

Meeting Minutes 29 January 2001

Meeting With the SBML Team At Caltech

Author:

Warren Hedley (Bioengineering Institute, University of Auckland)

Contributors:

Hamid Bolouri (Erato Systems Biology Project, Caltech)

Andrew Finney (Erato Systems Biology Project, Caltech)

Melanie Nelson (Physiome Sciences Inc.)

Herbert Sauro (Erato Systems Biology Project, Caltech)

1 Introduction

This document provides a summary of the discussion between representatives of the SBML development team (Hamid Bolouri, Andrew Finney, and Herbert Sauro) and the CellML development team (Warren Hedley and Melanie Nelson) that took place at the SBML team's Caltech headquarters on January 29 2001.

Andrew Finney had thoughtfully prepared an agenda for the day's discussion, which was pretty much ignored by all present. However the focus did meander through the topics on that agenda, which included:

- Warren introducing recent changes to and future plans for CellML.
- Melanie presenting an overview of Physiome's activities, focusing on biochemical network modelling and emphasizing Physiome's commitment to CellML.
- Discussion of translation from CellML to SBML and vice versa.

This document has a section for each of the points above, as well as a summary of some current and future SBML features not currently supported in CellML, and a proposed passage of text for the website discussing the relationship between the two languages.

2 Changes To CellML

The first item on the agenda was a summary of recent changes to CellML from Warren. Being incredibly thoughtful (not to mention sensitive), Warren had brought several copies of the parts of the CellML specification pertaining to units, reactions and metadata for review by the SBML team. Units and reactions were discussed in detail, but metadata was left for another day — let's face it, no-one but database geeks care about metadata anyway.

2.1 The Units Specification

One of Herbert's initial comments on the units specification was quite surprising: "Why not use MathML to define the relationship between new units and a set of pre-defined units?" This hadn't really occurred to the Auckland team, as it had seemed most important to maintain compatibility with SBML in this area. However it was agreed that, given the equations in Section 2.3 of the units specification, the relationship between complex and simple units could be defined unambiguously. Using MathML would allow completely general and arbitrary combinations of units including offsets specified in either the simple or complex units, but this approach would make consistency checking, and the transformation to simpler languages a nightmare.

The main differences between CellML and SBML in the area of units definition are the addition of the **multiplier** and **offset** attributes to the `<unit>` element; the remaining differences are purely

in element and attribute naming. The SBML folks saw the benefits of having a **multiplier** attribute and hinted that this might be added to SBML before too long. There were more objections however to the **offset** attribute, which introduces significant complexity given the rules that must be defined to control its use.

Andrew wasn't too worried about this however because he had assumed that software would barf if two variables with different units were equated (i.e., he wouldn't have to worry about converting **fahrenheit_inches** to **celsius_metres**). The CellML specification hints that software should be able to handle a conversion like this because the units share the same dimensionality.

2.2 The Reactions Specification

The information described by the CellML syntax presented in the reactions part of the CellML specification roughly corresponds to the information that can be defined using SBML's syntax. The SBML team had no real problems with the CellML approach to modelling and the notation other than that its complexity was overwhelming. It is likely that a good guide needs to be written that gently introduces the CellML approach to reaction modelling before getting into syntax specifics.

The single biggest difference between SBML and CellML is the specification of kinetic rate laws. SBML defines reaction rate using Infix notation placed inside an attribute, or with references to a common library of rate functions. CellML defines reaction rate equations using a complete MathML definition.

Overall, CellML's syntax is more general and flexible, giving the modeller much more powerful functionality. However, the functionality presented in the examples part of the reaction specification is equivalent to that provided by SBML. The areas in which CellML provides more functionality than SBML include the ability of CellML to:

- Define arbitrary relationships between “*delta*” variables and reaction rate using MathML
- Explicitly define stoichiometry of reaction participants other than reactants and products (such as inhibitors).
- Explicitly associate a participating variable with a role in a specific reaction direction.

3 Presenting Physiome

One of the topics for discussion requested by the SBML team some time before the meeting was a talk from Melanie about Physiome's activities, and in particular, their plans for reaction/pathway modelling software. Melanie had some slides prepared by Adam Muzikant at Physiome Sciences and answered questions from the SBML team members.

Perhaps the most surprising thing to come out of this discussion was the discovery that the SBML team had not considered Physiome to be taking CellML very seriously. They in turn were surprised to find out that CellML was important to Physiome. This indicates that Physiome needs to emphasise more the role of CellML as an exchange format in its presentations, and not just mention it in passing.

As previously mentioned, the main obstacle to the adoption of CellML is the difficulty of reading in MathML. It was suggested that Physiome make software available that could convert MathML to other formats — the preferred format would be Infix mathematics as far as the SBML team is concerned. This would make it possible for software that reads SBML to read CellML without too much trouble. Warren suggested providing a library that converted MathML into some standard tree form, where the API used to traverse this tree would have as much in common as possible as that used to traverse an Infix tree (obviously MathML's extra functionality would need to be handled specially). The SBML guys were particularly keen on seeing open source code but not overly optimistic.

To continue co-operation between the SBML team and the CellML team, it was suggested that the SBML folks visit Physiome, perhaps in conjunction with a SBML workshop in Washington in May. The SBML team should be given an opportunity to see Physiome's pathway editor — they were quite interested in how the graphical representation of pathway models would be handled, as this is an area where there is very little standardisation. They are also interested in seeing if there would be any way to make functionality of Physiome software available to other packages using the Systems Biology Workbench. This would involve making an API or external interface for some components of Physiome's software publicly available.

Another way to ensure that the SBML team can continually have an overview of and input into the CellML project is to invite the project co-ordinator Hamid Bolouri to have a seat on the CellML Advisory Board. It is important that both development teams have as much information available to them about the other project as possible, so that they can give correct information when asked about how the projects relate and interact.

4 Translation Between CellML and SBML

Most of the afternoon was spent considering how a simple CellML model (the [two reaction model](#)¹ from the CellML website) could be translated into SBML. It was clear that some information about a reaction would be lost in the translation, as SBML does not capture explicitly assign role or direction information for species. It was suggested by Warren that such a translation would involve looking for components containing `<reaction>` elements and then tracing all reactant and product variables back to where they are declared with a `public_interface` attribute value of "out" to find the original name of each species. SBML's compartment idea could just be ignored (i.e., all species could be placed in a single compartment), or it could be checked whether there were any containment relationships between species components and other components which might then be regarded as compartments. Any mathematics which fell outside `<reaction>` elements, or mathematics which did not fit the recommended best practice shown in the examples in the specification would probably have to be ignored, as it could probably not be represented in SBML without human intervention.

The translation from SBML to CellML appears relatively straightforward, and Warren volunteered to demonstrate that this could be done with an XSLT transformation. XSLT would probably not be a suitable tool for the parsing of Infix mathematics so that would have to be left to an external Java class.

As increased complexity is added to each of the languages, it is anticipated that translation between the two formats would still be possible, because any model could be "flattened" into the basic language syntax (probably at the expense of some verbosity) which could then be translated. Flattening might involve, for instance, creating complete instances of any components that are re-used in a CellML document, or a complete specification of all of the compartments and reactions in an SBML array.

5 Unresolved SBML -> CellML Issues

Discussion of translation from SBML to CellML highlighted some current and future features of SBML that do not have clear mappings to information in CellML. The first of these is the `fast` attribute on SBML's `<reaction>` element. According to Herbert, this is simply a boolean flag that allows processing software to make some assumptions about reaction kinetics that allow some optimisation in integration. An SBML to CellML translator could ignore this flag. However it raises some questions regarding whether CellML should provide a facility for modellers to define a number of sets of equations for the same reaction. The equations used in simulation would depend on the timescale at which the simulation was run: a rate law

¹http://www.cellml.org/examples/examples/signal_transduction_models/basic_reaction_models/two_reaction_model_doc.html

when the reaction is relatively slow, and equilibrium equations when the reaction is relatively fast. The CellML team urgently needs to determine if this functionality is needed, so that they can make any changes necessitated to the reaction specification as soon as possible.

Another issue is the specification of compartments in SBML. Currently, SBML contains no facilities for defining anything more complicated than the equivalent of CellML's containment relationship (i.e., one compartment is contained within another). However, the volume of the compartment is mathematically important (it might be used in rate law calculations for example).

Looming in the "Future Enhancements" section of the SBML specification is a reference to creating component arrays. This would allow the compact specification of large multi-dimensional arrays of compartments, species, and reactions where the reactions might occur across compartment boundaries for example. Such functionality can be safely delayed in CellML too, however it is something that the CellML team needs to be conscious of, and it would be nice to have input into the development of those facilities.

Finally, we have the specification of stochastic mathematics. There are no examples of the use of stochastic mathematics in the SBML specification so it is unclear if SBML can currently handle this. MathML does not have any facilities specifically targeting this area, so CellML could probably not handle stochastic problems without a little bit of effort. This is probably also a problem area which both teams can contribute to.

6 CellML/SBML co-operation

Those present at the meeting proposed formalising the relationship between the SBML and CellML projects to minimise confusion about how the two projects were related, and to let people know that the two projects were proceeding co-operatively. It was proposed that the following text be placed on both websites, along with links to the other project (note that the text has been extended a bit by Warren):

The [CellML](#)² and [SBML](#)³ projects have similar, but not identical, goals. CellML™ and SBML are both XML-based exchange formats for describing cellular models. CellML describes the structure and underlying mathematics of cellular models in a very general way and has facilities for describing any associated metadata. SBML is aimed at exchanging information about pathway and reaction models between several existing applications.

The CellML and SBML development teams are actively discussing how the two languages can be made to work together. It is anticipated that it will be reasonably straightforward to convert between CellML and SBML without losing information about model structure and function.

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

²<http://www.cellml.org/>

³<http://www.cds.caltech.edu/erato/>