

Meeting Minutes 25 August 2000

Author:

Warren Hedley (Bioengineering Institute, University of Auckland)

Contributors:

Poul Nielsen (Bioengineering Institute, University of Auckland)

David Bullivant (Bioengineering Institute, University of Auckland)

1 Summary

Some major advances were made at the August 25 meeting. The most important of these related to the definition of ontology parallel to model definition and the links between the two. The CellML data model was also expanded on (although it is still very simple). We also added one more item to the graphical notation standard for diagrams.

2 Graphical Notation

I'd left out one critical component from my proposal for a graphical notation that first appeared in the August 24 meeting minutes: namely a shape for representing components that occur inside subspaces or boundaries (the mechanisms of CellML '99). Some examples of these kinds of components are the channels and pumps that you find inside a membrane-type boundary. As we were running out of simple shapes, we decided to resort to a hexagon for rendering these components.

3 Ontology

The August 25 meeting was notable for an agreement on the method in which ontology definition could be successfully merged with the existing network/component/connection data model.

3.1 What Is Ontology?

The title of this section should probably be "what do we mean by ontology in the context of CellML", but that doesn't fit on one line, so I had to abbreviate it. Ontology generally refers to theories about existence, and is really a branch of philosophy, but the concepts have been adopted in the fields of software engineering and data modelling for describing the many forms of data that might occur when tackling a particular problem.

As far as cellular modelling is concerned, we are more interested in defining a set of "**component-classes**" and "**connection-classes**". (These classes can also be thought of as "**templates**" or "**type-definitions**", but I'm going to refer to them as classes from here on in.) Components and connections in the model can then associate themselves with the classes defined in the ontology: each component associated with a particular class is known as an **instance** of that class. A class definition may contain a set of properties that all instances of that class automatically inherit (unless otherwise specified), and also define a set of properties that an instance may not possess. Most importantly, the total set of properties that a component or connection possesses is the union of the properties defined in its class definition (if they exist) and any properties defined in the instance of the component or connection that appears in the network.

3.2 Why Have Ontology?

The fact that the actual structure of the model can be stored in either the ontology (the collection of class definitions) or in the network (the actual model definition) in an arbitrary way allows CellML processing software some flexibility in its handling of CellML models. Consider the following cases:

- A model can stand alone without any ontology information attached. This may be convenient for exchange but doesn't allow for the re-use of information common to components and connections of similar class (although re-use could easily be done using some generic scheme like FieldML's CRIM, or a low-level scheme based on XML entities).
- A model is defined in terms of some standard component library, where the class definitions contain no properties but are used by processing software for the purposes of rendering. Some restrictions on how components of differing classes can interact may be defined. For example, an ontology could specify that a component of class membrane could not be physically inside a component of class channel.
- All of the properties of a model are stored in the ontology, with a class definition for every component and connection in the model, with each class definition containing all of the information needed to run the model. The actual model definition just contains an instance of each of the class definitions and no additional information.

3.3 Advanced Features

The concepts in the ontology section of CellML closely resemble those in object-oriented programming. More advanced concepts from this field can also be introduced into CellML that dramatically increase the power and usability of CellML. The most obvious of these is **inheritance**. If one class definition is identified as **extending** another, then it automatically obtains all of the properties that the defined in the extended class. For instance, a *cellular-membrane* class might extend a *membrane* class, obtaining all of the variables and equations defined in the membrane class definition, and adding some of its own. Some mechanism for not inheriting properties, and for resolving naming conflicts may be needed.

In object-oriented programming, the only thing more important than a class-definition's properties are its **methods**. Assuming that CellML must be able to incorporate blocks of computer code or scripting language, it may be appropriate to add method declarations to class definitions, where the method body contains code or script that operates on the variables declared in the class definition or variables obtained explicitly from an undisclosed location.

3.4 Defining An Ontology

A simple and elegant way to define a set of class definitions is to use the data models for the components that we are "templating" to form the template. That is, we can use the data model for a component where it appears in a model, as the basis for a component class definition as it appears in the ontology. Inside of an ontology the *class* property of a component or connection (see data model in Section 4) identifies the class definition. It must be present and unique within the set of all *class* properties defined in the ontology. Inside a model definition, the *class* property of a component or connection associates the current object with the class definition of the same name in the ontology.

More on ontology definition in the next set of meeting minutes.

4 Data Model

The August 25 meeting marked the first real formalisation of the basis for the CellML data model. The result is shown in Figure 1.

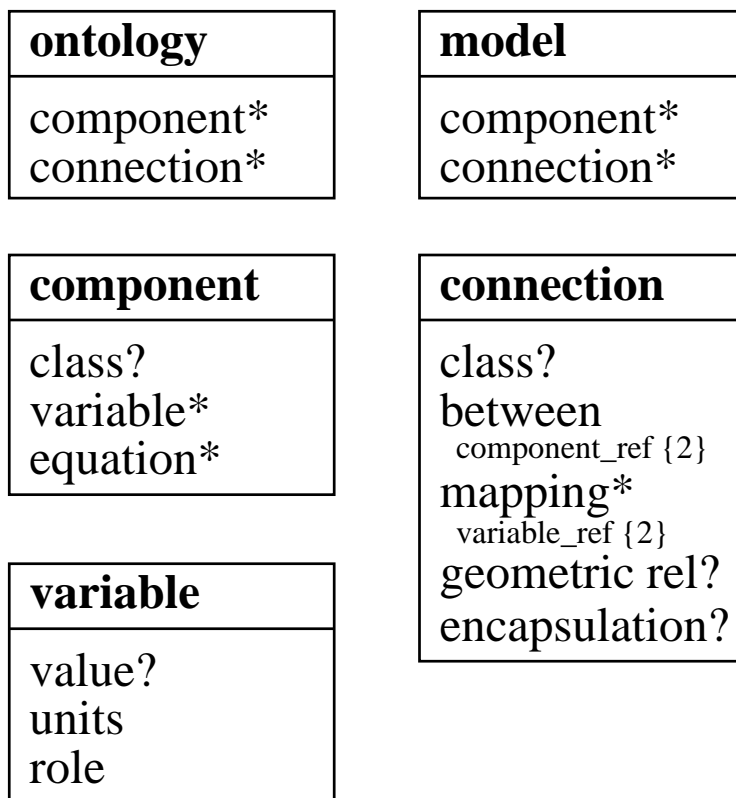


FIGURE 1: A starting point for the CellML data model, with the key complex data types and their basic properties. This diagram is not in UML and doesn't attempt to conform to any particular standard. After some properties are question marks (?), asterisks (*) and braces ({}). A question mark means that a property is optional: zero or one of these properties may be present in the object. An asterisk means that any number of these properties may be present. A number in braces means that the property should be present exactly that many times. If no symbol is present, exactly one of the properties must be present. See text for more details.

There are some constraints on available relationships between objects in the data model that are best expressed in words:

- Connections may be **directed**. Where geometric (in particular the “*is-in*” relationship) or encapsulation information is present in the connection, the first component referenced in the *between* property of the connection assumes a “*child*” role in the connection. That is the first component *is-in* and/or *is-encapsulated-by* the second component. This means that the specification of geometric or encapsulation relationships is reduced to the presence of a flag in the connection, rather than a more complex specification using component references. It also restricts the user from specifying potentially inconsistent geometric and encapsulation relationships.
- To make the traversal of the network more efficient and prevent some possibilities for setting up inconsistent or ambiguous relationships between components, there can only be one connection between

any two components in a network. All relationships between components (mappings, geometric information, etc.) must be in this connection.

E-mail questions, criticism, submissions or info to info@cellml.org
Input document last modified : Mon Feb 02 15:25:01 NZDT 2004