CellML 2010 Workshop
CellML 1.2
and future CellML/SBML interoperability

Poul Nielsen

AUCKLAND BIOENGINEERING INSTITUTE
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

CellML interoperability

- Interoperability with external systems has, until recently, been severely hampered by the lack of a quality API.
- The CellML API is reasonably mature, thanks to the excellent efforts of a number of people – especially Andrew Miller.
- A local driving force has been to meet the needs of OpenCell.
- However, the API is not widely used despite the availability of many useful services:
  - Core reading/writing/querying/modifying models;
  - CellML validation;
  - Translation tools;
  - Simulation tools.
- Use of the CellML API should be encouraged and supported.

CellML/SBML interoperability

- Most translation tools are based on XSLT, are old and buggy, and cannot handle CellML 1.1 models.
- BioModels handling of CellML models has highlighted the need for reliable and accurate translation between SBML and CellML.
- Recent moves within EBI seem to be addressing this need – building a translator based on libSBML and CellML API (Nico Rodriguez).
- We need to offer full support (and participation) of CellML community for this effort.

Language evolution

- CellML 1.1 has been a stable specification since 2006.
- It is a conceptually simple language that has sufficient expressiveness to represent a wide range of models.
- A number of changes for CellML 1.2 have been proposed, but none have been finalised sufficiently to warrant an imminent update to the specification:
  - Removal of the reaction element;
  - Addition of variable typing
    - Scalar: real, integer, boolean, rational, complex.
    - Structured: set, list, array, matrix.
    - User-defined: record, dynamic.
  - Provision for shift/delay functions (and events);
  - Enabling stochastic variables and probability density functions (PKPD).

Removal of the reaction element

- The removal of the reaction element has been signaled for many years.
- CellML has been designed as a generic declarative model specification language expressing the mathematical relationships between entities, free of domain-specific semantics.
- The reaction element is unusual because it has unique biological semantics that can be better expressed using metadata.
- Models using the reaction element are being removed from the CellML model repository, replaced with refactored versions.

Variable typing

- Some variables would be expressed more cleanly using scalar typing (e.g. non-negative integer types for stoichiometry)
- The addition of structured types would enable the specification of topologically dynamic models, and provide more natural representation of tensors.
- Despite support for different scalar and structured types in MathML, there exist a number of issues with adding variable typing to CellML:
  - We have no simple and clean mechanism for specifying variable typing that integrates well with CellML's units;
  - There doesn't currently appear to be a sufficiently compelling need, for adding a typing mechanism, to warrant the technical and political pain that would result.

Shift/delay and events

- Events are able to be accommodated using the piecewise function defined in MathML.
- Over the past year an infinitessimal shift/delay function has been tested. This enables variables to be reset according to rules defined in a piecewise function.
- The addition of finite a shift/delay function, although conceptually simple, has not yet been tested as the current integrator is not designed to handle such relationships.
- We need to add/test this functionality soon as there are many models that have finite shift/delay expressions.

Stochastic variables

- The pharmacokinetic/pharmacodynamic (PKPD) community has a need to represent models that involve stochastic variables.
- Such variables are typically expressed in terms of probability density functions, or are statistically derived from external databases.
- Should CellML (and SBML) provide mechanisms to accommodate stochastic variables, or should all relationships be defined exactly and stochastic variation be accommodated externally using, for example, simulation experiment description language (SED-ML)?

CellML 1.n (where n > 1)

- How do we know when to draft/accept new specifications?
  - Andrew Miller has drafted a CellML umbrella specification, separating normative from informative content.
  - Currently have distributed version control system to allow multiple specification candidates to be independently/collaboratively developed.
  - We have in place a well-defined workflow for proposing, evaluating, and accepting candidate specifications.
  - Members of the CellML community are encouraged to participate in this process by developing secondary specifications and submitting them for peer evaluation.


cellML


AUCKLAND
BIOENGINEERING INSTITUTE
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND