SBML Level 3
Hierarchical Model Composition

Lucian Smith
COMBINE 2010
3 Submodels

Andrew Finney
afinney@cds.caltech.edu
ERATO Kitano Systems Biology Workbench Development Group
Control and Dynamical Systems 107-81
California Institute of Technology, Pasadena, CA 91125

Version of November 27, 2000
Modular Modeling of cellular systems

Martin Ginkel, Jörg Stelling
Max-Planck-Institute for Dynamics of complex technical Systems
Magdeburg, Germany

1st June 2001
Modular SBML

Proposal for an Extension of SBML towards level 2

Martin Ginkel
Max-Planck-Institute Dynamics of complex technical Systems, Magdeburg, Germany

ginkel@mpi-magdeburg.mpg.de

10th June 2002
BioSpice MDL Model Composition and Libraries

This document describes a set of capabilities supporting the construction of models from components and the collection of those components in libraries. Since BioSpice is currently using SBML as the MDL, this presentation is slanted toward describing extensions to SBML.

Please note this is work in progress. A number of sections are incomplete and we are looking for feedback on the general approach being proposed.

Model Composition

Models of realistic systems will be large. We presume there will be recurrent structures within models of a given system and between models of different and possibly related systems.
Systems Biology Markup Language (SBML) Level 3 Proposal: Model Composition Features

Andrew Finney
afinney@cds.caltech.edu

October 31, 2003
The 5th SBML Hackathon

SBML Composition Workshop 2007

University of Connecticut Health Center, Farmington, CT, USA, 9-10 September 2007

Model composition refers to the ability to include models as submodels inside other models. This requires defining the interfaces between the models and protocols for connecting parts of models together. One of the anticipated extensions for SBML Level 3 is to support model composition.
The 5th SBML Workshop - Martin Goals

SBML

1. General Goals
   - (UseCase? 1): Composing tool takes multiple traditional SBML templates, and composes them into a single model by blessed changes to the original models.
   - (UseCase? 2): Modular modeling tool creates models from preexisting templates to be composed in the sense of software engineering.
   - (UseCase? 3): Analysis tools read and analyze (simulate, compare, etc.) the composite models.
   - Important: Balance between complexity of storage format and convenience of implementation.
   - General question: Is it useful/necessary to make composition of models into larger models possible?

2. Goals for Composition Extension
   - Express multiple submodels or modules
   - Express instantiation
   - Create directed links

0 September 2007
Object Oriented Modeling in Promot

Motivation

- Structured Models are very common in other simulation descriptions
- Examples: Simulink, Dymola, Modelica, gProms, CellML
- Technical simulations naturally reflect the modular structure of the modeled systems (and there are still doubts about modules in biology)
- Learn from the experiences of other simulation fields.
- Promot was originally developed for process engineering
- It uses object-oriented models, composed from modules

Introductory Example

- EGF Model based on [Schoeberl 2002] implemented in Promot
- Model is constructed with modular structure, submodels are designed to be composed
- Modules for the major specific functional networks of the system
- Modules interact with their siblings by terminals (ports, interfaces)
- Modules are independently developed, tested and analyzed

Express instantiation

Create directed links
Andrew 2007 Comments about Model Composition

Folks

The following issues were raised which were not covered by the existing model composition (despite the tone of the language I'm actually writing this for comments! - that includes if you have already thought these issues though ages ago

\begin{itemize}
\item[a)] Explicit 'MustBeOverloaded' flag on ports
\begin{itemize}
\item Currently ports (elements that are explicitly part of a model's interface) by default call when the containing model is instantiated. Ports must have an additional flag 'MustBeOverloaded' that when true indicates that the port must be linked to. It is recommended that simply containing, at the top level, ports with 'MustBeOverloaded' true values is simulated.
\end{itemize}
\item[b)] N to N links
\begin{itemize}
\item (on links a 'from' reference overloads a 'to' reference) The syntax for links should be and one or more 'to' object references. A link with zero 'from' references deletes the more than one species or compartment 'from' references (otherwise you could split a link to nothing is not allowed)
\end{itemize}
\item[c)] Reaction overloading
\begin{itemize}
\item Reaction links should be allowed. Reaction overloading differs from species overloading the reaction network bipartite graph the flattened model containing a link from one species combines the edges from both models incident to both species e.g. model1: s1 -> s2
\end{itemize}
\end{itemize}
Andrew 2007 Comments about Model Composition

Folks

The
(des if yo

a) E

Curr when that cont

b) N

(on and more

link is nothing is not aligned)

c) Reaction overloading

Reaction links should be allowed. Reaction overloading differs from species overloads the reaction network bipartite graph the flattened model containing a link from one species to another combines the edges from both models incident to both species e.g. model1: s1 -> s2

Wolfram Liebermeister, MPI-MG Berlin
Computational Systems Biology

SBML composition meeting
UCHC, September 9-10 2007
Andrew 2007 Comments about Model Composition

Model merging with semantic SBML

Model Composition for Macromolecular Regulatory Networks

Ranjit Randhawa, Clifford A. Shaffer, Senior Member, IEEE, and John J. Tyson
Issues To Address

The following are unresolved issues and questions that arose from the videoconference discussions at UCHC during the SBML Composition workshop. The comments below conflict with some of the comments below, in which there are questions below as they are, in order to describe the space of possibilities, but the model down to L2v3 is a desirable feature and should serve as a design interpretation immediately and transparently by software tools that use adoption of the composition facility.

AF: This approach also simplifies the definition of some of the semantic model (in L3) is valid if the flattened L2 model is valid. This is not to this way :).

[edit] Terminology

SBML Inclusion: Separating pieces of SBML into separate files and having a mechanism to include the external pieces into a given file.

SBML Model Composition: composing a model from submodels. AF suggests the following subdivisions within composition:
Hierarchical Model Composition (Hoops 2007)

[edit] Proposal title

Hierarchical Model Composition

[edit] Proposal authors

Stefan Hoops
Virginia Bioinformatics Institute
Blacksburg, VA 24061, USA
Email: shoops at vbi.vt.edu

Nicolas Le Novère
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge CB10 1SD, UK
Email: lenov at ebi.ac.uk

Ion Moraru
Center for Cell Analysis and Modeling
University of Connecticut Health Center
Hierarchical Modeling

Ion Moraru
Stefan Hoops

SBML Workshop
Gotheburg
Sweden
Hierarchical Modeling

Ion Moraru
Stefan Hoops

SBML Workshop
Gothenburg
Sweden
Hierarchical Model Composition

'Replace'

X → B

A → B

B → C
1. Separate the concepts of 'replacement' and 'deletion' into parallel elements: ListOfReplacements/Replacement, and ListOfDeletions/Deletion

2. Annotate the Deletions with a list of conceptual replacements for those elements.

3. Let ConversionFactors be non-const; possibly make them required instead of optional.

4. Loosen reliance on xpointers by allowing ObjectRefs to reference by metaID and/or by increasing the number of elements with SIDs.

5. Separate model definitions from model instantiations.

6. Turn 'Ports' into annotations. (this one is not my proposal, but a proposal 'from the floor', as it were).

7. The spec should be written in such a way as to explicitly accommodate packages in general, without needing to refer to particular packages.

8. Function Definitions must only apply to the model they are a member of. Do we want a more general list as well?
Design Goals

- Aggregation, Composition, or Black Box
Design Goals

- Interact cleanly with other packages
Design Goals

- Core still valid without comp constructs
Design Goals

- Ignore verbosity of model, but don’t over-complicate
Design Goals

- Allow modular access by reference
Incorporate past design goals
Basic structure: Definitions and Submodels

SBML (extended)

listOfModelDefinitions

0,1

ListOfModelDefinitions

modelDefinition

0,...,*

Model (extended)

model

Model (extended)
Basic structure: Definitions and Submodels

Model *(extended)*

`listOfSubmodels`

0, 1

ListOfSubmodels

`submodel`

0, ..., *

Submodel

SBase
Basic structure: Definitions and Submodels

sbml l3 core

model
Basic structure: Definitions and Submodels

sbml comp

model

submodel

submodel

submodel

modelDefinition

modelDefinition

submodel

submodel

modelDefinition

modelDefinition

submodel
Basic structure: Definitions and Submodels

sbml comp

model

submodel

submodel

submodel

modelDefinition

submodel

modelDefinition

submodel

modelDefinition

submodel

modelDefinition

submodel

comp

l3core
Basic structure: Definitions and Submodels

Submodel

- id: SId [use="optional"]
- lengthConversionFactor: SIdRef [use="optional"]
- volumeConversionFactor: SIdRef [use="optional"]
- areaConversionFactor: SIdRef [use="optional"]
- substanceConversionFactor: SIdRef [use="optional"]
- timeConversionFactor: SIdRef [use="optional"]
- extentConversionFactor: SIdRef [use="optional"]

ModelRef

- model: SIdRef [use="optional"]
- xlink:href: string [use="optional"]
- md5: string [use="optional"]

ListOfDeletions

- deletion: Subelement
  - id: SId [use="optional"]
Subelements: ways to refer to submodel elements

- submodel: SIdRef
- port: PIdRef {use="optional"}
- idRef: SIdRef {use="optional"}
- metaidRef: metaidRef {use="optional"}
- xpath:element: string {use="optional"}
- deletion: SIdRef {use="optional"}

Subelement

0,1

Subelement
Subelements: ways to refer to objects

sbml comp

model

subElement

submodel

submodel

modelDefinition

port

Required

Subelements: ways to refer to objects

sbml comp

- model
  - subElement
    - submodel
    - submodel
  - modelDefinition
    - SID
    - port

Required
Subelements: ways to refer to objects
Subelements: ways to refer to objects

sbml comp

model

subElement

submodel

modelDefinition

SId

port

metaID

Required

xpath
Subelements: ways to refer to objects

sbml comp

model

subElement

subElement

submodel

submodel

modelDefinition

SId

port

metaID

submodel

...
Ports and Replacements

```
SBase (extended)

portid: PId {use="optional"}

replaces
0,1

ListOfReplacements

subelement
0,...,*

ReplacedElement

identical: boolean {use="optional"}
conversionFactor: SIdRef {use="optional"}
```
Ports and Replacements

sbml comp

model

subElement

submodel

modelDefinition
Ports and Replacements
Ports and Replacements

sbml comp

model

Sbase “replacement”

replaces

subElement

subElement

subElement

modelDefinition

portId

modelDefinition

SId

modelDefinition

metald
Any old reference to a replaced element’s SId now refers to the new element.

Any old reference to a replaced element’s metaId now refers to the new element.

All old math must use conversionFactors from Replacement elements, if defined, or the Submodel’s conversionFactors, if defined and relevant.

If flagged ‘identical=true’, the replacing element must be identical to what it replaces.
Deletions revisited

Submodel
- id: Sld {use="optional"}
- lengthConversionFactor: SldRef {use="optional"}
- volumeConversionFactor: SldRef {use="optional"}
- areaConversionFactor: SldRef {use="optional"}
- substanceConversionFactor: SldRef {use="optional"}
- timeConversionFactor: SldRef {use="optional"}
- extentConversionFactor: SldRef {use="optional"}

modelRef

ModelRef
- model: SldRef {use="optional"}
- xlink:href: string {use="optional"}
- md5: string {use="optional"}

listOfDeletions
- 0,1

ListOfDeletions
- deletion
- 0,1,*

Deletion
- id: Sld {use="optional"}
Deletions Revisited

sbml comp

model

submodel

deletion

deletion

deletion

modelDefinition

---

Validation

- Determine if the deletion/replacement rules can be followed.
- Determine if following the rules results in a valid ‘flat’ model.
- That’s it! (no type checking!)
If a conversionFactor is defined for a replacement, that takes precedence.
If not, relevant conversionFactors are applied to all remaining subelements in the submodel.
If an element has a unit type, it can be converted by the corresponding conversionFactor.

Example: A compartment with spatialDimension=3: areaConversionFactor

Example 2: A species with 'hasOnlySubstanceUnits=true' in a 3D comp: substanceConversionFactor/areaConversionFactor

Species that replace species with different values for 'hasOnlySubstanceUnits' use the value of compartment in their conversion.
Acknowledgements

- Mike Hucka
- Herbert Sauro
- All the previous authors and contributors to previous Hierarchical Model Composition proposals, especially Stefan Hoops, Andrew Finney, Martin Ginkel, and Jonathan Webb!