complete

Integration completed.


COR 0.9 [Editorial Mode] - ..\..\Documents and Settings\Administrator\Desktop\Models\noble_1962_version05.cellml
File Edit View Run Tools Help
noble_1962_version05

$$\alpha_m = \frac{100\cdot(-V-48)}{e^{\frac{-V-48}{15}}-1}$$

```
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_secon
   beta_m = 120{per_millivolt_seco
   ode(m, time) = alpha_m*(1{dim
enddef;

def comp sodium_channel_h_
   var h: dimensionless
   var alpha_h: per_se
   var beta_h: per_s
   var V: millivol
   var time: sec

   alpha_h =
   beta_h
   ode(h
enddef;

def
```

		second
		per_second
	...9...	dimension...
	40	millivolt
	325.4368...	microS
..._max	400000	microS
..._Na	-37793.9...	nanoA

time: 0.029 s

- Having very similar goals, their authors decided to work on a combined and improved environment: OpenCOR.

General Information

- OpenCOR is an open source project.
- No license yet, but it might have to be GPL (i.e. not business friendly).
- A (very) simple website has been setup.

  ⇨ http://www.opencor.ws/ ⇦
- The project is hosted on GitHub.

  ⇨ https://github.com/opencor/opencor/ ⇦
- Currently being developed (using Qt), built, run and tested on:
  - Windows 7;
  - Ubuntu 11.04 (Natty Narwhal; both 32-bit and 64-bit); and
  - Mac OS X 10.7 (Lion).

General Philosophy

- OpenCOR can be used either from the command line or through a graphical user interface (GUI).
- Everything is available as a plugin. This means that:
  - Features can be enabled/disabled as the user sees fit; and that
  - New features can be easily added (e.g. support for SBML).
- A four-step approach:
  - Organise CellML files (CellML Model Repository, File Browser and File Organiser plugins);
  - Edit CellML files using different (plugin) views (e.g. Raw, Raw CellML and CellML Annotation);
  - Simulate CellML files using the Single Cell (plugin) view; and
  - Analyse simulation data (pending).

Plugin Approach

- OpenCOR currently offers the following interfaces:
  - Core: to initialise/finalise a plugin, as well as to load/save its settings;
  - File: to specify the file types supported by a plugin;
  - GUI: to customise OpenCOR's GUI (i.e. menus, docking windows and views);
  - Internationalisation: to support the translation of a plugin; and
  - Solver: to specify the interface to a numerical solver and make it available to other plugins.
- A plugin can have dependencies on other plugins.
- A user can decide whether a plugin is to be loaded (e.g. the Raw CellML plugin) while other plugins will only be loaded if needed (e.g. the CellML plugin).

Organise CellML Files

- CellML Model Repository plugin: an interface to PMR2 through Web services (REST/JSON).
- This work is still in progress, so:
  - Currently: implemented a proof of technical feasibility by retrieving a list of CellML files (and allowing it to be searched and workspaces to be looked up); but
  - In the future: clone a workspace, create an exposure, etc.
- File Browser plugin: to get access to physical files, be they CellML files or not.
- File Organiser plugin: to virtually organise physical files.

Edit CellML Files

- People work in different ways and there exist different approaches to modelling.
- It should therefore be possible to edit a CellML file in more than just one way.
- We therefore have different CellML editing views (e.g. a Raw CellML view, a COR-like view or a tree-like view).
- Those CellML views are connected to a CellML File Manager to keep track of changes to a CellML file.
- Note: a similar approach could be used to edit SBML files, SED-ML files, etc.

Simulate CellML Files (I)

- When it comes to simulation, speed is paramount.
- A CellML file should therefore be compiled. For example:
  - OpenCell generates some C code (using the CCGS service) which is then compiled (using gcc) into a shared library.
  - COR uses its own internal compiler to generate binary code.
- However, to rely on gcc is not convenient and internal generation of binary code is tedious. So, OpenCOR:
  - Generates some C code (still using the CCGS service);
  - Parses that code to get an AST representation of the model;
  - Uses that AST rep. to generate some LLVM assembly code;
  - Asks LLVM to compile the assembly code into binary code; and
  - Executes the binary code using LLVM's JIT execution engine.

Simulate CellML Files (II)

- Two types of solvers are to be supported initially:
  - ODE solvers (done); and
  - DAE solvers (pending).
- Plugins only know the following about a solver:
  - Its name;
  - Its type (i.e. ODE or DAE solver); and
  - Its parameter names and types (so it can be customised).
- An instance of the solver can also be retrieved.
- Forward Euler and CVODE are two ODE solvers which can currently be used.

Demonstration time!

Useful Links

- OpenCOR:
  - Web site: http://www.opencor.ws/; and
  - GitHub repository: https://github.com/opencor/opencor/.
- Other CellML environments:
  - OpenCell: http://www.opencell.org/; and
  - COR: http://cor.physiol.ox.ac.uk/.
- Miscellaneous:
  - Qt: http://qt.nokia.com/;
  - CellML API: http://www.cellml.org/tools/api/;
  - LLVM: http://www.llvm.org/; and
  - SUNDIALS (e.g. CVODE and IDA): https://computation.llnl.gov/casc/sundials/main.html.

Virtual Physiological Human
network of excellence

SEVENTH FRAMEWORK PROGRAMME

preDiCT

British Heart Foundation

RICORDO
INTEROPERABLE ANATOMY AND PHYSIOLOGY