Composition and Aggregation for Biological Pathway Modeling

Ranjit Randhawa*, Cliff A. Shaffer* and John J. Tyson**

Synopsis

Today's pathway models are small compared to the amount of information known about a particular cellular pathway, in part because current modeling languages and tools are unable to handle significantly larger models. Thus, most pathway modeling work today focuses on building small models of individual pathways since they are easy to construct and manage. The hope is someday to put these pieces together to create a more complete picture of the underlying molecular machinery. While efforts to make large models benefit from reusing existing components, unfortunately, there currently exists little tool or representational support for combining or composing models. We have identified four distinct modeling processes related to model composition: fusion, composition, aggregation, and flattening. We present concrete proposals for implementing all four processes in the context of the Systems Biology Markup Language (SBML).

Fusion

Fusion is the process of combining two of more models into a single "flat" model to create larger models. We are prototyping a tool that assists modelers incorporating information from another model into an existing one. However, the resulting monolithic model is inherently as complex as the sum of the complexities of the submodels, and there is a limit to how large such a model can become and still be comprehensible.

Naming conflicts between submodels are resolved using mapping tables, which are produced for each of the eight SBML component types, in the following order: (1) Compartments, (2) Species; (3) Function Definitions; (4) Rules; (5) Events; (6) Unit; (7) Reactions; and (8) Parameters. The two options available to the modeler while resolving naming conflicts are:
1. Define two or more SBML components to be equivalent.
2. Remove the link/association between two or more SBML components (which have previously been incorrectly linked together) across the different submodels.
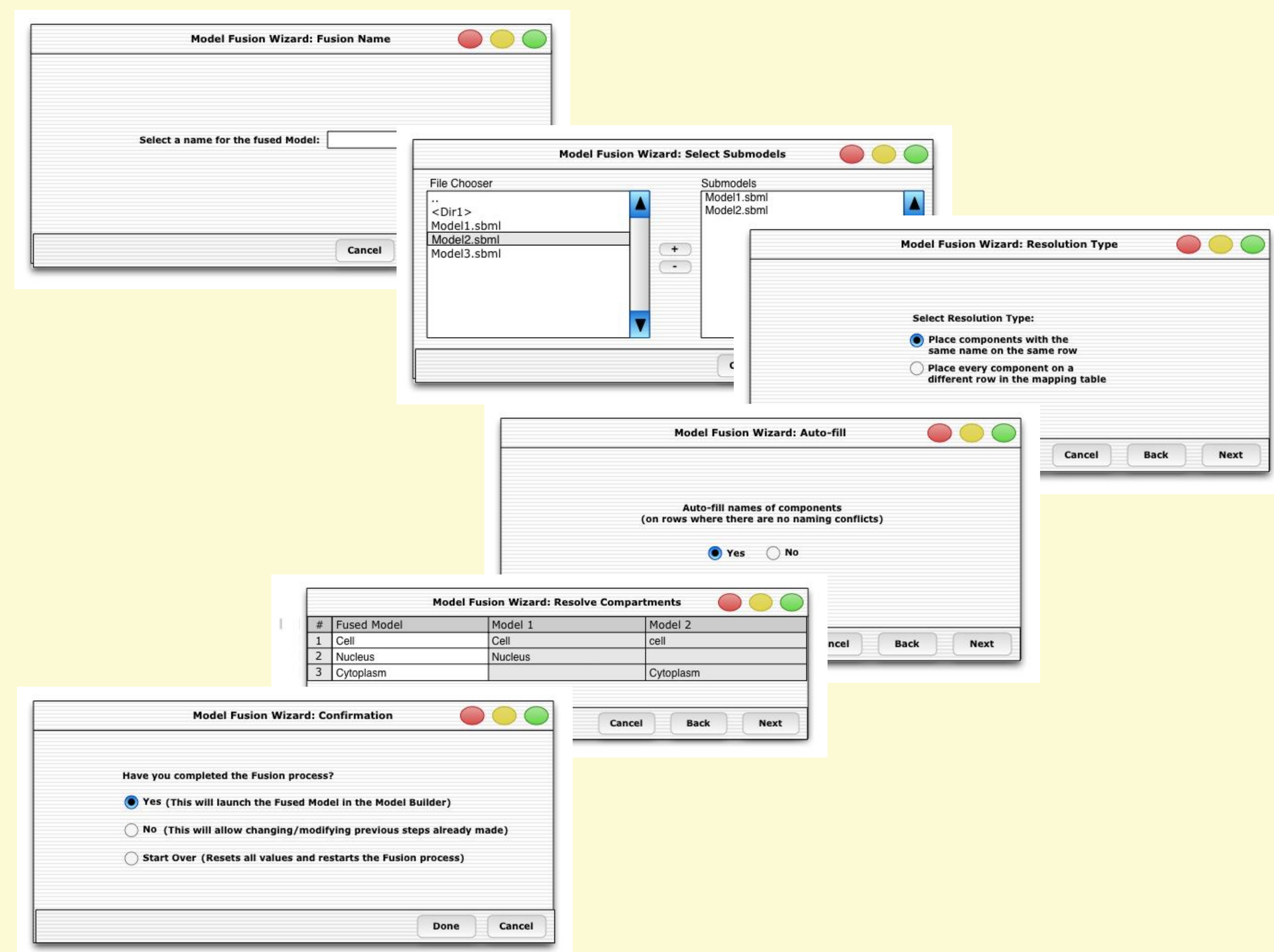

Figure 2: Prototype of the Model Fusion Wizard

Composition

Composition allows modelers to combine a collection of submodels. Submodels are themselves complete models, not elements within a model such as species, parameters, etc. We have defined a small set of features to be added to the SBML language [2] to support hierarchical composition.
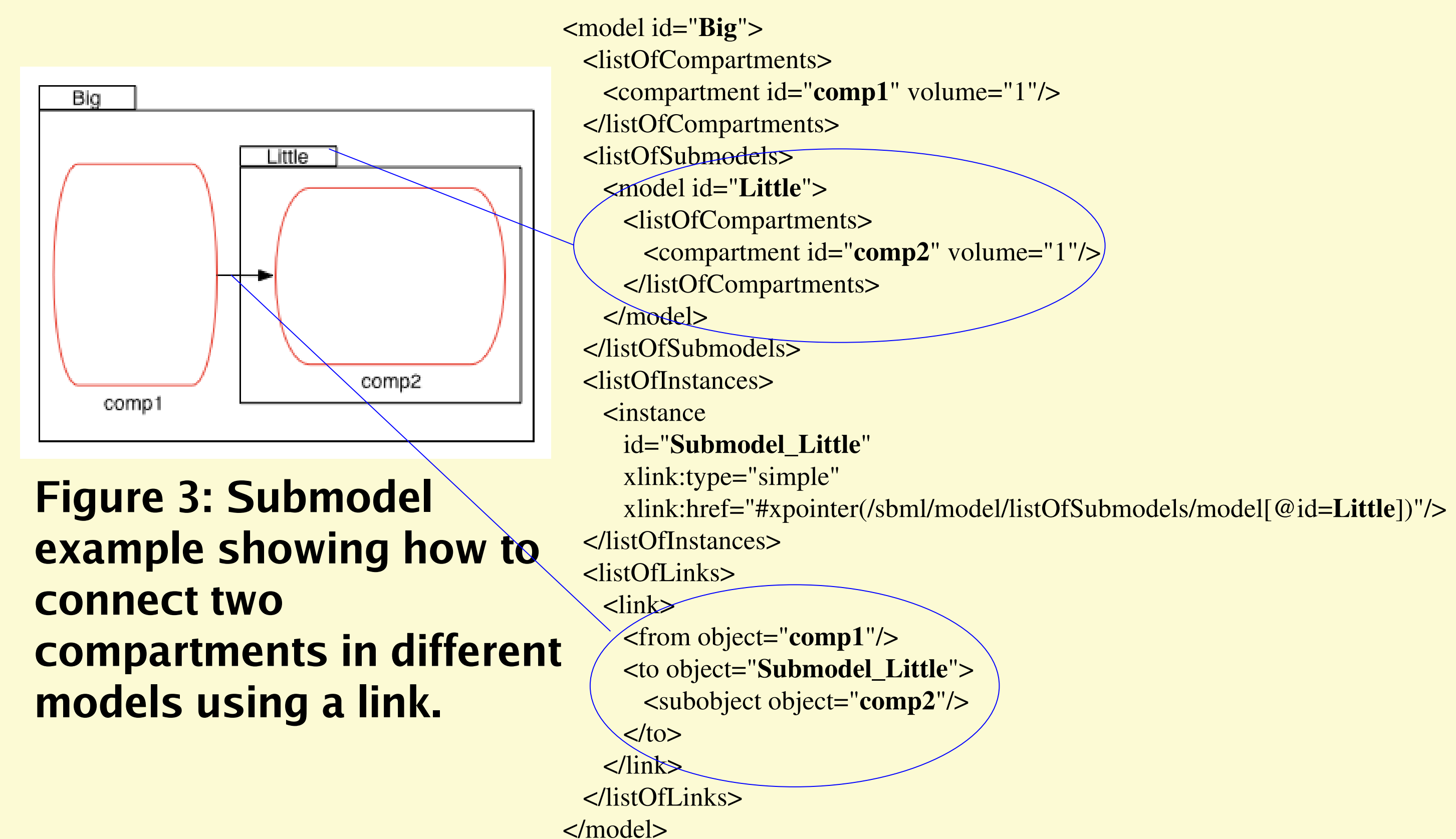

Figure 3: Submodel example showing how to connect two compartments in different models using a link.

SBML features:
<listOfSubmodels>: Contains a list of **<submodel>** structures which are valid SBML models.
<listOfInstances>: Contains a list of **<instance>** structures which refer to a particular <model>. An <instance> indicates that a copy of a submodel is being instantiated within the current model.
<listOfLinks>: Contains a list of **<link>** structures which link two entities in separate (sub)models of a composed model.

Aggregation

Aggregation is a variation on composition that allows users to provide interfaces to components so as to restrict access to the information of a submodel by exposing only certain variables. The fundamental difference between aggregation and composition is the amount of access to model information. An aggregated model contains a list of input and output ports (interfaces) that link to internal species and parameters. Submodels can then be connected via their interfaces to create larger models, also with restricted interfaces. Like composition, we implement aggregation through added SBML language constructs.

The **<port>** structure (enclosed in a **<listOfPorts>** structure) allows a modeler access to a particular species or parameter in a submodel. Input ports can access one or more of the parameters that appear within the submodel. The output ports can access one or more of the species that appear within the submodel. Once ports are selected a modeler can link submodels together (using **<link>** structures) to create more complex models.
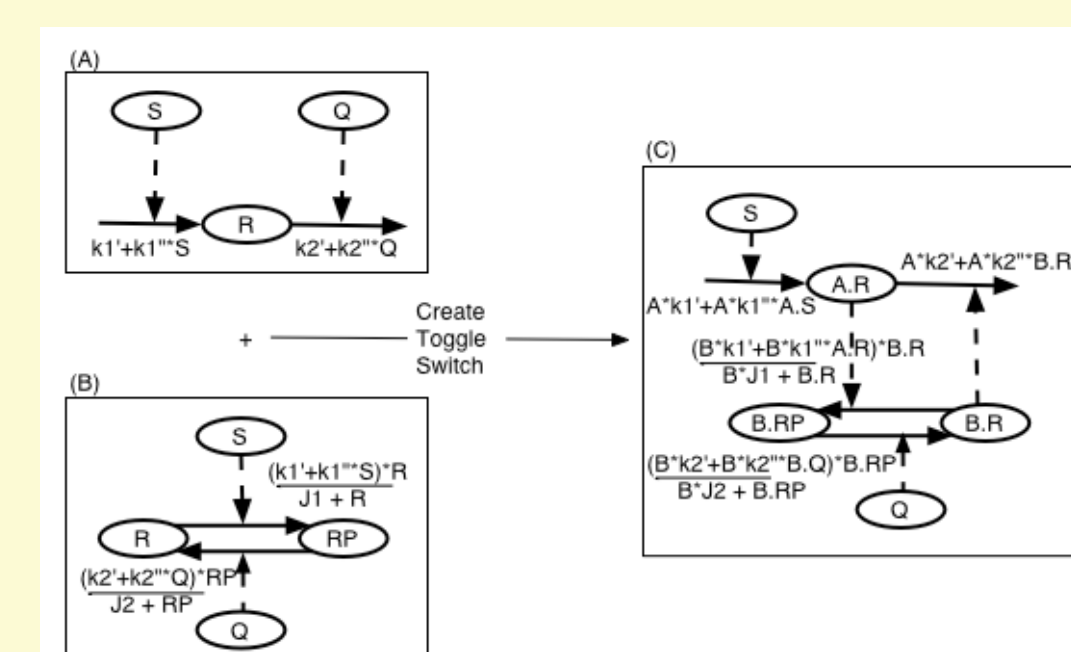

Figure 4: A toggle switch model (C) [3], a two-way discontinuous switch can be created by linking together two simpler models, a linear response element (A) and a hyperbolic response element (B).
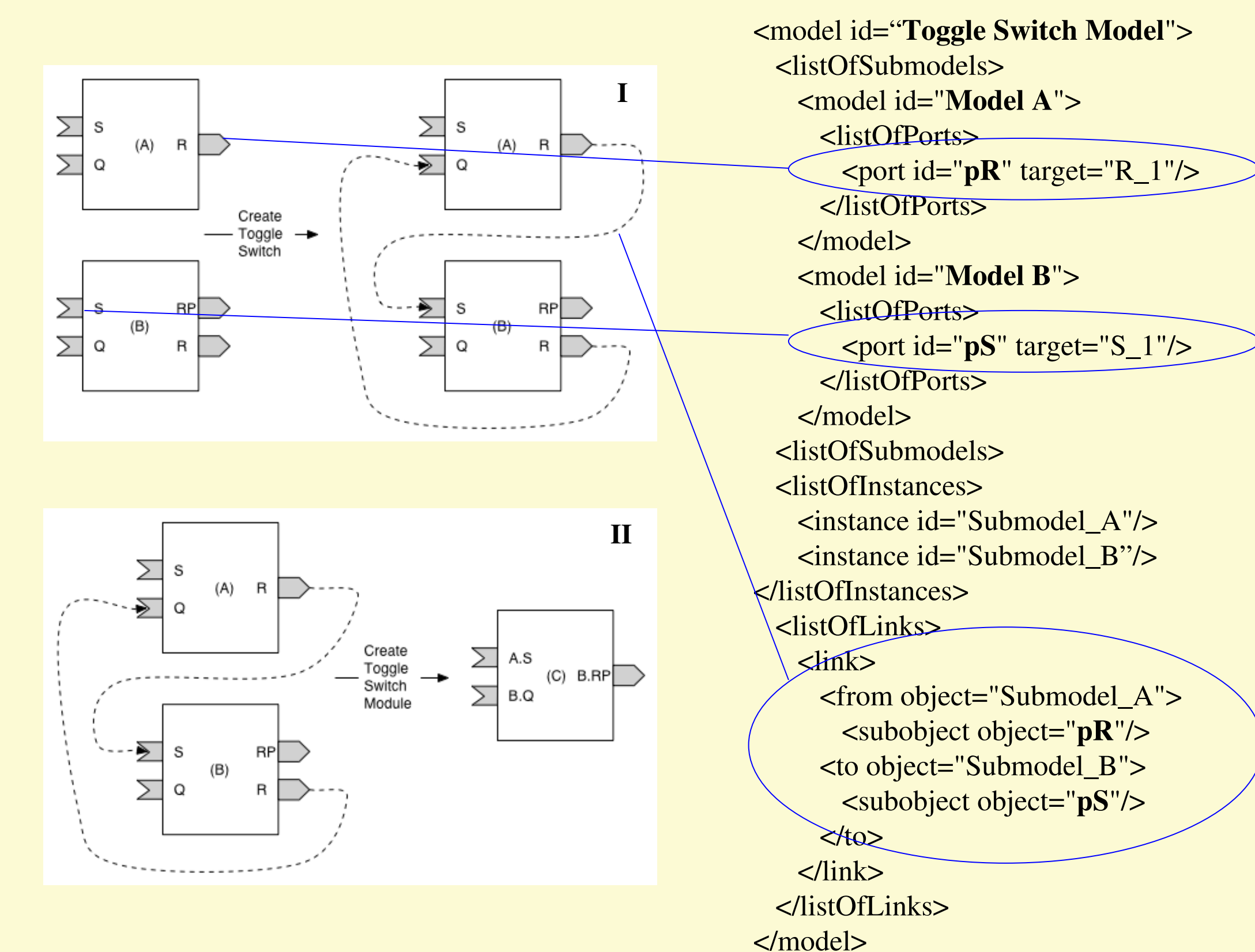

Figure 5: (I) Iconified toggle switch with input/output ports. S and Q are signals, R and RP are responses. SBML code of the aggregated model is also shown with relevant sections highlighted. (II) Iconified toggle switch can now be made into a new model with its own set of inputs and outputs by defining its set of port structures.

Flattening

Flattening converts a composed or aggregated model with its hierarchy of connections to one without any hierarchy or connections. In SBML terms, flattening a composed or aggregated model will result in a valid SBML Level 2 [1] model. Flattening automatically converts composed or aggregated models to fused models, so they can use existing simulators that lack support for composition or aggregation.

Flattening is a three step process:
1. Separation: Submodels are first separated from each other based on the information within the <listOfSubmodels> and <listOfInstances> structures. Then components are separated one at a time in the same order as described in Fusion.
2. Saving: The connections/links between models is saved for reference during the Resolution step.
3. Resolution: The components of the submodels are sorted/separated/assigned based on the model (or submodel) they originated from. The resolution step is similar to the resolution step in fusion, except here it is done automatically.

References

[1] A Finney, M Hucka, and H Bolouri. Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Available at http://sbml.org/specifications/sbml-level-2/version-1/html/ sbml-level-2.html, 2003.

[2] M Hucka, A Finney, H M Sauro, H Bolouri, J C Doyle, H Kitano, A P Arkin, B J Bornstein, D Bray, A Cornish-Bowden, A A Cuellar, S Dronov, E D Gilles, M Ginkel, V Gor, I I Goryanin, W J Hedley, T C Hodgman, J-H Hofmeyr, P J Hunter, N S Juty, J L Kasberger, A Kremling, U Kummer, N Le Novere, L M Loew, D Lucio, P Mendes, E Minch, E D Mjolsness, Y Nakayama, M R Nelson, P F Nielsen, T Sakurada, J C Schaff, B E Shapiro, T S Shimizu, H D Spence, J Stelling, K Takahashi, M Tomita, J Wagner, and J Wang. The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. Bioinformatics, 19(4):524–531, 2003.

[3] J J Tyson and B Novak. Regulation of the Eukaryotic Cell Cycle: Molecular Antagonism, Hysteresis, and Irreversible Transitions. Journal of Theoretical Biology, 210:249–263, 2001.

For More Information:

http://jigcell.biol.vt.edu/

* Department of Computer Science, 660 McBryde Hall (0106) and
** Department of Biological Sciences, 2119 Derring Hall (0406)
Virginia Tech, Blacksburg, VA 24061