

---

# Meeting Minutes 30 April 2004

Autumn A Cuellar

- Poul's update: Andrew Finney from the SBML [<http://sbml.org/>] group e-mailed saying that they've used Maria Schilstra's XSLT scripts to translate several of the CellML models to SBML and run them. Andrew was wondering how we felt about him making those models available from the SBML model repository. Poul thinks it's fine as long as proper acknowledgement is given to Catherine. Carey mentions workflow - is feedback about the models going to get back to us? Shane suggests we look into the integration of model repositories.
- Autumn's update: I've made the changes to the 6 April Meeting Minutes [20040406\_meeting\_minutes.html] suggested by Matt. I've also added a model [../examples/repository/smith\_crampin\_Na\_K\_pump\_2004\_doc.html] coded by Alice Boit to the repository.
- Matt's update: Has sent a whole bunch of changes to Mike Lovell-Smith to be made to the Anatomy Ontology. Has students working on visualizations to help people visualize imports. They are building on the libraries that Hadley wrote when working on the pathway renderer. Matt gave them the option of wrapping those libraries in C++ or Python, but they've chosen to work only in Java.

Matt also has been working on instantiating imports in Python (used to have to go to the separate model to visualize/edit), allowing a shallow copy first, then a deep copy. He wrote an XSLT script that allows visualization of imported components and variables and the connections between them. The XSLT script was mostly for the students to show them how the visualization tool will have to apply filters because it can get pretty murky when you are visualizing a model that imports from a model that imports from another model.

A script for visualizing metadata is in development, will handle model annotation. It will help Matt suss out what needs to be added to ontologies to be useful to the CellML repository.

Matt has also started playing around with translating the ontology to OWL format and using inference tools on it.

- Shane's update: No response to the discussion e-mail [<http://www.cellml.org/pipermail/cellml-discussion/2004-April/000003.html>] about the FieldML Road Map. Regarding comments on the CellML Road Map, see below (author's note: my notes on what little discussion was generated are contained in square brackets []).

## 1.2.1 CellML Repository

- Sync up instances that Cathrine made with the ontologies representation. (Import scripts?)
- Facilitate more than one person to work on the ontologies at once.
- Set up FieldML database and connect that to the ontology representation.
- Add script that enables multiple FieldML objects to be combined into larger objects on the fly (with AnatML this is all done statically).
- Add a FieldML to VRML filter that the server can run on the fly.
- Add pdb to FieldML filter to enable proteins to be loaded (see Connections to protein databases just below).

- What are the other steps to making Ontologies useful (I'm not sure what these are).....
  - i.e. implementation of reasoning
  - completion of hierarchies  
[development of all the things that can go into ontologies. admittedly, this is an ever-expanding list.]
  - useful ontology editing tools  
[Matt: When the ontology is exported into OWL, any OWL editor will serve this purpose. Shane: Okay, that's fine, but my point is that it seems like there's a lot missing here that should be fleshed out.]
- Connections to protein databases — Peter has Srdan working on this right now. He is adding static links to particular proteins. If we had a workflow he could be adding instances to our cellml database instead, adding to the work rather than working independent of it.
- What is the process for new models to make it onto the website....
  - i.e. has the pole zero constitutive law been worked on?
  - How does the cvs, database and website get updated with a new model.

[Shane: There's several people in this Insitute using CellML now. How do we make the process of putting their models in the repository simpler, allowing them to dump their models somewhere and encouraging them to do so. Matt: We should also give some thought to a useful namespace schema so people know how to name and reference their models. The URI of a model is its absolute form of identification. When you're doing deep imports from models, you quickly get a spaghetti mess of URI model references.

My current thinking is something that reflects more an author centric, or time centric space for the models. The ontology work gives us a few warnings about naming a model according to some kind of physiological properties; instances of models can have more than one parent from different domains, for example from physiological function and structure, and even multiple parents within one of these. So it is difficult to assign them a URI that is based of some rules about what types is may represent.

Anyway, once we have some rules for defining URI spaces, we can create the physical representation of them in the current repository, so that URI maps to physical location, and people know how to add their models to the repository through CVS.

In an e-mail exchange after the meeting, Matt brought up another issue to begin thinking about, best left in his words:

While talking about URIs, I wanted to mention one thing that we should clarify too; the relationship between the URI used in imports, and the cmeta:id and name attributes in the model element. In the spec, the [name](#) attribute is the one that “allows the model to be unambiguously referenced”, so perhaps we deal with this first. My feeling is that the URI used in the imports represents an unambiguous reference/id that can be used for 1) identifying the model you are importing, and 2) happens to be where you can find it if interpreting the URI as a URL. Should we include the whole URI in the model name? should we consider the use of xmlns:base to set a base URI from which we can reference the whole model, or parts of the model? Perhaps I could spend more time explaining what I mean in the next meeting.

]

- How does the database take account of variations in models for different species or if someone improves/modifies a model to get it to work differently after the paper.

[Bio-entity (in metadata) is too broad a category. Matt: That's just for simple typing, the ontologies are for more specific typing.]

## 1.3 Repository Visualisation

- A prototype web interface has been created by Matt. We could develop this a bit further, allowing easy navigation and incorporating access to the CellML, FieldML and image libraries. This could then replace AnatML.

[Matt brought up the problems with visualization that he'd been having, mentioned by Poul last week [20040423\_meeting\_minutes.html]. Regarding the problem with visualizing a tree in which a class is a subclass of more than one parent class, Shane thought it would be nice to be able to see the broader picture, as well. He asked about the possibility of querying for the shortest path from the current class to the root class. Matt mentioned that he'd started using RDF inference engines, which looked promising but required more work (full-scale evaluation that includes OWL inference tools, etc.). As Matt puts it, "We don't want to spend too much time trying tools that will become non-useful once we make the transition." So for the moment, Matt is going to stick to just rendering the super-classes.]

### 2.2.2 Visualisation with the CMISS (cmgui) cell simulator.

- I guess Andre told you that this no longer really exists. It could be revived but is currently not built into cmgui.

### 2.4.? Constitutive Laws

- CellML is being used to describe constitutive laws for CMISS (cm) problems. (Martyn, VJ, Andre, Holger, Kevin, Jae, Espen)

### 2.4.? Generalised CellML in CMISS (cm)

- Duane is using CellML to set the coefficients for the diffusion equation in CMISS(cm). He has done this by starting a generalised (non domain specific) Math importer for reading CellML into CMISS (cm). He has however only added his one problem type but presents an interesting pathway ahead.

### 2.4.? Computed variables

- David is developing a set of computationally differentiable functional objects. Initially these are being stitched together programmatically (via python, perl, C or Matlab) but CellML may provide a more accessible way to describe much of this. (although the different levels of aim, CellML is aimed at either a conceptual and/or mathematical description whereas computed variables are necessarily a formulational level so they can be machine implemented.)

## 2.? Simulation Visualisation

- CMISS (cmgui) is able to generate iso surfaces, glyphs and so on representing spatial geometry and solutions of computational problems.

[Provides a direct link to CMISS from the Road Map.]

## 3.1 Development of API implementations

- I am personally convinced that it is a large waste of resource to have different implementations of an API for different languages. I think we need a plan for a convergence of the API's we already have for CellML and a plan for migrating existing uses in the institute to whatever is new.

## Miscellaneous thoughts

[Someone mentioned that changes to the Road Map happen rapidly. How do members of Team CellML update the Road Map? It's already stored in CVS. Autumn, send instructions on how to check out the Road Map and other web pages from CVS, and how to edit pages. Also, work on displaying a development version that's different to the Road Map posted on the Projects page.]