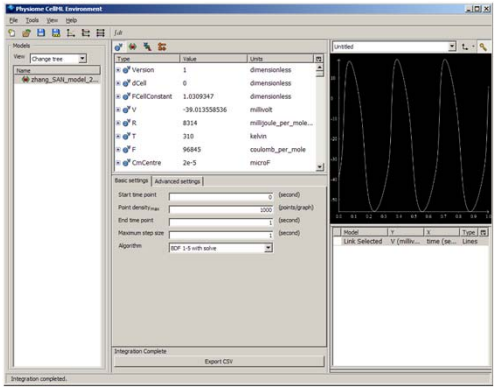
OpenCOR

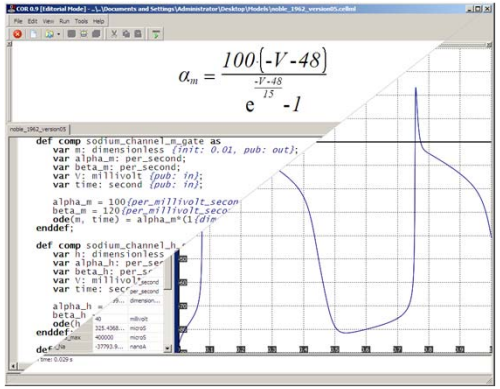
Alan Garny

alan.garny@dpag.ox.ac.uk

OpenCell and COR

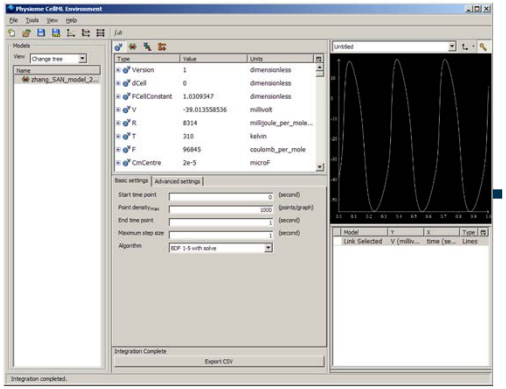
- The two main CellML environments are OpenCell and COR.
- OpenCell:
  - Version 0.8 was released in October 2010.
  - Support for both CellML 1.0 and 1.1.
  - Works on Windows, Linux and Mac OS X.
  - Now officially discontinued.
- COR:
  - Version 0.9.31.1409 was released in November 2010.
  - Support for CellML 1.0.
  - Works on Windows.
  - No new major feature in years, but still maintained.





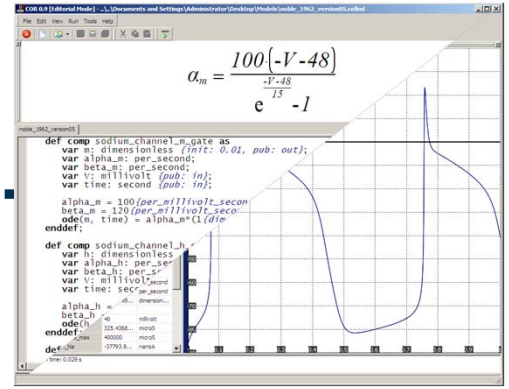
OpenCell and COR

- The two main CellML environments are OpenCell and COR.
- OpenCell:
  - Version 0.8 was released in October 2010.
  - Support for both CellML 1.0 and 1.1.
  - Works on Windows, Linux and Mac OS X.
  - Now officially discontinued.
- COR:
  - Version 0.9.31.1409 was released in November 2010.
  - Support for CellML 1.0.
  - Works on Windows.
  - No new major feature in years, but still maintained.



OpenCOR

*(**Open COmbined Resource**)*



General Information

- OpenCOR is an open source project.
- Agreement on a (business friendly) license is yet to be reached (3-clause BSD license?).
- The project is hosted on SourceForge using Git.

⇨ http://sourceforge.net/projects/opencor/ ⇦

- A (very) simple website has been set up.

⇨ http://www.opencor.ws/ ⇦

- Currently being developed, built, tested and packaged on:
  - Windows 7;
  - Ubuntu 10.10 (Maverick Meerkat; both the 32-bit and 64-bit versions); and
  - Mac OS X 10.6 (Snow Leopard).

Develop, Build, Test and Package

- Some pre-requirements:
  - Git (the `git` package on Ubuntu):

```
$ git clone git://opencor.git.sourceforge.net/gitroot/opencor/opencor        (Read Only)
$ git clone ssh://login@opencor.git.sourceforge.net/gitroot/opencor/opencor  (Read/write)
```

Develop, Build, Test and Package

- Some pre-requirements:
  - Git (
    - $ g
    - $ g

```
3rdparty                    // Various third-party libraries used by OpenCOR
 ├─ LibQxt                  // The LibQxt library
 ├─ QScintilla              // The QScintilla library
 ├─ QtMmlWidget             // The QtMmlWidget library
 └─ QtSingleApplication     // The QtSingleApplication library
build                       // Where OpenCOR-related build files are generated
distrib                     // Various files used for the packaging of OpenCOR
doc                         // User and developer documentation
 ├─ 3rdparty                // Third-party library used for the documentation
 │   └─ googleCodePrettify  // The Google Code Prettify library
 ├─ developer               // Developer documentation
 ├─ res                     // Various resource files used by the documentation
 └─ user                    // User documentation
i18n                        // Internationalisation files
res                         // Various resource files used by OpenCOR
 ├─ boomy                   // The Boomy multiplatform icon library
 ├─ flags                   // The Printable world flags icon library
 └─ oxygen                  // The Oxygen icon library
src                         // Source code files for OpenCOR
 ├─ misc                    // Files that do not fit anywhere else
 ├─ ui                      // User interfaces
 │   ├─ edit                // User interfaces for the editing side of things
 │   └─ organise            // User interfaces for the organisation side of things
 └─ widget                  // Widgets
     ├─ edit                // Widgets for the editing side of things
     ├─ misc                // Widget files that do not fit anywhere else
     └─ organise            // Widgets for the organisation side of things
winConsole                  // Console version of OpenCOR (Windows only)
 ├─ build                   // Where the build files are generated
 ├─ res                     // Various resource files
 └─ src                     // Source code files
```

(Read Only)

or *(Read/write)*

Develop, Build, Test and Package

- Some pre-requirements:
  - Git (the `git` package on Ubuntu):

    ```
    $ git clone git://opencor.git.sourceforge.net/gitroot/opencor/opencor        (Read Only)
    $ git clone ssh://login@opencor.git.sourceforge.net/gitroot/opencor/opencor  (Read/write)
    ```

  - CMake 2.8+ (the `cmake` package on Ubuntu);
  - C++ toolchain:
    - Windows: `MinGW 4.4.0`;
    - Ubuntu: the `g++` package; and
    - Mac OS X: `Xcode`.
  - Qt 4.7.2:
    - Windows and Mac OS X: binaries available; and
    - Ubuntu: it has to be built from its source code which requires some additional packages.
  - Qt Creator 2.1.0; and
  - NSIS (Windows only).

Note: the above pre-requirements are meant for someone who wants to develop, build, test and package OpenCOR in the exact same way as it is done by the OpenCOR team. However, when it comes to Qt and Qt Creator, older versions may also work (e.g. on Ubuntu, you might just want to install the qtcreator package).

⇨ http://www.opencor.ws/developer/developBuildTestAndPackage.html ⇦

Develop, Build, Test and Package

- Development is currently single-handed.
- This will, however, change once the project has reached a certain level of maturity.
- Best practices:
  - File structure; and

Develop, Build, Test and Package

- Development is currently single-handed
- This w eached a certain
- Best p
  - File

```
3rdparty                    // Various third-party libraries used by OpenCOR
  ├─ LibQxt                 // The LibQxt library
  ├─ QScintilla             // The QScintilla library
  ├─ QtMmlWidget            // The QtMmlWidget library
  └─ QtSingleApplication    // The QtSingleApplication library
build                       // Where OpenCOR-related build files are generated
distrib                     // Various files used for the packaging of OpenCOR
doc                         // User and developer documentation
  ├─ 3rdparty               // Third-party library used for the documentation
  │   └─ googleCodePrettify // The Google Code Prettify library
  ├─ developer              // Developer documentation
  ├─ res                    // Various resource files used by the documentation
  └─ user                   // User documentation
i18n                        // Internationalisation files
res                         // Various resource files used by OpenCOR
  ├─ boomy                  // The Boomy multiplatform icon library
  ├─ flags                  // The Printable world flags icon library
  └─ oxygen                 // The Oxygen icon library
src                         // Source code files for OpenCOR
  ├─ misc                   // Files that do not fit anywhere else
  ├─ ui                     // User interfaces
  │   ├─ edit               // User interfaces for the editing side of things
  │   └─ organise           // User interfaces for the organisation side of things
  └─ widget                 // Widgets
      ├─ edit               // Widgets for the editing side of things
      ├─ misc               // Widget files that do not fit anywhere else
      └─ organise           // Widgets for the organisation side of things
winConsole                  // Console version of OpenCOR (Windows only)
  ├─ build                  // Where the build files are generated
  ├─ res                    // Various resource files
  └─ src                    // Source code files
```

Develop, Build, Test and Package

- Development is currently single-handed.
- This will, however, change once the project has reached a certain level of maturity.
- Best practices:
  - File structure; and
  - Coding style: the rules were unashamedly taken (paraphrased, if not simply copied/pasted) from the coding style document written by the developers of Qt Creator.

⇨ http://www.opencor.ws/developer/bestPractices.html ⇦

Develop, Build, Test and Package

- OpenCOR uses various third-party libraries.
- However, all the required files are in our Git repository.
- For the relevant software libraries, a/the `CMakeLists.txt` file was created/modified so that the library can be built as part of our build process.
- Third-party libraries currently being used:
  - Software:
    - Google Code Prettify (Apache v2.0);
    - LibQxt 0.6.1 (CPL v1.0 and LGPL v2.1);
    - QScintilla 2.4.6 (GPL v2.0, GPL v3.0 and commercial);
    - QtMmlWidget 2.4 (LGPL v2.1); and
    - QtSingleApplication (BSD).
  - Image:
    - Printable world flags (no specific license);
    - Boomy multiplatform (no specific license); and
    - Oxygen 4.6.1 (LGPL v3.0).

⇨ http://www.opencor.ws/developer/thirdPartyLibraries.html ⇦

Develop, Build, Test and Package

- OpenCOR can be built either from the command line or using Qt Creator.
- Command line:
  - `cleanproj[.bat]`: cleans the OpenCOR environment;
  - `cleanprojall[.bat]`: cleans all of the OpenCOR environment (i.e. including third-party libraries);
  - `makeproj[.bat]`: compiles and links everything that is required to get a release version of OpenCOR; and
  - `runproj[.bat|.vbs]`: runs OpenCOR.
- Qt Creator:
  - It's pretty much a matter of just opening the main `CMakeLists.txt` file and of clicking the Run button.
  - Support for parallel compilation and debugging (using `gdb` under the hood).

⇨ http://www.opencor.ws/developer/developBuildTestAndPackage.html ⇦

Develop, Build, Test and Package

- Packaging is done from the command line, using `packageproj[.bat]`:
  - Windows: both a NSIS setup program and a ZIP file are generated;
  - Linux: a `.tar.gz` file is generated; and
  - Mac OS X: both a PackageManager solution and a ZIP file are generated.

  Note: though built on Ubuntu, OpenCOR should also work with other versions of Ubuntu, as well as other Linux distributions though additional libraries may be required in this particular case.

  ⇨ http://www.opencor.ws/developer/developBuildTestAndPackage.html ⇦

General Philosophy

- OpenCOR can be used both as a command line tool and through a graphical user interface.

```
agarny@pc-hblin:~/OpenCOR$ ./OpenCOR -h
Usage: OpenCOR [OPTION]... [FILE]...
Start OpenCOR and open the FILE(s) passed as argument(s).

 -a, --about     Display OpenCOR about information
 -h, --help      Display this help information
 -v, --version   Display OpenCOR version information

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.
agarny@pc-hblin:~/OpenCOR$ ./OpenCOR –a
OpenCOR 0.1
GNU/Linux 2.6.35-28-generic

OpenCOR is a cross-platform CellML-based modelling environment which can be
used to organise, edit, simulate and analyse CellML files.
agarny@pc-hblin:~/OpenCOR$ ./OpenCOR –v
OpenCOR 0.1
agarny@pc-hblin:~/OpenCOR$
```

General Philosophy

- OpenCOR can be used both as a command line tool and through a graphical user interface. Some general features:
  - Fully customisable (an on-going process);
  - Multi-lingual (English and French at the moment);
  - Auto updates (both for development versions and release versions);
  - Crash report (still need to think a bit more about this one);
  - Plugin approach (to allow for community involvement); and
  - User friendly (as much as possible!).
- OpenCOR will be used to organise, edit, simulate and analyse CellML and, in the long-term future, SBML files:
  - CellML and SBML support will be through the Auckland CellML API and the SBML API, respectively;
  - Access to the CellML Model Repository and BioModels Database (through Web services);
  - Support for ontologies (through Web services to RICORDO);
  - Support for SED-ML (through the CellML and SBML APIs?); and
  - 2D and 3D representation of a model (using SVG and cmgui, respectively).

ORGANISE

- Access to:
  - The CellML Model Repository and BioModels Database to search, download, upload, etc. a model;
  - A file browser and therefore local files (be they CellML/SBML documents or not); and
  - A file organiser to virtually arrange files (e.g. a virtual folder that contains a symbolic link to all the CellML files used in a modelling study).

Editing (CellML Files Initially)

- Editing of a CellML file using one of several views:
  - Raw XML view;
  - COR-like view (i.e. a compact format);
  - Improved COR-like view (i.e. COR-like view without CellML-specific concepts such as public/private interfaces, encapsulation hierarchy and connections);
  - Tree-like view (with the ability to show/hide various CellML features); and
  - Graphical view (using SVG or cmgui, and based on the metadata/ontology information present in a CellML file).
- Some other editing features:
  - Graphical rendering of unit/variable definitions, mathematical equations, etc.;
  - CellML validation (list of warnings/errors highlighting a particular problem);
  - Basic metadata support (i.e. model authors/description/curators, etc.);
  - Advanced metadata support so that, using domain-specific ontologies (RICORDO), models can be comprehensively annotated;
  - Diff of two CellML files (only for text-based views); and
  - Export of CellML files to various programming (e.g. C/C++, F77, Java MATLAB, Python) and word processing languages (MS Word 2007/2010 and TeX); and
  - Import/export from/to SBML.

SIMULATE

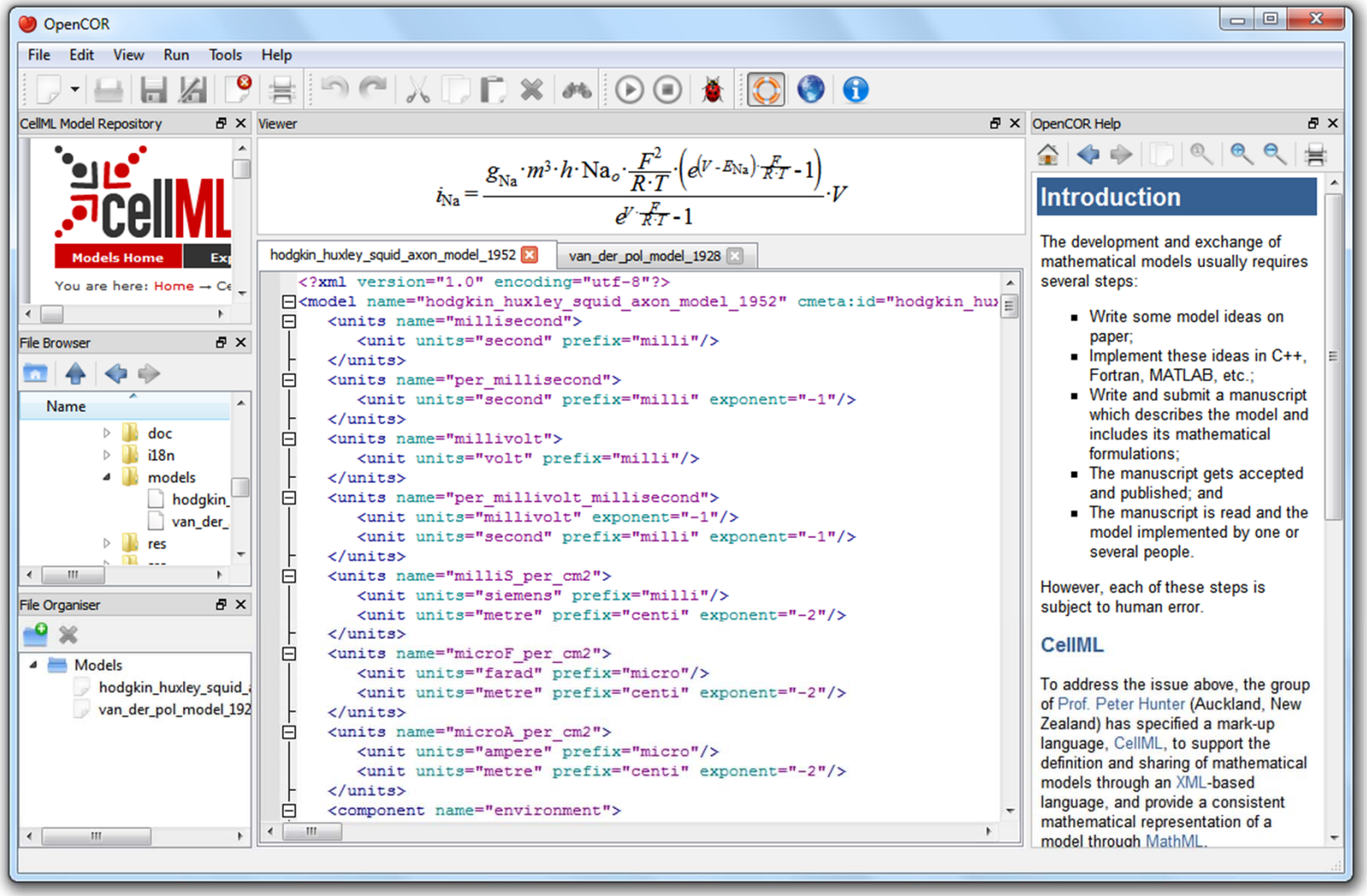
- Some general features include:
  - Editing of simulation (e.g. start/end points) and numerical solver parameters (dependent on the solver to be used);
  - Editing of model parameters (using a tree-like view);
  - Support for models consisting of differential algebraic equations (CVODE and IDA as default solvers);
  - Run/pause/stop a simulation;
  - Plotting of simulation results (against any model param.);
  - Export of simulation results to the CSV format; and
  - Create a new or update an existing CellML file based on the results of a simulation.

ANALYSE

- Analysis features will mainly be provided by the community through the use of plugins.
- For example, a plugin to analyse cardiac action potentials and extract some key parameters from them (e.g. upstroke velocity, action potential amplitude, action potential duration at 90% repolarisation).

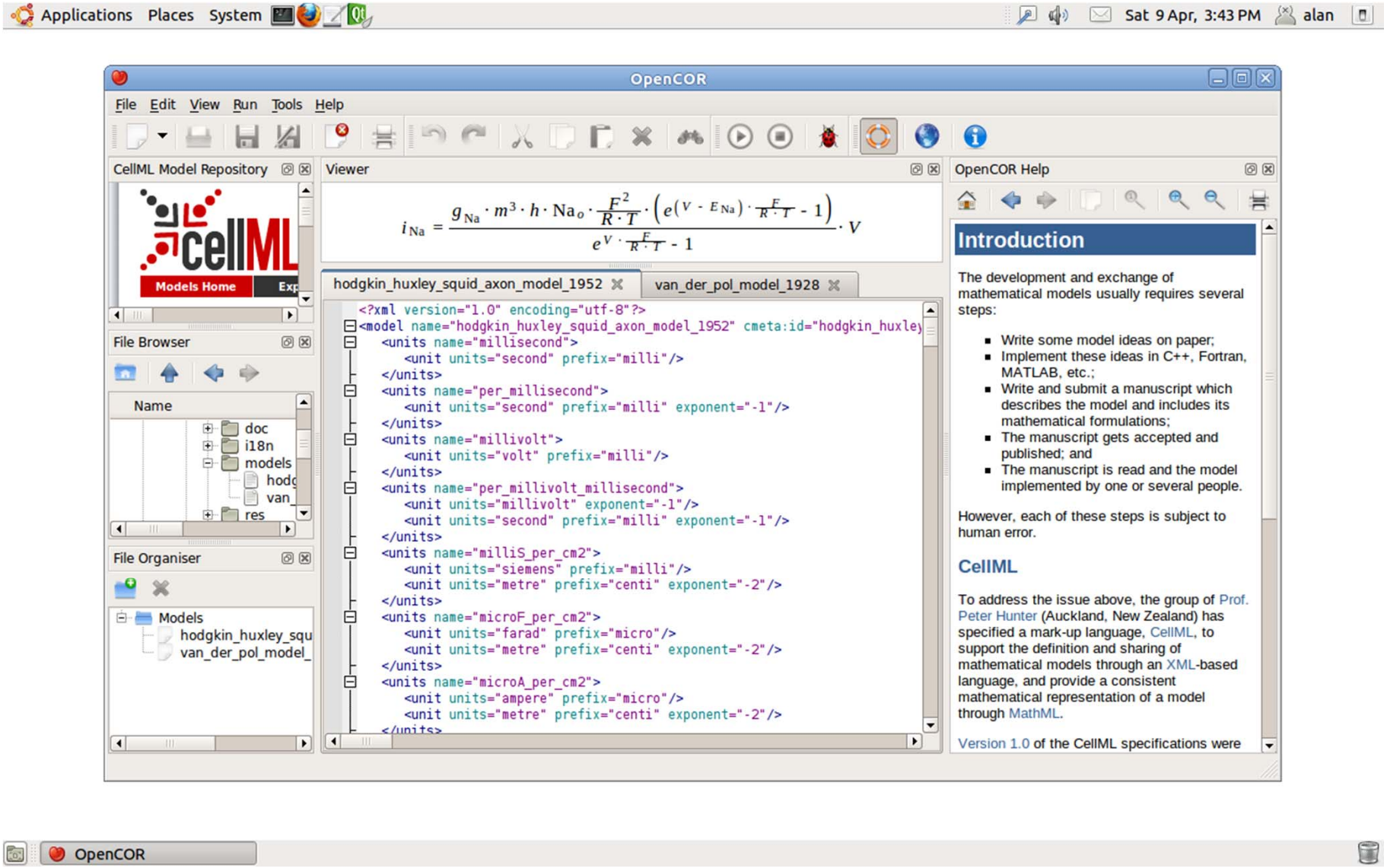
What OpenCOR Currently Looks Like...

... on Windows



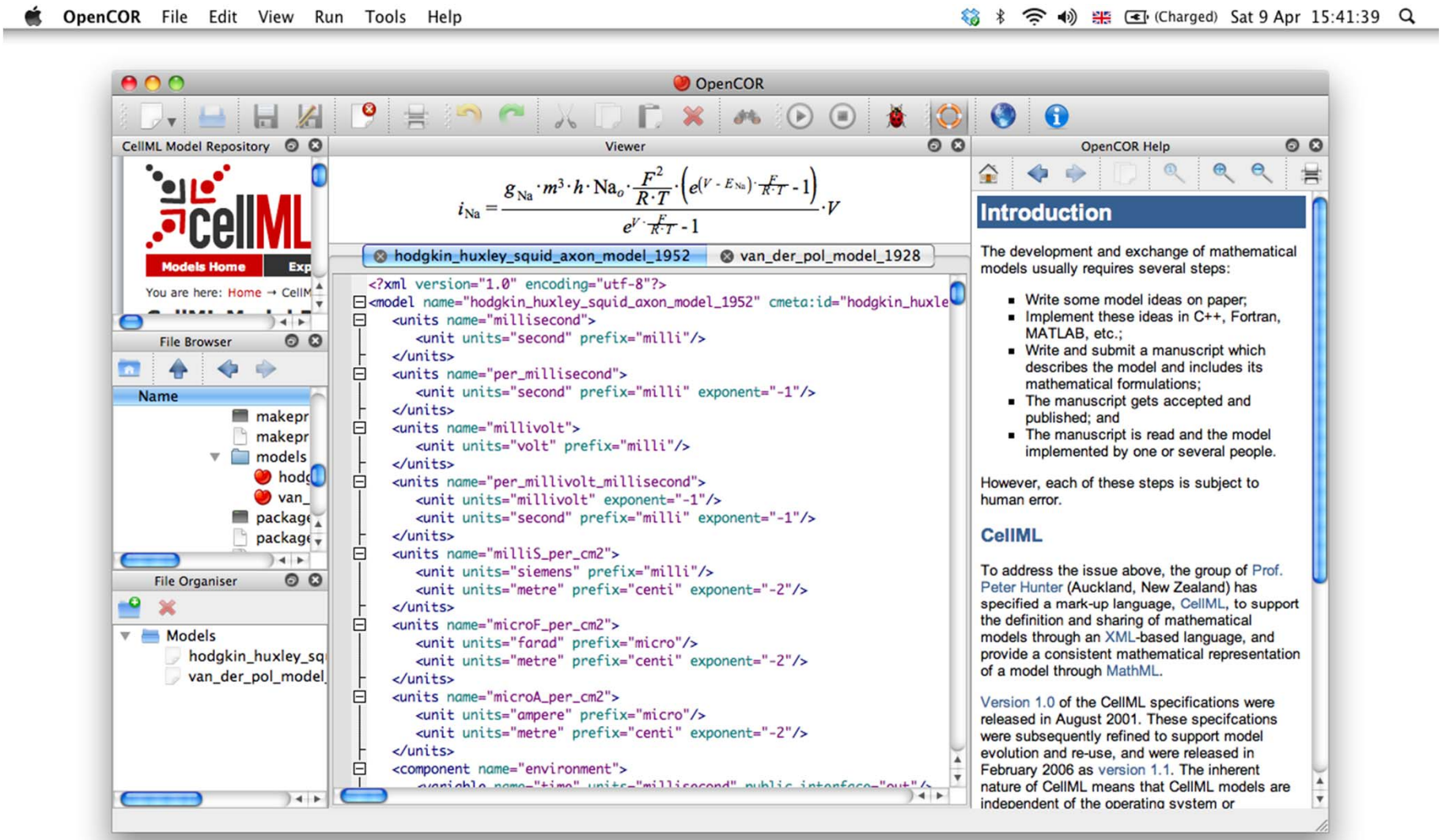
What OpenCOR currently looks like...

... on Linux

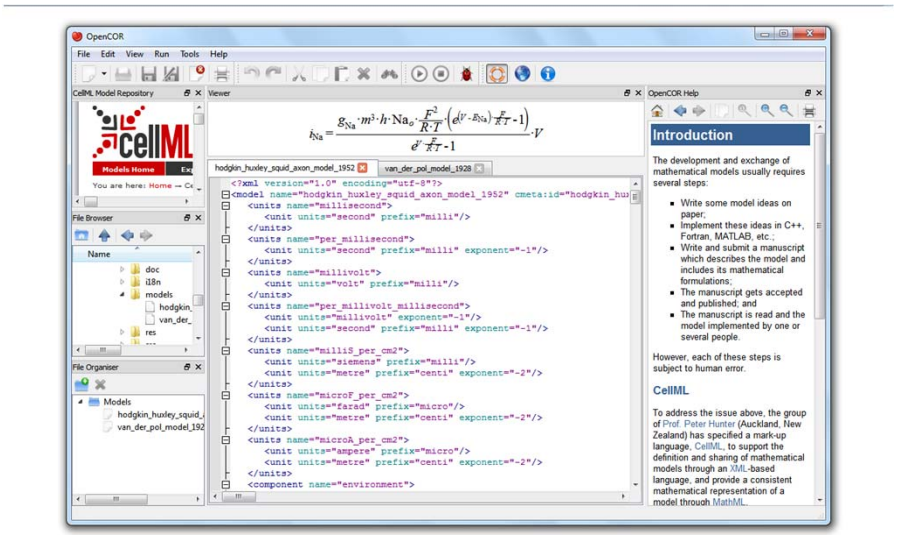


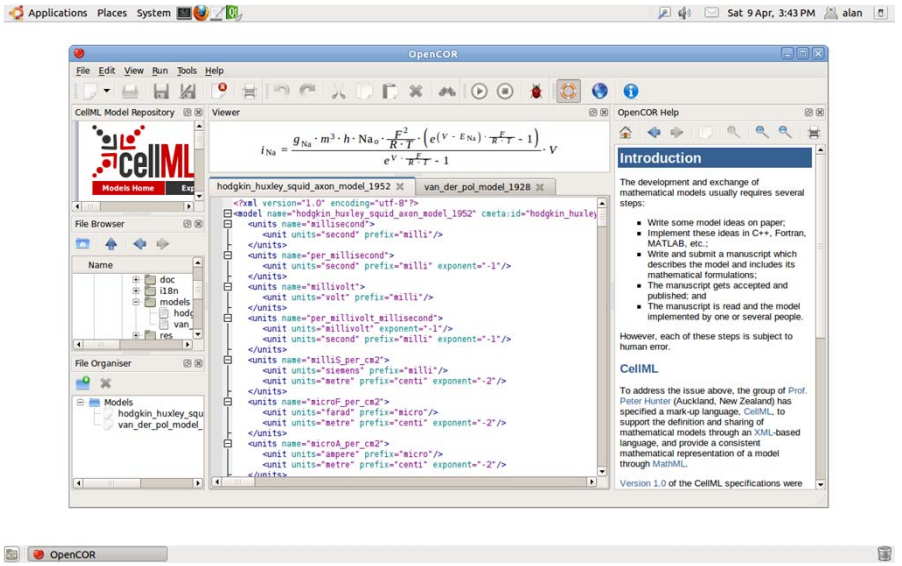
What OpenCOR currently looks like...

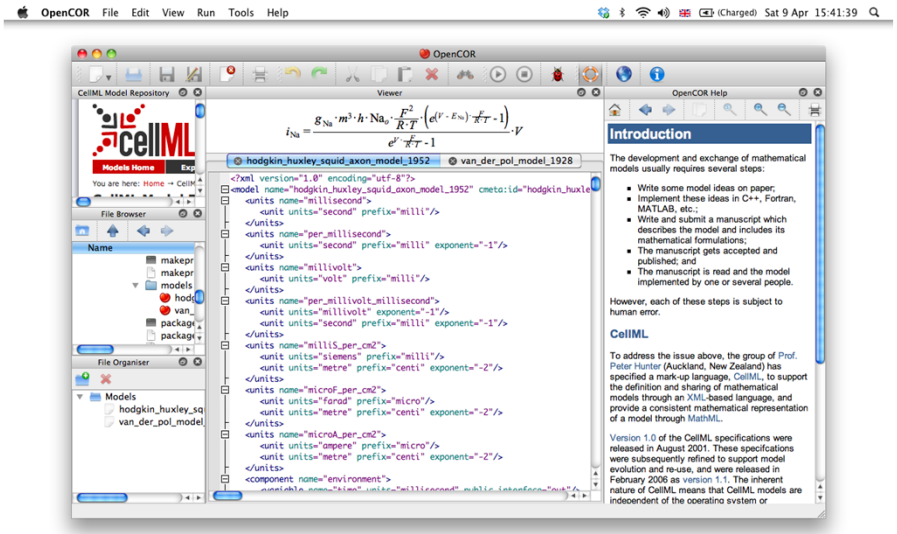
... on Mac OS X



What OpenCOR currently looks like…







Still aiming at a first public release by the end of the year…

ACKNOWLEDGEMENTS