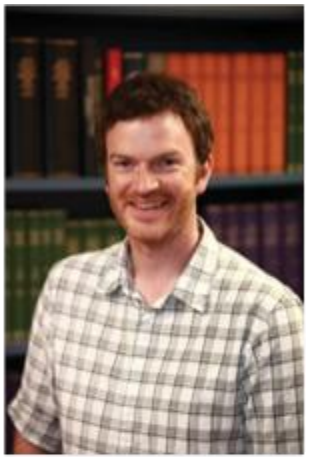
Toward CellML 1.2

The CellML Editorial Board

CellML Editors

Poul Nielsen
2011-2014
NZ

Edmund Crampin
2011-2012
NZ



David Nickerson
2011-2014
NZ

Alan Garny
2011-2013
France

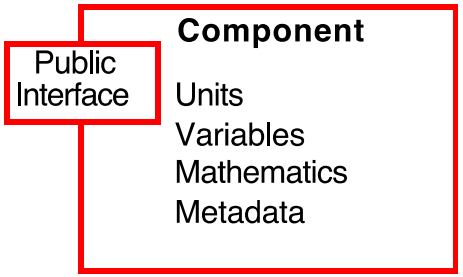
Jonathan Cooper
2011-2013
UK

A brief overview of CellML as it is

- CellML is designed to support the definition and sharing of (lumped parameter) models of biological processes.
- CellML includes information about:
  - Model structure (how the parts of a model are organizationally related to one another);
  - Mathematics (equations describing the underlying biological processes);
  - Metadata (additional information about the model that allows scientists to search for specific models or model components in a database or other repository).
- A public repository of over 500 published signal transduction, electrophysiological, mechanical, and metabolic pathway processes is available at *http://models.cellml.org/*.

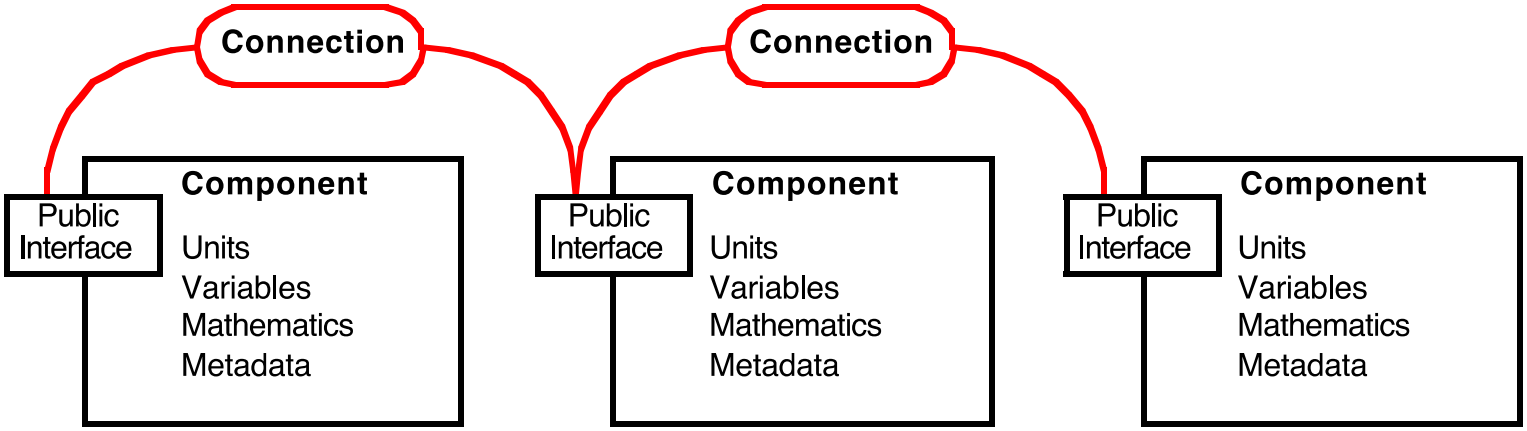
CellML components

- CellML has a simple structure based upon connected *components*.
- Components abstract concepts by providing well-defined interfaces to other components.
- Components encapsulate concepts by hiding details from other components.



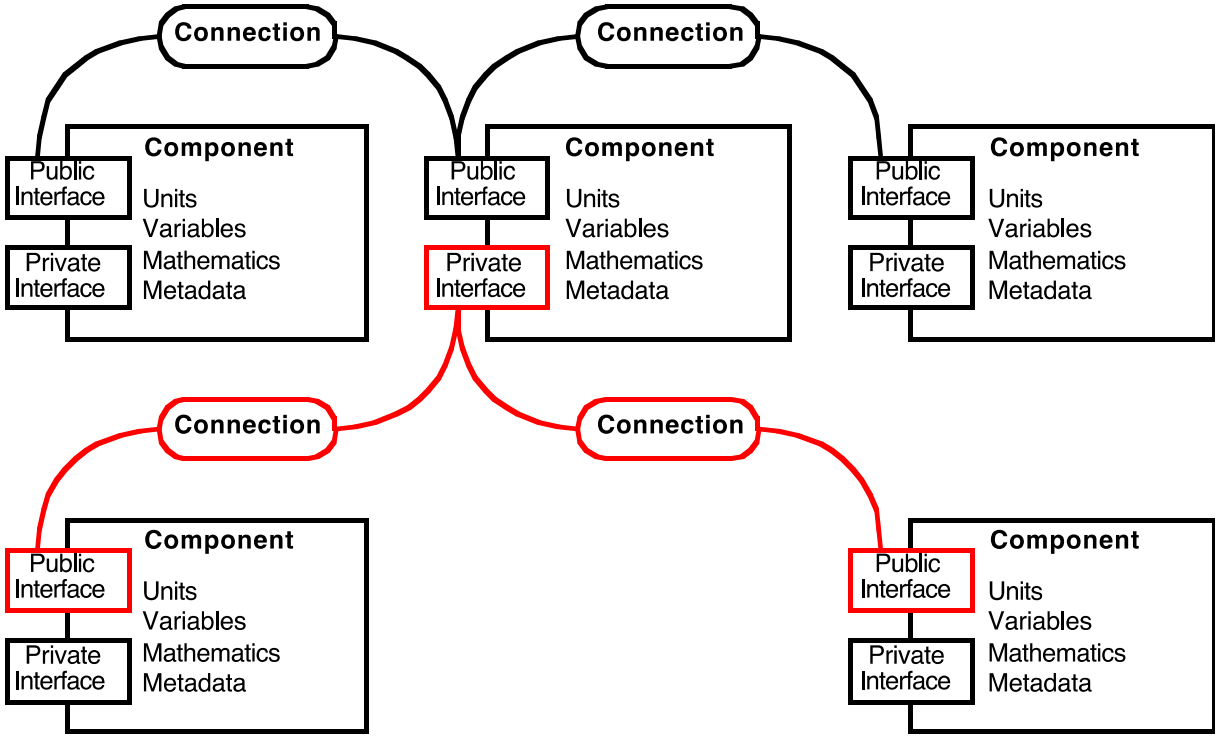
CellML connections

- *Connections* provide the means for sharing information by associating variables visible in the interface of one component with those in the interface of another component.
- Consistency is enforced by requiring that all variables be assigned appropriate physical *units*.



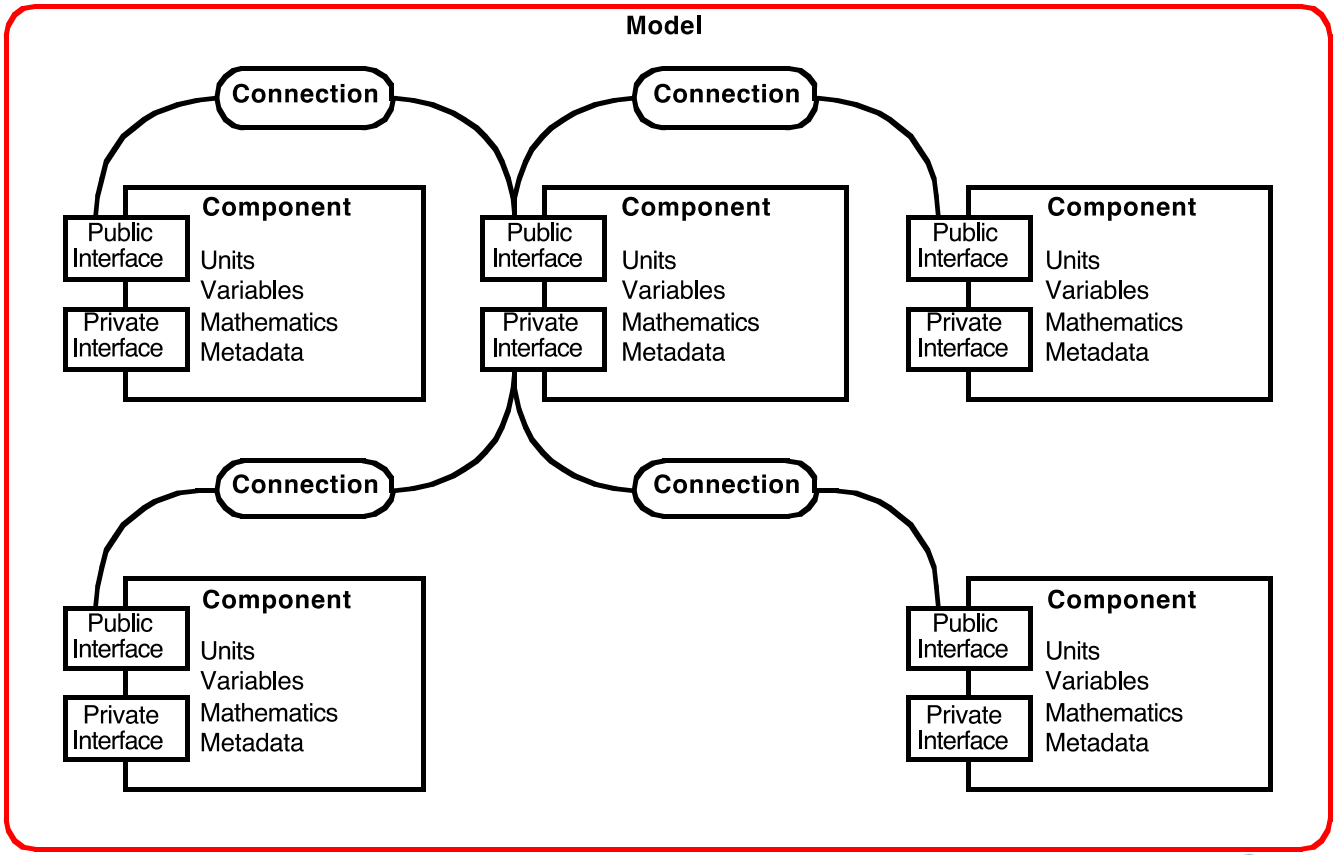
CellML encapsulation

- Encapsulation hierarchies are enabled using *private interfaces*.



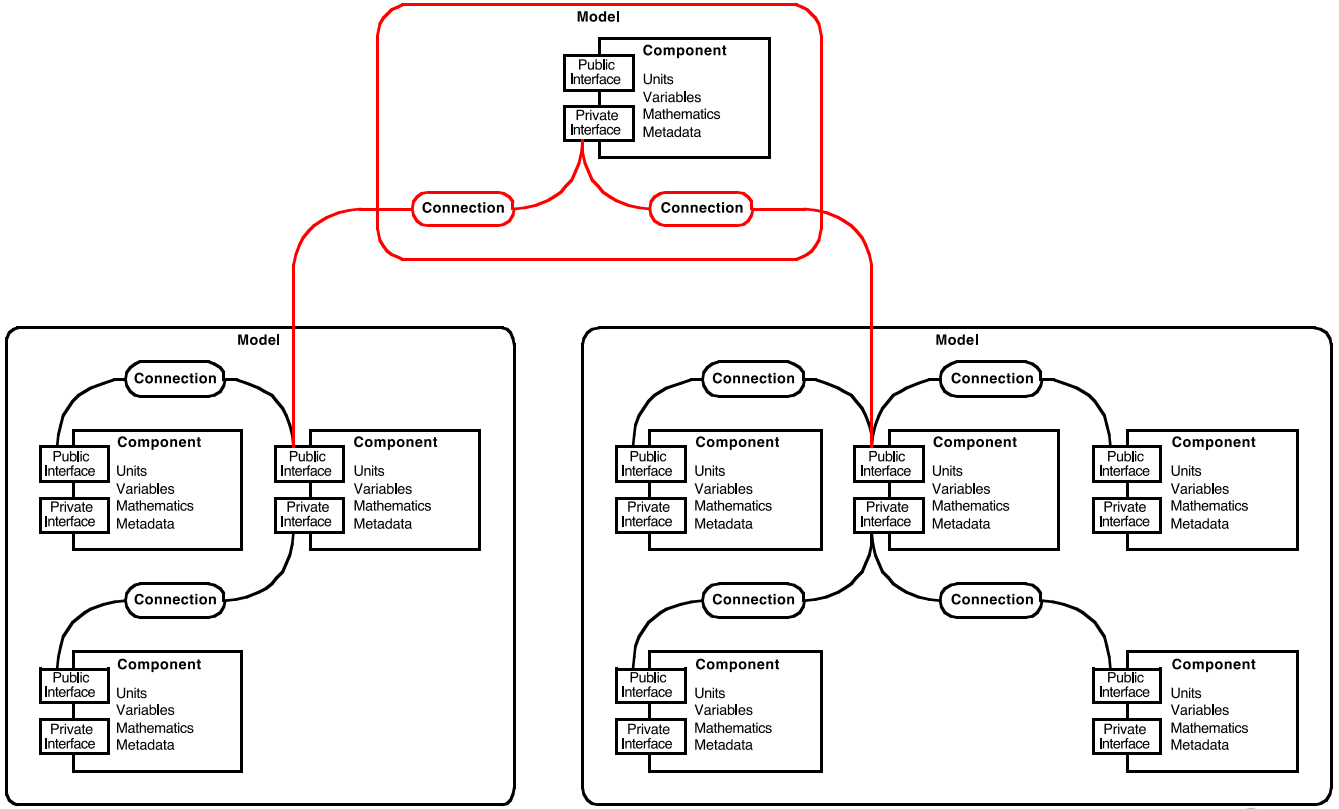
CellML model

- A *model* is the root element for a CellML document. It is a container for components, connections, units, and metadata.



CellML import

- Model reuse is enabled by the *import* element.



Editors' vision for CellML 1.2

- Tidy up the specification (normative + informative).
- Establish the process for future development.
- Incremental change from CellML 1.1:
  - Directly addressing current user's needs.
- Timely software support:
  - Need for API support;
  - Ease the burden on tool developers to support CellML 1.2.
- Grounded in concrete, real world, use-cases.

Editors' recommendations

- All that follows is agreed upon by the editorial board.
- We are now putting this out to the community for discussion.
- Still some open issues on which we are actively seeking input from the community.
- Nothing set in stone (including the previous slide!).
- We will hold a vote on the proposed scope of CellML 1.2 once consensus has generally been achieved.

'Minor' changes

- Remove the directionality of connections.
- Simplify the encapsulation mechanism.
- Remove the reaction construct – the final bit of biology hanging around in the language!

- Various miscellaneous clarifications in the language of the specification, including:
  - Produce a 'normative' version with concise technical details; and
  - An 'informative' version with more explanation and demonstrations/ examples.

Core + secondary specifications

- CellML 1.2 Core specification:
  - Basic concepts;
  - As permissive as possible;
  - Foundation for future versions of the specification.
- "CellML 1.2" includes a collection of secondary specifications which place restrictions on what is valid to express in CellML 1.2 Core.
- Software support, for example:
  - Editing and visualisation tools likely to support the core specification;
  - Simulation tools only able to support some secondary specification(s).

Example: Mathematics

- Core specification: any MathML can be expressed.
- Secondary specification: only index-1 DAE systems and algebraic expressions consisting of this set of MathML operators are valid.
- "Fixes" current ambiguity regarding the CellML subset of MathML.
- Provides a restricted set of mathematical expressions which simulation tools can reasonably be expected to support.

Example: '`evaluatedAt`' operator

- A new operator (`csymbol`) introduced in CellML 1.2 Core to evaluate a variable at a specific "time".
- Allows the concept of delayed differential equations, infinitesimal delays, etc. to be expressed.
- CellML 1.2 Secondary restricts the usage of this operator to:
  - Infinitesimal delays (reset rules, like SBML events);
  - Setting initial values (i.e. all non-infinitesimal occurrences of `evaluatedAt` operator refer to the same value of "time").

MathML 3

- Several 'issues' in MathML 2 that lack semantic clarity:
  - But they are well understood in regard to CellML (and SBML, SED-ML, etc.).
- Semantic clarity achieved by making use of MathML 3:
  - Normative specification can be more precise.
- It is possible to translate MathML 2 into non-strict MathML 3 – and to restrict the use of non-strict MathML 3 such that it can be translated to MathML 2.

MathML 3

- MathML 3 provides greater flexibility for the future – but is it too early to change?
  - Interoperability with SBML, SED-ML, etc.;
  - Tool support;
  - Cleaner semantics for extending with OpenMath content dictionaries and types.
- Example: semantics of an ODE

```
<apply><diff/><bvar><ci>t</ci></bvar><ci>V</ci></apply>
```

≡

```
<apply>
  <apply><csymbol cd="calculus1">diff</csymbol>
    <bind><csymbol cd="fns1">lambda</csymbol>
          <bvar><ci>t</ci></bvar><ci>V</ci></bind></apply>
  <ci>t</ci>
</apply>
```

Future development

- Normative specification provides explicit points of reference for future proposals.
- Adoption of COMBINE governance and development guidelines (once they exist):
  - For now, we are trying to move into line with SBML processes.

Future development

- Secondary specifications can be developed for any purpose:
  - Tools can be unambiguous about their support for "CellML";
  - Popular and well-supported secondary specifications likely candidates for adoption into official CellML specifications.
- Timely method for easing restrictions placed by previous specifications in a compatible manner:
  - DDEs are not allowed in CellML 1.2, but may be allowed with no change required to the Core specification.

Useful links

- http://www.cellml.org/.
- Proposals for CellML 1.2 – Physiome tracker item 55.
- Unclassified proposals for future versions of CellML – Physiome tracker item 2886.
- CellML mailing list: http://lists.cellml.org/mailman/listinfo/cellml-discussion.

Acknowledgements

- The CellML community for all their input.
- The CellML team at the Auckland Bioengineering Institute:
  - Especially Andrew Miller for all his work setting up the infrastructure for the normative specification.

- Funding via: VPH-NoE, VPH-SHARE, 2020 Science.