Meeting Minutes 12 September 2000

Author:
Warren Hedley (Bioengineering Institute, University of Auckland)
Contributors:
David Bullivant (Bioengineering Institute, University of Auckland)
Yi Ge (Physiome Sciences Inc.)
Kam-Chuen Jim (Physiome Sciences Inc.)
Scott Lett (Physiome Sciences Inc.)
Jian Li (Physiome Sciences Inc.)
Poul Nielsen (Bioengineering Institute, University of Auckland)
Hao Yu (Physiome Sciences Inc.)

1 Summary

The 12 September was a pretty eventful day at Auckland with a teleconference with the software team at Physiome Sciences, and three meetings at Auckland. During the teleconference, some possibly contentious issues were sorted out and some priorities for further development were set. During the three meetings the proper specification of non-mathematical (i.e., qualitative) pathway models was discussed, along with the implications for component grouping and the need for a role of `in-out` on variables.

2 Teleconference Issues

Warren's major questions for the Physmos are given in the following list. Unfortunately Melanie answered most of them via E-mail before the teleconference even started. The list also contains conclusions reached as a result of the E-mail and teleconference.

- **Will there be a meeting of the CellML working group to coincide with the BMES meeting in Seattle?** No, some legal issues must still be overcome before the working group can sit down together.
- **Requirement 10: What does it mean by represent "biological information"? Does this mean non-modelling information? If so, isn't this metadata? Similarly for "non-geometric information".** "Biological information" refers to biological things that modellers need to be able to represent in their models. For example, in a signal transduction pathway the modeller may know that one protein's activity is modulated by another, but not how. CellML must be able to represent this information. More on this in Section 3.
- **Requirement 10: Do we still require spatial variation in CellML V1.0?** We can use MathML to express spatial variation until FieldML is better defined. Bitmap geometries could also be supported, but this probably falls into the data category, so is not strictly in CellML's scope.
- **Requirement 11: Does anyone have any ideas about how they want to tackle stochastic math?** No.
- **Requirement 9: Does the example imply there must be a selective inheritance mechanism (not the original wording)?** Yes.
- **Requirement 7: Should variable scope be handled in the ontology rather than in CellML?** For instance, if someone wants to specify that the concentration of K+ in some subspace can only be used (ie. mapped) by the adjacent membranes, that's not something really belongs in CellML. It is not part of the model — it is a rule that belongs in the associated ontology definition.

Of course Physiome also had some questions/opinions on Auckland's work, which are summarised in the following list (taken from Melanie's summary of the teleconference.)

- The Physiome team were generally happy with the component/connection data model proposed by Auckland, and in particular the implication that reactions in pathway models were components (i.e., they have the same data model as subspaces or boundaries in an electro-physiological model.)
- Variables are by default private. The only way to share variables between components is to declare them with role `out` and explicitly map them in a connection.
- One of the major reasons for the existence of encapsulation is that a single component can provide a simple interface to a complex network. The only way to pass variables from the complex network to components on the level of the interface components is through the interface.

3 Qualitative Models

Debate still rages over this issue. This document merely reports our thinking on 12 September 2000. The question is: how should CellML describe qualitative information in pathway models, information like "*component* `B` *inhibits the transformation of component* `A` *into component* `C`". An engineer might well contend that this information is not part of a model (a model being math), but can be viewed as documentation, metadata, or classification information. Try explaining this to someone who deals in these qualitative models (or their colleagues) and they claim that this information is the whole essence of the "model". Auckland proposed relegating the "*inhibits*" relationship to metadata, documentation or ontology status, inferring that the importance of this information was in fact no more than saying that a particular component was a channel.

Unfortunately, between the 12th and the time that this document is being polished, Melanie went and submitted a fine document explaining exactly why qualitative models are models. This means I won't waste my breath here, trying to explain why they're not. The Auckland team are a pushover sometimes. Look for more on qualitative modelling in the minutes of the 14th.

4 The `in` to `in`, `out` to `out` saga.

A problem that has been taking up a lot of our time is getting variables to and from encapsulated components through an encapsulator without having to change the external interface of the encapsulator. Consider two subspaces separated by a membrane. The subspaces each declare a calcium concentration, and the membrane declares a calcium current. Some mappings are made to pass the variables around. The subspaces contain conservation of calcium concentration equations and the membrane contains a current calculation which depends on the concentrations in both subspaces.

Now we decide to pull a pre-packaged calcium channel out of a component library and insert it into our membrane, deleting our calcium current equation in the membrane. We don't want to have to change the rest of the model, so the new channel is encapsulated, and we want to use the membrane as an interface. The calcium current is now declared in the channel, and the current equation still depends on the concentrations in the subspaces. The concentration conservation still depends on the current. However the subspaces can't be connected to the channel.

The following solution was proposed by a member of the Auckland team to solve this problem — possibly the low-point of CellML development over the last month. In connections between an encapsulator and one of its encapsulated child components, variables with role `in` in the parent may be mapped to variables with role `in` in the child, and variables with role `out` in the child may be mapped to variables with role `out` in the parent. (Note that the order in which the variables are listed here reflects the direction of the flow of the variable's value.)

This solution could be described as counter-intuitive and inelegant among other things. I've been going on about having an `in-out` or `through` role for some time, but others here like to point out that if I have to change the role of variables in the membrane, I've essentially changed it's external interface, which kind of defeats the purpose of encapsulation.