MathSBML and Systems Biology Simulations

Bruce E. Shapiro[1], Michael Hucka[2], and Andrew Finney[3]

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA; [2]Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, USA; [3]Science and Technology Research Center, University of Hertfordshire, Hatfield, UK

http://www.sbml.org

Systems Biology Markup Language (SBML): A Tool-Neutral Model Definition Format

SBML is a computer-readable format for representing models of biochemical reaction networks, applicable to:

- metabolic networks,
- cell-signaling pathways,
- genomic regulatory networks, and
- other modeling problems in systems biology.

SBML is based on XML, a standard medium for representing and transporting data that is widely supported on the Internet as well as in computational biology and bioinformatics.

Because SBML is completely tool-independent, it enables

- use of multiple simulation and analysis tools in a single research project without rewriting models for each tool
- publication of models in peer-reviewed journals—other researchers can download and use your model even if they use a different modeling environment
- survival of models—they can outlive the software used to create them, making your work still useful even if a particular simulation package is no longer supported

SBML is currently supported by nearly 40 tools.

BASIS
BioCharon
Bio Sketch Pad
BioSpreadsheet
BioUML
BSTLab
CADLIVE
CellDesigner
Cellerator
Cellware
Cytoscape
DBsolve
Dizzy
E-CELL
ESS
Gepasi
Jarnac
JDesigner
JigCell
JSIM
JWS
Karyote
libSBML
MathSBML
MOMA
Monod
NetBuilder
PathArt
PathScout
ProcessDB
SBW
SCIpath
SigPath
STOCKS
StochSim
TeraSim
Trelis
Virtual Cell
WinSCAMP

MathSBML: A Tool for Manipulating SBML Models

We have developed MathSBML, an ***open-source, freely available software package*** (LGPL license) for working with Systems Biology Markup Language (SBML) models. MathSBML is a *Mathematica* package that allows users to read, write, and modify SBML files; view and interpret their contents; convert models to other formats such as FORTRAN, XPP, and HTML; and run numerical simulations.

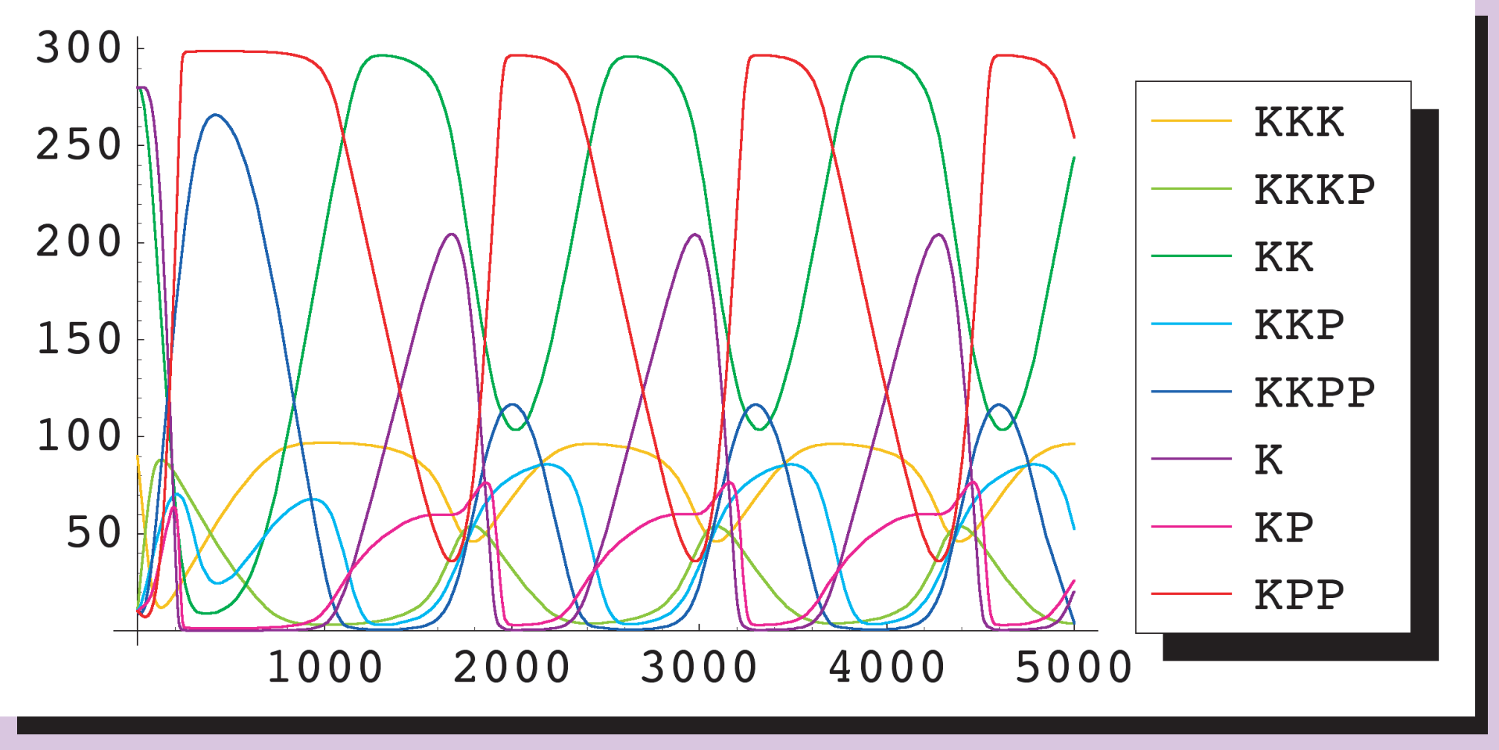
IAt is simplest, MathSBML can generate an interpretive, tabular listing of the model in terms of its individual components. However, it takes the interpretation one step further by translating all of the component reactions and rules into a combined differential/algebraic system representing all model variables (species and parameters).

When populated with appropriate initial conditions and constant and parameter values, this differential/algebraic system can be solved numerically; MathSBML contains a simulator component that performs this function.

Supppose for example, that the MAPK model of [Kholodenko, 2000] has been encoded in the SBML file MAPK.xml (this model is available from the SBML model repository at http://www.sbml.org; some of the variable names have been simplified for the current presentation, and the file has been updated to SBML level 2, but are otherwise unchanged. The model is illustrated to the right using the SBML on-line tool suite [Mathews, 2003]) . The model can be read, run, and plotted wtih the three following commands:

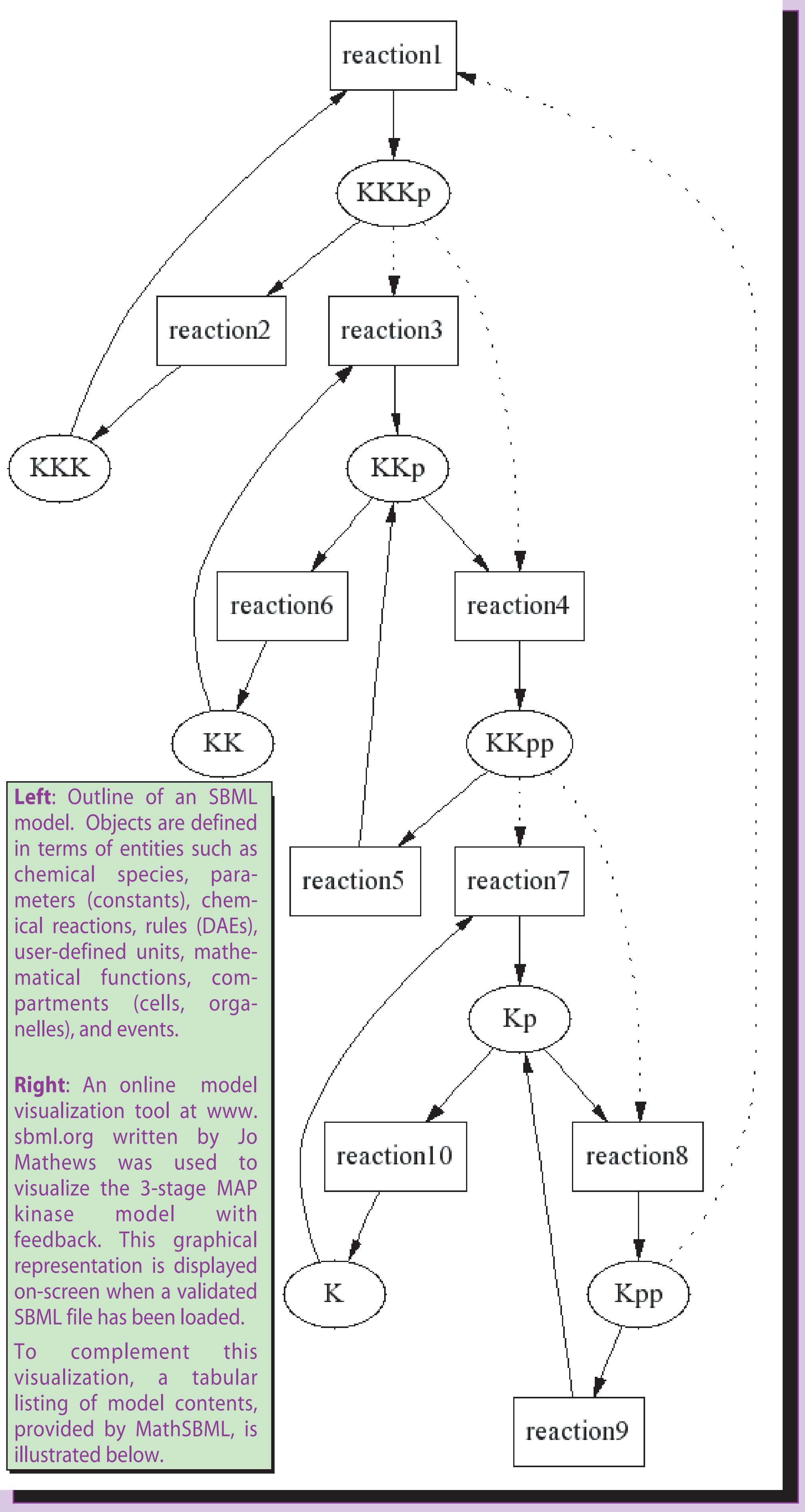
```
m = SBMLRead["desktop/PSB/MAPK.xml"];
s = SBMLNDSolve[m, 5000];
SBMLPlot[s];
```

MathSBML is implemented as a *Mathematica* package. The first line above reads the file into an internal data structure that is compatible with standard *Mathematica* functions such as NDSolve. In this data structure all rules and reactions contained in the SBML are translated into a system of differential/algebraic equations. The second line solves the system numerically, and the third generates a plot. Advanced *Mathematica* "power users" can apply standard functions like NDSolve directly on the output of SBMLRead if so desired.



Plot of the MAP-kinase cascade model generated with MathSBML.

```
<?xml version="1.0"
      encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.
org/sbml/level2" level="2"
version="1">
 <model id="ex1">
  <listOfFunctionDefinitions>
   ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
   ...
  </listOfUnitDefinitions>
  <listOfCompartments>
   ...
  </listOfCompartments>
  <listOfSpecies>
   ...
   </listOfSpecies>
  <listOfParameters>
   ...
  </listOfParameters>
  <listOfRules>
   ...
  </listOfRules>
  <listOfReactions>
   ...
  </listOfReactions>
  <listOfEvents>
   ...
  </listOfEvents>
 </model>
</sbml>
```



Left: Outline of an SBML model. Objects are defined in terms of entities such as chemical species, parameters (constants), chemical reactions, rules (DAEs), user-defined units, mathematical functions, compartments (cells, organelles), and events.

Right: An online model visualization tool at www.sbml.org written by Jo Mathews was used to visualize the 3-stage MAP kinase model with feedback. This graphical representation is displayed on-screen when a validated SBML file has been loaded.

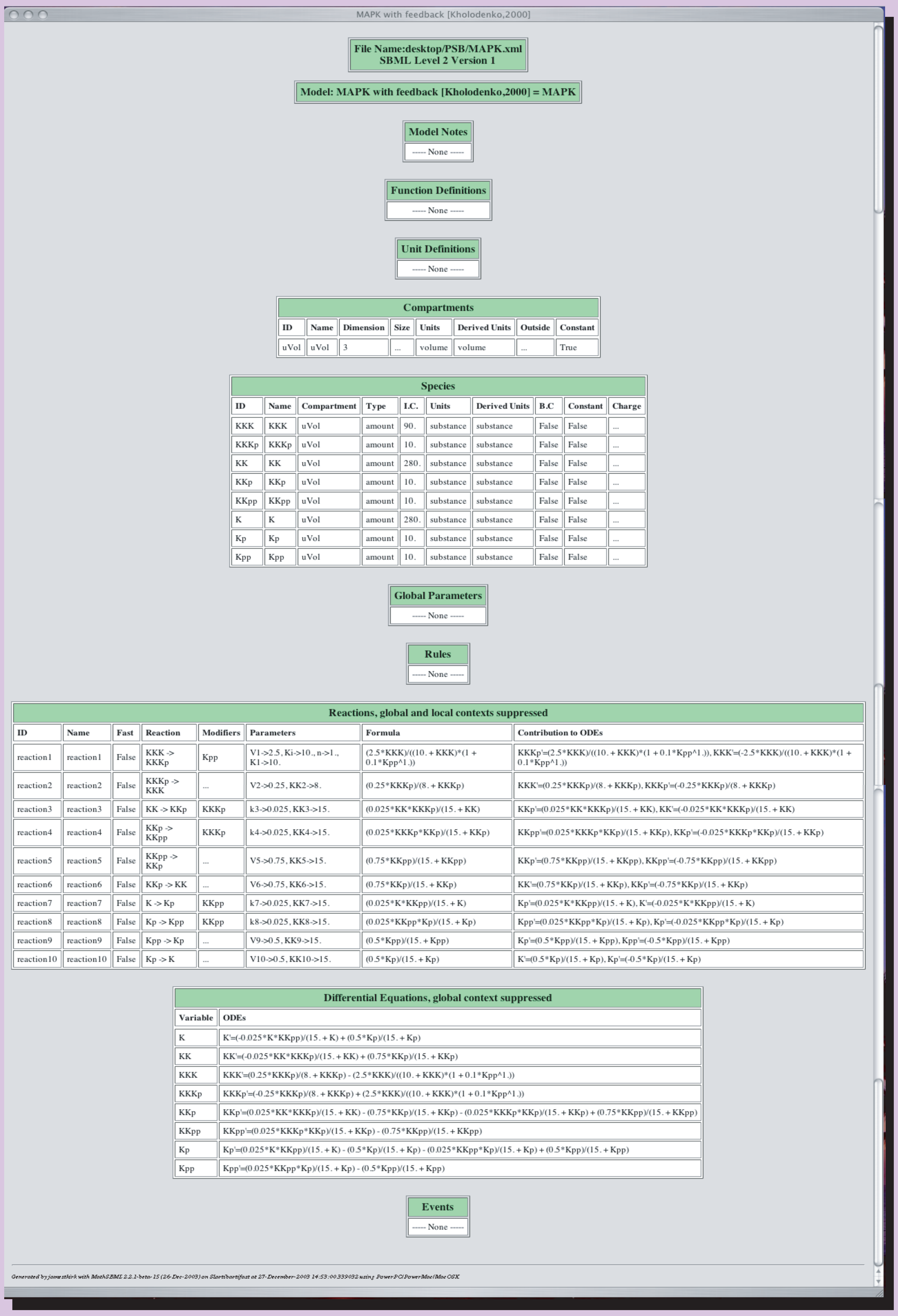
To complement this visualization, a tabular listing of model contents, provided by MathSBML, is illustrated below.

Viewing SBML Models

MathSBML does not provide any graphical visualization tools, although such a tool is available on the SBML web site; the MAPK model visualized above to the right is an example.

To complement this visualization, MathSBML provides a method for translating an SBML file into a human-readable tabular format. For example, the same MAPK file can be translated to to HTML (as illustrated below) via the single command

```
SBMLWrite[inputfile->"desktop/PSB/MAPK.xml",
 outputfile->"desktop/PSB/MAPK.html", format->"HTML"]
```



MAPK with feedback [Kholodenko,2000]

File Name:desktop/PSB/MAPK.xml
SBML Level 2 Version 1

Model: MAPK with feedback [Kholodenko,2000] = MAPK

Model Notes
----- None -----

Function Definitions
----- None -----

Unit Definitions
----- None -----

Compartments

ID	Name	Dimension	Size	Units	Derived Units	Outside	Constant
uVol	uVol	3	...	volume	volume	...	True

Species

ID	Name	Compartment	Type	I.C.	Units	Derived Units	B.C	Constant	Charge
KKK	KKK	uVol	amount	90.	substance	substance	False	False	...
KKKp	KKKp	uVol	amount	10.	substance	substance	False	False	...
KK	KK	uVol	amount	280.	substance	substance	False	False	...
KKp	KKp	uVol	amount	10.	substance	substance	False	False	...
KKpp	KKpp	uVol	amount	10.	substance	substance	False	False	...
K	K	uVol	amount	280.	substance	substance	False	False	...
Kp	Kp	uVol	amount	10.	substance	substance	False	False	...
Kpp	Kpp	uVol	amount	10.	substance	substance	False	False	...

Global Parameters
----- None -----

Rules
----- None -----

Reactions, global and local contexts suppressed

ID	Name	Fast	Reaction	Modifiers	Parameters	Formula	Contribution to ODEs
reaction1	reaction1	False	KKK -> KKKp	Kpp	V1->2.5, Ki->10., n->1., K1->10.	(2.5*KKK)/((10. + KKK)*(1 + 0.1*Kpp^1.))	KKKp'=(2.5*KKK)/((10. + KKK)*(1 + 0.1*Kpp^1.)), KKK'=(-2.5*KKK)/((10. + KKK)*(1 + 0.1*Kpp^1.))
reaction2	reaction2	False	KKKp -> KKK	...	V2->0.25, KK2->8.	(0.25*KKKp)/(8. + KKKp)	KKK'=(0.25*KKKp)/(8. + KKKp), KKKp'=(-0.25*KKKp)/(8. + KKKp)
reaction3	reaction3	False	KK -> KKp	KKKp	k3->0.025, KK3->15.	(0.025*KK*KKKp)/(15. + KK)	KKp'=(0.025*KK*KKKp)/(15. + KK), KK'=(-0.025*KK*KKKp)/(15. + KK)
reaction4	reaction4	False	KKp -> KKpp	KKKp	k4->0.025, KK4->15.	(0.025*KKKp*KKp)/(15. + KKp)	KKpp'=(0.025*KKKp*KKp)/(15. + KKp), KKp'=(-0.025*KKKp*KKp)/(15. + KKp)
reaction5	reaction5	False	KKpp -> KKp	...	V5->0.75, KK5->15.	(0.75*KKpp)/(15. + KKpp)	KKp'=(0.75*KKpp)/(15. + KKpp), KKpp'=(-0.75*KKpp)/(15. + KKpp)
reaction6	reaction6	False	KKp -> KK	...	V6->0.75, KK6->15.	(0.75*KKp)/(15. + KKp)	KK'=(0.75*KKp)/(15. + KKp), KKp'=(-0.75*KKp)/(15. + KKp)
reaction7	reaction7	False	K -> Kp	KKpp	k7->0.025, KK7->15.	(0.025*K*KKpp)/(15. + K)	Kp'=(0.025*K*KKpp)/(15. + K), K'=(-0.025*K*KKpp)/(15. + K)
reaction8	reaction8	False	Kp -> Kpp	KKpp	k8->0.025, KK8->15.	(0.025*KKpp*Kp)/(15. + Kp)	Kpp'=(0.025*KKpp*Kp)/(15. + Kp), Kp'=(-0.025*KKpp*Kp)/(15. + Kp)
reaction9	reaction9	False	Kpp -> Kp	...	V9->0.5, KK9->15.	(0.5*Kpp)/(15. + Kpp)	Kp'=(0.5*Kpp)/(15. + Kpp), Kpp'=(-0.5*Kpp)/(15. + Kpp)
reaction10	reaction10	False	Kp -> K	...	V10->0.5, KK10->15.	(0.5*Kp)/(15. + Kp)	K'=(0.5*Kp)/(15. + Kp), Kp'=(-0.5*Kp)/(15. + Kp)

Differential Equations, global context suppressed

Variable	ODEs
K	K'=(-0.025*K*KKpp)/(15. + K) + (0.5*Kp)/(15. + Kp)
KK	KK'=(-0.025*KK*KKKp)/(15. + KK) + (0.75*KKp)/(15. + KKp)
KKK	KKK'=(0.25*KKKp)/(8. + KKKp) - (2.5*KKK)/((10. + KKK)*(1 + 0.1*Kpp^1.))
KKKp	KKKp'=(-0.25*KKKp)/(8. + KKKp) + (2.5*KKK)/((10. + KKK)*(1 + 0.1*Kpp^1.))
KKp	KKp'=(0.025*KK*KKKp)/(15. + KK) - (0.75*KKp)/(15. + KKp) - (0.025*KKKp*KKp)/(15. + KKp) + (0.75*KKpp)/(15. + KKpp)
KKpp	KKpp'=(0.025*KKKp*KKp)/(15. + KKp) - (0.75*KKpp)/(15. + KKpp)
Kp	Kp'=(0.025*K*KKpp)/(15. + K) - (0.5*Kp)/(15. + Kp) - (0.025*KKpp*Kp)/(15. + Kp) + (0.5*Kpp)/(15. + Kpp)
Kpp	Kpp'=(0.025*KKpp*Kp)/(15. + Kp) - (0.5*Kpp)/(15. + Kpp)

Events
----- None -----

Generated by jamestkirk with MathSBML 2.2.1-beta-15 (26-Dec-2003) on Slartibartifast at 27-December-2003 14:53:00.339032 using PowerPC/PowerMac/Mac OSX

The MathSBML Model Editor Provides an API for SBML Model Manipulation

MathSBML contains a simple model editor, allowing users to create SBML models compatible with other simulators, as well as a *Mathematica* text-command based API that can be used to produce arbitrarily complex SBML files.

The model editor contains a suite of commands to add, modify, or remove single SBML objects (such as a reaction, chemical species, or equation) from the current model. The model may be either created de-novo or read from an external file. After building the model, the user can test it by running simulations, continue to modify it, or write the results as an SBML file., in any order.

To build a model of a simple enzymatic reaction, for example,

$$S+Enz \underset{k_2}{\overset{k_1}{\rightleftarrows}} SE \xrightarrow{k_3} P+Enz$$

one would enter the following sequence of commands:

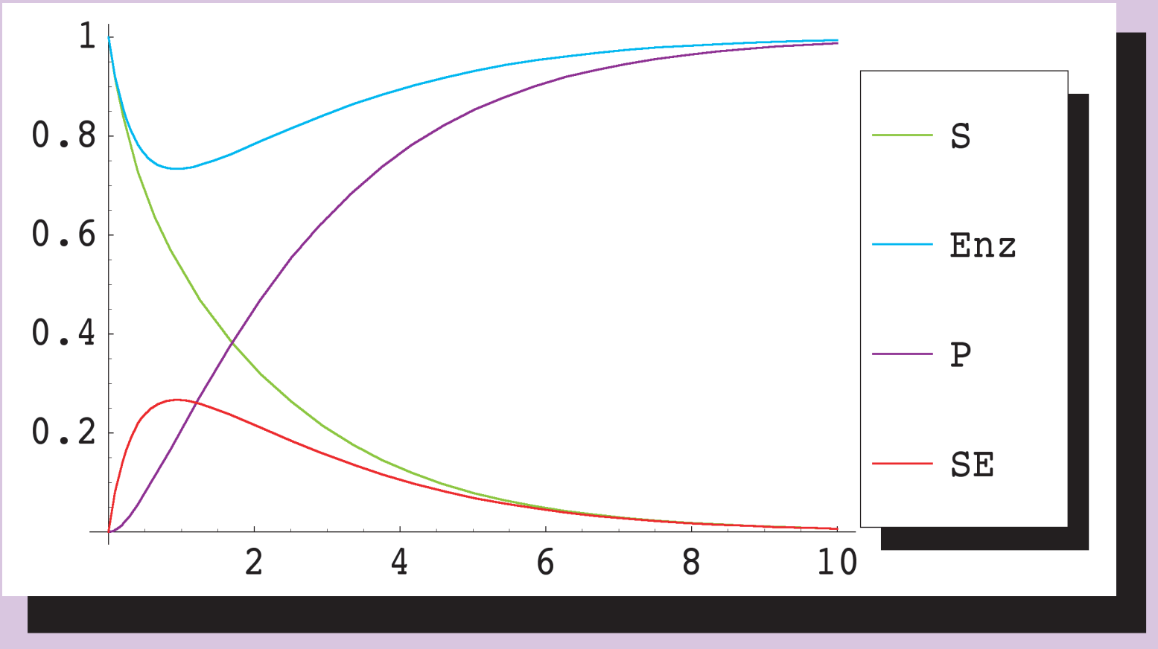
```
newModel["EnzymeKinetics"];
addSpecies[S, initialConcentration -> 1];
addSpecies[Enz, initialConcentration -> 1];
addSpecies[P, initialConcentration -> 0];
addSpecies[SE, initialConcentration -> 0];
addReaction[S+Enz->SE,kineticLaw->k1*S*Enz,
  parameters->{k1->1}];
addReaction[SE->S+Enz,kineticLaw->k2*SE,
  parameters->{k2>0.5}];
addReaction[SE->P+Enz,kineticLaw->k3*SE,
  parameters->{k3->1}];
```

It is possible to generate an SBML file form the model just constructed with

```
createModel["Desktop/PSB/Enzyme.xml"];
```

Alternatively, one may want to explore the model first. If all parameters and initial conditions have numerical values one could run a simulation and plot the results,

```
SBMLNDSolve[ loadSimulator[ ], 10 ]//SBMLPlot;
```



The functions `loadSimulator` and `createModel` can be performed in either order.

Extensibility and Compatibility

Since MathSBML is implemented as a *Mathematica* package, it is fully extensible and is compatible with all *Mathematica* capabilities such as the solution of systems of differential and algebraic equations, numerical integration and phase plane analysis.

MathSBML supports SBML Level 1, Version 1 and Version 2, as well as SBML Level. The MathSBML Model Editor supports SBML Level 2. Events are not currently supported in the simulator, but this capability will be added in the future.

MathSBML will run on any platform that has *Mathematica* 4.1 or higher installed. The solution of differential-algebraic systems (SBML models that have algebraic rules) requires *Mathematica* 5.0 or higher; purely differential systems (SBML without algebraic rules) can be solved on *Mathematica* 4.1.

MathSBML has been validated with over 4000 different models derived from various sources such as KEGG.

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 at:

http://www.sf.net/projects/sbml

Extensive documentation is included with the download, and is also available online at the SBML web site:

http://www.sbml.org/mathsbml.html

Acknowledgements

Development of MathSBML was supported by the JST ERATO Kitano Symbiotic Systems Project.

References

Finney A, Hucka M, "Systems biology markup language: Level 2 and beyond," *Biochem. Soc. Trans.* (2003) 31:1472-1473.

Hucka M et al, “The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models”, *Bioinformatics* 19: 524-531 (2003).

Kholodenko B, "Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades," *Eur. J. Biochem, 267*:1583-1588 (2000).

Mathews J, "Active web pages supporting SBML," 8th SBML Workshop, St. Louis, MO,USA (2003).

For additional information contact via email: bshapiro@caltech.edu