**SBML: A Tool-Neutral Model Definition Format**

SBML — the Systems Biology Markup Language — is a computer-readable format for representing models of biochemical reaction networks, applicable to:

- **metabolic networks**, including pathways,
- **gene networks**, regulatory networks, and
gene-modeling problems in systems biology.

SBML is based on XML, a standard medium for representing and transporting data that is widely supported and recognized as an international standard in both computational biology and bioinformatics.

Because SBML is tool-independent, it enables:

- A **repository of simulation and analysis tools** in a single project without rewriting models for each tool.
- **Publication of models** in peer-reviewed journals — other researchers will now be able to download and use your model even if they use a different software environment.

SBML is well on its way to becoming a lingua franca of computational systems biology. SBML is currently supported by over 60 tools including the following:

- **BioMol**
- **BioModelsDB**
- **Cytoscape**
- **GraPharm**
- **J-Designer**
- **KBase**
- **NetOBIE-SB**
- **Sige**
- **SysteMouse**
- **ToBi**
- **VeoSim2**

**MathSBML: A Package for Manipulating SBML Files**

MathSBML is an open-source Mathematica package that facilitates working with SBML models. Features include:

- Importing SBML models to Mathematica
- Simulation and plotting of SBML models
- Simulation of differential-algebraic systems
- Export of models to SBML-HTML, Berkeley Madonna, XPP/Orion formats
- Complete API for creating and modifying models
- Tabular model display and export to HTML
- Ability to use any Mathematica capability

**Model Interoperability**

With MathSBML, investigators can explore SBML models with a consistent set of Mathematica features. Mathematics is one of several platforms widely used by computational biologists and is valued in many academic and commercial environments (e.g., over 300 US colleges and universities have licensed Mathematica).

MathSBML provides a consistent model interchange with this environment as well as a candidate reference implementation of SBML. Support of other general-purpose languages, including C/C++, Java, Python, Perl, and Maple, has also been developed separately using the Wolfram Language (see references).

MathSBML supports SBML Level 1, Version 1 and 2, and SBML Level 2. The MathSBML Model Editor supports SBML Level 2.

MathSBML will run on any platform that supports Mathematica 4.1 or higher. The solution of differential-algebraic systems via the built-in numerical algebraic solver requires Mathematica 5 or higher; purely differential systems SBML without algebraic relations can be solved on Mathematica 4.1.

### The Model Builder: An SBML API

MathSBML contains a simple model editor, allowing users to create SBML models compatible with other simulators, as well as a Mathematica test-command based API that can be used to produce (usually very simple) SBML files. The model editor contains a suite of commands to add, modify, or remove simple SBML objects such as a reaction, chemical species, or equation from the current model. The API was created in order to facilitate de novo or read from a file. After building the model, the user can test it by running simulators, continue to modify it, or write the results as an SBML file in any order.

### An Example

The following code illustrates a growing cell with a mitotic oscillator [Stelling, B. 2003] in which cell division is initiated when a variable (Mass) that divides in half when sum of the following tests are true at the same time: H1 and H2 and Mass<1. The API commands to build and save the model as a file are shown in the box to the right. To read this file into the simulator, one could enter:

```math
simname = "example.xml1";
```

A file on data can be translated to html via:

```math
SBMLLink["example.xml1", outputfile -> "html.html"]
```

The output is illustrated in the large figure below. Other output formats include: XPP, Berkeley Madonna and Fortran.

When building a model, the user can save it to a file, or test it with the simulator, at any time. For example:

```math
<<mathsbml.m;
```

returns a data structure (called s, in this case) that is compatible with the simulator. Experienced Mathematica users can manipulate the data structure directly.

To run a simulation for sim<200, times units, and plot all model variables, one could enter:

```math
simname = "example.xml1";
```

The resulting screenshot plot is shown below.

### API Commands to Build the Model

<table>
<thead>
<tr>
<th>Set Main Model Parameters</th>
</tr>
</thead>
</table>
| `addSpecies[]` | Create a new species in the model.
| `addParameter[]` | Create a new parameter in the model.
| `addRule[]` | Add a rule to the model.
| `addReaction[]` | Add a reaction to the model.
| `addFunction[]` | Create a new function in the model.
| `addCompartment[]` | Create a new compartment in the model.

### Availability and Documentation

MathSBML is available free of charge from SourceForge under an LGPL license. It is open source and may be downloaded with a single click from our download site:

```
http://sourceforge.net/projects/sbml
```

For example, the site map contains links to documentation on all API function points, as illustrated in the figure below.

### Acknowledgements

MathSBML was developed at the California Institute of Technology. Support has been provided by ERATO Kitano Project, the US National Institutes of Health, and the US National Science Foundation.

### References