

SBML Level 2: Experiences and Consequences for a Modeling Environment

In our use (Virginia Tech's) of SBML Level 2, we have found that the evolutionary nature of the specification can cause many problems for a development group.

The first is that of maintaining a version to version compatibility, so that no one loses their model because the specification was changed. Also, one must fight within those version changes to ensure their group does not lose any of its functionality, which would cause the group to use the annotation layer to regain the functionality.

Second, the annotation layer provides for an expression of that which is not in the language, such as conservation relations. However, one can project that over time, so many annotations may creep into existence that maintenance from a coding standpoint may become an issue. A certain threshold needs to be established so that these annotations are not duplicated among the various groups. If they are duplicated, then a new feature should be introduced into the specification.

Next, annotations and notes must be preserved between modeling environments and exterminated when they become no longer strongly valid. Such a definition would be that if $A \rightarrow B$ becomes $C \rightarrow B$, all annotations, RDF's, and notes, associated with the reaction are destroyed. If this is not done, the model may become incoherent to the biologist. This becomes even more important when dealing with annotations for a wiring diagram such as with BSP.

Finally, the use of MathML 2.0 in SBML Level 2 needs to be examined. It is not being used to its full power, as only a small subset is used, and it is also being abused syntactically in the area of functions (lambdas) under recent proposals. Furthermore, it is not readable by anyone (except for the few who understand the notational style) in its natural state, as opposed to the mathematical expressions of SBML Level 1.