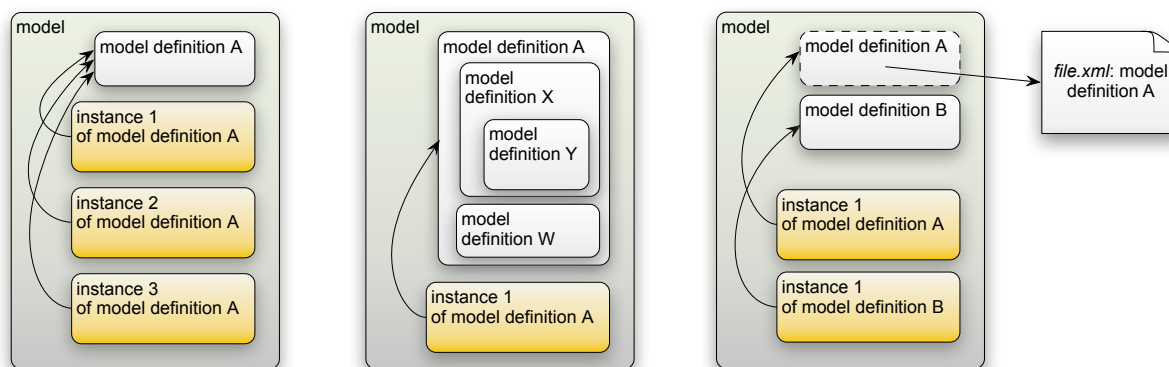


Introduction and motivation

In the context of SBML, *hierarchical model composition* refers to the ability to include models as submodels inside other models. The goal of this facility is to support the ability of modelers and software tools to do such things as (1) decompose larger models into smaller ones, as a way to manage model complexity; (2) incorporate multiple instances of the same component model within one or more enclosing models, to avoid literal duplication of repeated elements; and (3) create libraries of vetted models, much as is done in software development, electronics design, and other engineering fields. This document describes a proposal for an SBML Level 3 package to support this kind of hierarchical model composition.

The figure below illustrates some of the scenarios targetted by this proposal. From left to right, the figure shows a model composed of multiple instances of a single, internally-defined, template submodel definition; a model composed of a submodel that is itself composed of submodels; and a model composed of submodels, some of which are defined internally and some of which are defined in external files.



The effort to define a hierarchical model composition facility for SBML has a long history; we summarize it in the next section. It has also been known by different names. Originally, it was called *modularity*, to reflect the notion of dividing a model into structural and conceptual modules. It was renamed *model composition* when it became apparent that the name “modularity” was too-easily confused with the notion of XHTML 1.1 modularity¹ and decomposition of an SBML file into separate files. To make clear that the purpose is structural *model* decomposition, regardless of whether the components are stored in separate files, the SBML community adopted the name SBML Hierarchical Model Composition in the mid-2000’s.

To support a variety of composition scenarios, this package provides for optional black-box and white-box encapsulation by means of defined data communication interfaces (here called *ports*). In addition, it also provides for a separation of model *definitions* (i.e., blueprints, or templates) from *instances* of those definitions, as well as optional external file storage, and a recursive model decomposition facility that allows arbitrary submodel nesting.

¹ “XHTML 1.1 — Module-based XHTML”, <http://www.w3.org/TR/xhtml11/>, W3C, 31 May 2001.

Background

Problems with current SBML approaches

SBML Level 3 Core has no direct support for allowing a model to include other models as submodels. Software tools either have to implement their own schemes outside of SBML, or (in principle) one could also use annotations to augment a plain SBML Level 2 model with the necessary information to allow a software tool to compose a model out of submodels. However, such solutions would be proprietary and tool-specific, and not conducive to interoperability. There is a clear need for an official SBML language facility for hierarchical model composition.

Past work on this problem or similar topics

The SBML community has discussed the need to add model composition features to SBML since its very beginning, some ten years ago. In an internal discussion document titled “Possible extensions to the Systems Biology Markup Language”² principally authored by Andrew Finney (and, notably, written even before SBML Level 1 Version 1 was finalized in March of 2001), the first of the four titular possible extensions is for “submodels”. In that document, the main model object can contain a list of submodels, each of which are model definitions only, and a list of submodel instantiations, each of which are references to elements in the submodel list. Finney’s proposal also extends the syntax of SBML identifiers (the `Sid` data type) to allow entity references using a dotted notation, in which `X.y` signifies element `y` of submodel instance `X`; the proposal also defines a form of linking model elements through “substitutions”. The proposal also introduced the concept of validation through what it called the “expanded” version of the model (now commonly referred to as the “flattened” form, meaning translation to a plain SBML format that does not use composition features): if the flat version of the model is valid, then the model as a whole must also be valid.

In June of 2001, at the Third Workshop on Software Platforms for Systems Biology, Martin Ginkel and Jörg Stelling presented their proposal titled “XML Notation for Modularity”³, complete with an accompanying proposal document and sample XML file, partially in response to deficiencies or missing elements they believed existed in the proposal by Finney. In their proposal, Ginkel and Stelling present a “classic view” of modularity, where models are packaged as black boxes with interfaces. One of their design goals is to support the substitution of one module for another with the same `defineInterface`, thereby supporting the simplification or elaboration of models as needed. Their proposal emphasizes the reuse of models and with the possibility of developing libraries of models.

Martin Ginkel again presented an expanded version of that proposal⁴ in the July 2002 *Fifth Workshop on Software Platforms for Systems Biology*, in the hope that it could be incorporated into the definition of SBML Level 2 that was being developed at the time. This proposal clarified

² <http://sbml.svn.sourceforge.net/viewvc/sbml/trunk/specifications/sbml-level-3/old/sbml-team-proposals/original/sbmlex.txt>

³ Presentation at <http://sbml.org/images/7/73/Vortrag.pdf>, pre-meeting proposal at <http://sbml.org/images/3/3d/Joerg-sbml-proposals.pdf>, and sample file at <http://sbml.org/images/1/18/Sbml2.txt>

⁴ <http://sbml.org/images/9/90/Sbml-modular.pdf>

the need to separate model definitions from model instantiations, and, further, the need to designate one model per document as the “main” model.

In March of 2003, an independent proposal⁵ by Jonathan Webb was posted to the sbml-discuss mailing list. The proposal itself has not survived intact on the web (the host site disallowed indexing of their site, preventing even Archive.org from saving it), but from the follow-up discussions between him, Andrew Finney, and Martin Ginkel, it seems that this proposal included a unified, generic approach to making links and references to elements in submodels using XPath. Previous proposals used separate mechanisms for species, parameters, compartments, and reactions. Webb also brought up the issue of how to successfully resolve conflicting attributes of linked elements, debated whether formal interfaces were necessary or even preferable to directly access model elements, discussed type checking for linkages, and discussed issues with unit incompatibilities. Around this time, Martin Ginkel formed the Model Composition Special Interest Group⁶, a group that eventually reached 18 members, including Jonathan Webb.

Model composition did not make it into SBML Level 2 when that specification was released in June of 2003, because the changes between SBML Level 1 and Level 2 were already substantial enough that software developers at the time expressed a desire to delay the introduction of composition to a later revision of SBML. Andrew Finney (now the co-chair of the Model Composition SIG) presented yet another proposal⁷ in May of 2003, even before SBML Level 2 Version 1 was finalized, that aimed to add model composition to SBML Level 3. With only two years having passed between SBML Level 1 and Level 2, the feeling at the time was that Level 3 was likely to be released in 2005 or 2006, and the model composition proposal would be ready when it was. However, the SBML community was occupied far longer than expected with the implementation and exploration of SBML Level 2, and deliberately delayed the development of SBML Level 3.

In the interim, the effort to develop model composition features for SBML continued. Finney revised his 2003 proposal in October 2003⁸; this new version represented an attempt to synthesize the earlier proposals by Ginkel and Webb, supplemented with his own original submodel ideas, and was envisioned to exist in parallel with another proposal by Finney, for arrays and sets of SBML elements (including submodels)⁹. Finney attempted to resolve the differences in the two basic philosophies (essentially, black-box versus white-box encapsulation) by introducing optional “ports” as interfaces between a submodel and its containing model, as well as including an XPath-based method to allow referencing model entities. The intention was that a modeler who wanted to follow the classic modularity (black-box) approach could do so, but other modelers could still use models in ways not envisioned by the original modeler simply by accessing a model’s elements directly via XPath-based references. In both schemes, elements in the submodels were replaced by corresponding elements of the containing model.

⁵ http://sbml.org/Forums/index.php?t=msg&th=67&rid=0#msg_111

⁶ <http://www.mpi-magdeburg.mpg.de/zlocal/martins/sbml-comp>

⁷ <http://www.mpi-magdeburg.mpg.de/zlocal/martins/sbml-comp/model-composition.pdf>

⁸ <http://sbml.org/images/7/73/Model-composition.pdf>

⁹ Posted to sbml-discuss, http://sbml.org/Forums/index.php?t=msg&th=234&rid=0#msg_683

Finney's proposal also provided a direct link facility that allows a containing model to refer directly to submodel elements without providing placeholder elements in the containing model. For example, a containing model could have a reaction that converted a species in one submodel to a species in a different submodel, and in the direct-link approach, it would only need to define the reaction, with the reactant and product being expressed as links directly to the species defined in the submodels.

After Finney's last effort, activities in the SBML community focused on updates to SBML Level 2, and since model composition was slated for Level 3, not much progress was made for several years, apart from Finney including a summary of his 2003 proposal and of some of the unresolved issues in a poster¹⁰ at the 2004 Intelligent Systems for Molecular Biology (ISMB) conference held in Glasgow.

A candidate Level 3 Version 1 Core specification was not released until the end of 2009, and it is only today (near the end of 2010) that the release of a final Level 3 Version 1 Core specification is imminent. As a consequence of the lack of a concrete, finalized SBML Level 3 Core specification, all of the model composition efforts up to this point have been theoretical: they could not define a precise syntax as long as the underlying SBML Level 3 syntax was not finalized. Software tools that did implement some form of composition had to do so using proprietary approaches or extensions to SBML. This has now changed, thanks to the finalization of SBML Level 3 Version 1 Core, and the present proposal is an attempt to blend features of previous efforts into a concrete, Level 3-compatible syntax.

10 <http://sbml.org/images/9/9c/Ismb-2004-sbml-level-3-poster.pdf>