Systems Biology
Markup Language

SBML Frequently Asked Questions

Andrew Finney, Michael Hucka
{afinney,mhucka}@caltech.edu
ERATO Kitano Symbiotic Systems Project
Control and Dynamical Systems, MC 107-81
California Institute of Technology, Pasadena, CA 91125, USA
http://www.sbml.org/

Principal Investigators: John Doyle and Hiroaki Kitano

Revision : 1.24

September 2, 2003

This Frequently Asked Questions (FAQ) document answers questions about the Systems Biology Markup Language (SBML). It is a non-normative document that does not *define* any aspect of SBML; rather, it is intended to provide additional information about the language in an easily accessible and readable form. Only the SBML specifications define SBML. (See Section 4 for information on where to find the specification documents.)

The latest version of this FAQ is available in different formats at the following online locations:

HTML:	http://www.sbml.org/faq/sbml-faq.htm
PDF:	http://www.sbml.org/faq/sbml-faq.pdf
PostScript:	http://www.sbml.org/faq/sbml-faq.ps

If the question you have is not covered here, please consider submitting it for inclusion into a future copy of this FAQ. (See Question 9.1 for information about how to do this.)

FAQ Contents

1 SBML Introduction and Background

1.1 What is SBML?

The Systems Biology Markup Language (SBML) is a machine-readable format for describing qualitative and quantitative models of biochemical networks. It can also be used to express the interactions of biochemical networks with other phenomena. By a "biochemical network", we mean a system consisting of biochemical entities linked by chemical reactions that alter, transport and/or transform the entities.

The primary encoding of SBML is XML, a popular text-based language for expressing structured data in a generic fashion. As a result, software developers can quickly write tools to manipulate SBML by reusing existing XML software tools.

1.2 What is the purpose of SBML?

SBML is designed to enable the exchange of biochemical network models between different software packages. The aim is to (1) enable the enable the exchange of models between software tools with little or no human intervention, thus allowing the tools to be properly integrated; and (2) enable these models to be published in electronic form alongside, for example, peer-reviewed journal articles. The emphasis in SBML is on supporting quantitative models.

1.3 What can be represented in SBML?

An SBML model consists of a set of chemical entities linked by reactions that can transform one entity into another or transport entities between compartments. A *compartment* in SBML is a location having a defined size or extent (which may be in terms of volume, area, or length). Every chemical species in an SBML model must be located in a compartment.

SBML can represent not only biochemical networks, but also quantitative models of interaction between these networks and other phenomena. SBML can also describe discrete events that are triggered by state changes in the modeled system. The scope of SBML is constantly evolving through a community-led development.

1.4 Is SBML a database schema?

Not specifically. There is no reason why SBML models could not be stored in a database, nor is there any reason why you could not use SBML as a schema. However, this was not the motivation for creating SBML. An SBML model is meant to encode a consistent view of knowledge of a biological system. SBML is not meant to encode a large set of potentially conflicting knowledge about such a system.

1.5 Who developed SBML?

SBML originated out of a workshop on software platforms for systems biology held in the year 2000 with funding by the Japan Science and Technology Corporation (JST). The original authors were Michael Hucka, Andrew Finney, Herbert Sauro and Hamid Bolouri, but SBML is now very much a community effort. Please see Section 8 for more details about the history of SBML. See Section 7 about how you can participate in SBML's evolution.

1.6 Where is SBML defined?

The Systems Biology Markup Language is formally defined in specification documents (see Question 4.2).

1.7 Where is the SBML web site?

The home website for SBML is `http://www.sbml.org/`.

2 SBML Support

2.1 Which applications support SBML?

The matrix in Table 1 lists the software tools known to support SBML, along with information about which level of SBML they support. The names of the packages are clickable URLs pointing to more information about them. (If you know of other SBML-compatible software not listed here, please inform the editors of this FAQ.)

Table 1: Table of applications known to support SBML.

	SBML Level 1		SBML Level 2	
Application	**Read**	**Write**	**Read**	**Write**
BASIS		■		
BioSketchPad				■
BioSpreadsheet			■	■
BSTLab	■	■		
CADLIVE				
CellDesigner	■	■		
Cellerator		■		
Cellware			(in development)	
Cytoscape	■	■		
E-Cell			(in development)	
ESS			■	■
Gepasi	■	■		
Jarnac	■	■		
JDesigner	■	■		
JigCell			■	■
JSIM	■			
LIBSBML	■	■	■	■
MathSBML	■	■		
MicroCore PIMs	■	■	■	■
MicroCore MicroExpress	■	■	■	■
MOMA				■
Monod				■
NetBuilder		■		
PathArt		■		
PathScout		■		
SBML DLL Library	■	■		
SigPath		■		
StochSim		■		
Teranode				
Trelis	■	■		
Virtual Cell	■	■		
WinSCAMP (beta)	■	■		

2.2 Are software libraries available for programming with SBML?

The matrix in Table 2 lists software libraries known to support SBML. (If you know of others not listed here, please inform the editors of this FAQ.)

Table 2: Table of software libraries for SBML.

		SBML Level 1		SBML Level 2	
Library	**Language**	**Read**	**Write**	**Read**	**Write**
LIBSBML	C/C++	■	■	■	■
MathSBML	Mathematica	■	■		
SBML Windows DLL	(Windows DLL)	■	■		
JigCell's `sbml.jar`	Java			■	■

2.3 Are there large groups using SBML?

The following large consortia are known to us to be using SBML as their standard model definition language.

- DARPA BioSPICE
- International E. coli Alliance (IECA)

2.4 Where can I find examples of SBML models?

The specification documents for SBML, available from the project website (`http://www.sbml.org/`), includes numerous simple examples. The web site also hosts a repository of SBML models drawn from a number of published articles. Finally, the site contains a link to a large test suite containing thousands of model files in SBML Level 1 and Level 2 format. The test suite is also available directly for downloading from the SBML project area on SourceForge.net, at `http://sourceforge.net/projects/sbml`.

3 SBML Levels

3.1 What are SBML "Levels"?

SBML is being developed in a series of *levels*, where each level adds new features and fixes problems with the previous level. The lowest-numbered levels provide fundamental features that are common to all biochemical network models. Higher-numbered levels add more features that are specific to particular classes of tools. Any level can be used as a standard for interchanging models.

3.2 What is the current SBML Level?

SBML Level 2 Version 1 was finalized in June 2003. All new development is oriented towards Level 3.

3.3 What are the differences between Levels 1 and 2?

The changes in SBML Level 2 include: replacing SBML Level 1's text-string based format for mathematical expressions with MathML (a W3C standard), introducing support for metadata using the same metadata scheme as CellML, introducing support for named function definitions, introducing explicit modifier species such as catalysts in reactions, and introducing new constructs for discrete events and time delays. There are several other small changes introduced in Level 2. The complete list of changes is documented in the SBML Level 2 specification (see Question 4.2).

3.4 Why is Level 1 still being kept around if Level 2 has been defined?

There exist tools that either were developed before the creation of SBML Level 2 or for which Level 1 is more appropriate. Thus, SBML Level 1 continues to have relevance even with the existence of Level 2.

Note that since all Level 1 models can be translated to SBML Level 2, tools that read SBML Level 2 can be made to support Level 1 reasonably easily. Moreover, the availability of LIBSBML makes it much easier for application developers to support different SBML levels in software applications. Among other features, LIBSBML has a built-in Level 1 to Level 2 translation facility. (See Question 2.2.)

3.5 What is the point of having a Level 1 Version 2? Why not forget about it now that Level 2 is out?

First, please refer to the previous question about why it is important to have both Level 1 and Level 2 coexist. As to the question of why bother with a Version 2 definition, there were many reasons why it became necessary to introduce a Version 2 of Level 1. Here is a sample of the reasons:

- Some things needed to be clarified, like the fact that stoichiometric constants were supposed to be *positive* integers but the original Level 1 Version 1 specification never mentioned this.
- There were some inconsistencies in the original specification, such as whether the annotation attribute on SBase was suppposd to be spelled `annotation` or `annotations`.
- People kept requesting the ability to make more things optional in models, such as compartments.
- People repeatedly complained about certain issues such as the spelling of "species" vs "specie".

Level 1 is meant to coexist with Level 2, therefore it ought to be as good as we can make it because people will (hopefully) be using it. If people complain about problems, and we have the ability to fix them, and the fixes are basically isolated from people's code in parser libraries such as Ben Bornstein's LIBSBML, why not make the fixes?

3.6 What features are anticipated in Level 3?

People interested in SBML have organized themselves into a number of working groups focused on different topics. The topics of these working groups give an indication of the features anticipated for SBML Level 3 (see Question 7.6 for a list of working groups).

4 Specifications and Documentation

4.1 Are there publications about SBML?

There are currently two:

- *The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models*

  Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A. , Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novère, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J., *Bioinformatics*, 2003, vol. 19, no. 4, pp. 524–531.

- *Systems Biology Markup Language: Level 2 and Beyond*

  Finney, A., and Hucka, M., *Biochemical Society Transactions*, in press.

4.2 Where is SBML defined?

The following are the specification documents for the different levels of SBML:

- SBML Level 2. This is the latest edition of SBML.
- SBML Level 1 Version 2. This is the latest version of SBML Level 1 and supercedes the original SBML Level 1 Version 1 definition. As explained elsewhere (Question 3.4), Level 2 and Level 1 are intended to coexist—the availability of Level 2 does not render Level 1 obsolete.

- SBML Level 1 Version 1. Please note that SBML Level 1 Version 1 is now deprecated in favor of Version 2. SBML Level 1 Version 1 should no longer be used for any new development. Moreover, we encourage developers to use SBML Level 2 in their software. See Question 6.1.

SBML Level 2 makes use of other standards and specifications. In particular, the following are important references:

- The MathML 2.0 specification
- The CellML Metadata specification

5 The Design of SBML

5.1 What is the basic idea behind the SBML units system?

The idea is that the units associated with every math entity in a model should be precisely defined, while at the same time allowing for reasonable default unit definitions. The set of math entities important in this regard includes variables, parameters and the result of equations.

The motivation for having a units system in SBML is threefold: it allows the semantics of math entities to be defined precisely, it allows a consistent method for handling multi-compartmental models, and in the longer term, it will allow for consistency-checking and unit conversions in models that are composed from submodels.

5.2 Are the units used in a model not well-defined unless the model contains unit attribute values?

In a model that doesn't define values for unit attributes, only the units of parameters (and their rules) are undefined. The units of species, compartments and kinetic laws are well-defined via built-in defaults.

5.3 But isn't allowing default units a bit of cop-out?

The specification states that all math entities apart from parameters have precisely defined units. Unfortunately, the majority of software tools today offer little or no support for units. We could not have interoperability between tools if we imposed a mandatory, explicit units system.

5.4 Not having a precise definition of parameter units is a big hole, isn't it?

In Level 1 and Level 2, the only practical use for these would be for documentation purposes. We don't and can't expect software to determine and check the effective units of parameters by analyzing equations. In addition, many tools don't require units on parameters, so we allow parameter units to be optional.

We anticipate that some systems supporting model composition in Level 3 will check parameter units. However, it's still an open question to many people whether tools should impose constraints on parameter units and/or perform conversions between math entities as part of the composition process.

5.5 What are the 'built-in' units exactly?

There are built-in units for substance, length, area, volume and time. These units are, by default, used to form the units of various math entities in a SBML model. For example, by default, a species symbol in a kinetic law equation has substance/volume units (assuming that the species' compartment has 3 spatial dimensions).

These built-in units are in turn defaulted to specific units. For example, *substance* defaults to units of moles. An SBML model can explicitly set the underlying units of a built-in unit; for instance, it is possible to redefine *substance* to be millimole.

5.6 What are the benefits of the ‘built-in’ units?

The great benefit of having built-in units is that it allows a model to redefine the underlying units of a whole model precisely without having to assign units to every math entity explicitly.

The SBML specification ensures that the built-in units operate in a consistent and reasonable way across math entities.

5.7 How should models without compartments be encoded?

The models should locate all species in a single compartment with unit volume. The default units system of SBML will ensure that this unit volume representation is exactly equivalent to a model dealing with concentrations, including rate laws defined in substance/volume/time units.

5.8 Why doesn’t SBML Level 2 define a default compartment with unit volume?

There are several reasons:

1. A model that uses a single unit-volume compartment is making explicit an important underlying assumption about the model. Leaving it implicit would be more prone to errors.
2. A default compartment would be a special case which all SBML parsing programs would have to handle specially.
3. We would have to invent a reserved name to refer to the default compartment.
4. It would only save effort in developing the SBML writing component of a software tool. The writing component is the easy part; reading and interpreting is the harder part.

5.9 Why does SBML include “low-level” features such as rules in combination with biochemical concepts like reactions and species?

The aim of SBML is to enable the construction of quantitative models that describe both the activity of biochemical networks and interaction of biochemical networks and other phenomena. SBML allows the declaration of variables (non-constant parameters) and associated ODEs and DAEs to describe these phenomena. Examples of these phenomena include the mechanical force generated by muscle cells and the electrical potential across a synapse.

5.10 Why is there a distinction between assignment and algebraic rules? Aren’t they equivalent?

Although it is typically easy to transform between assignment and algebraic rules, we make the distinction in SBML because:

- Algebraic rules define the point in the model where there is a circular dependency between variables. For instance, the equations $x = 2y$ and $y = x + 1$ have a circular dependancy. It is not possible to form such a dependancy in scalar rules (see the SBML Level 2 specification). At least one of the example equations would have to be encoded as an algebraic rule in SBML.
- Many tools are not capable of supporting algebraic rules (DAEs)
- Those tools that do support make the distinction between assignment rules and algebraic rules.

5.11 Why can’t functions be recursive in Level 2?

Functions definitions in SBML Level 2 are designed to allow them to be substituted in place of the function call operator; that is, they are deliberately defined so that software tools can treat them like macros rather than functions. This would not be possible if functions were allowed to be recursive.

5.12 What on earth are “events”?

“Events” are discrete discontinuous events that can be triggered in response to state transitions in a model.

5.13 Is it possible to represent an entirely event-driven, deterministic model in SBML Level 2?

Yes, although we are not aware of any simulators that can support this kind of model.

5.14 What's the difference between the `boundaryCondition` and `constant` attributes on species in Level 2?

Level 2 introduces the `constant` to the `Species` structure in addition to the `boundaryCondition` from Level 1. One might expect these attributes to both define that the species doesn't vary during simulation; however, there is a difference in their semantics. The `constant` attribute indicates that the species concentration doesn't vary during simulation no matter what reactions or rules it occurs in. Such a constant species can't have an associated rule. The `boundaryCondition` attribute defines only that the species concentration is not defined by the set of reactions.

5.15 Why were `constant` attributes introduced in SBML Level 2?

Given a model that doesn't contain algebraic rules it is possible to infer which components (species, compartment and parameters) are meant to be variables by examining the set of scalar rules, rate rules and reactions. However given a model containing algebraic rules knowledge of which symbols are variables and which are constants is required to solve the equations. The occurence of a symbol in an algebraic rule doesn't imply that the symbol is a variable.

5.16 Why isn't there an explicit definition of SBML in terms of an ODE equivalent form?

On one hand, we were worried that a focus on the ODE representation would deter developers that employ other forms, for example stochastic discrete event simulation, from supporting SBML.

On the other hand, we simply have not had time to work on this area. We would welcome volunteers to work on this. There is an example translation of a model into ODEs in the specification for Level 2.

6 Implementing SBML support

6.1 Which level of SBML should I use in my software?

We recommend SBML Level 2, despite the fact that at the moment more tools support Level 1, because Level 2 fixes known problems with Level 1 and we anticipate that Level 2 will be forward-compatible with Level 3.

6.2 Is it a good idea to use SBML as my software's native model format?

Depending on the needs of your software, yes, this may be a good idea. JDesigner, JigCell and CellDesigner are examples of software tools that use SBML as their native formats.

6.3 What software libraries are available to help me program support for SBML in my software?

See Question 2.2.

6.4 The unit system seems to imply some implicit unit conversions may be required in various places when parsing a model. Can you give an overview of what you expect a model parser to do?

The first step is to establish what the units are of the various entities in your object model. These units may be generic and thus can be derived from the SBML built-in units.

There are 3 places where unit conversion may have to occur depending on the units used for variables internal to a simulation:

- *symbols in formulae*

Variables that are used in formulae may require a conversion from the object model units to the units associated with the SBML symbol before the SBML formulae are applied to the variables.

- *result of formulae*

  The result of SBML formulae may require a conversion from SBML units associated with the formulae result to the units associated with the simulation variables that are assigned the result of formulae. Particular attention should be placed on the results of kinetic laws which have substance/time units in SBML.

- *initial conditions*

  The initial values of variables may require conversion from the SBML units associated with the symbol to the object model units of the variable.

6.5 My software cannot support all the features of SBML—what should I do?

Don't Panic. Few if any packages support all aspects of SBML. Depending on what you wish do with a model you can either ignore features that are not relevant to your tool or report errors when certain features are used. In most cases a simulator will simply report an error if it encounters a feature that it is unable to support. Ignoring quantitative aspects of an SBML model when parsing it into an simulator or analysis tool is strongly discouraged, because it will make it unlikely a user will be able to reproduce results seen in the tool which generated the model.

6.6 What do I do about the fact that SBML does not encode all the information that I need to encapsulate in a model?

You should use `Annotation` elements. These are described in some detail in the Level 1 and Level 2 specification documents. `Annotation` elements can be enclosed within any SBML element and can contain elements of any namespace. Elements that are not in the SBML namespace should either locally redefine the default namespace or use namespace prefixes to clearly separate application specific data from different software packages. Data stored in annotation elements should not contain data that could be or is encoded in SBML.

6.7 How should I structure annotations?

The annotation data enclosed in a specific SBML element is assumed by other applications to be directly associated with that specific element. Therefore it is important to decompose and locate annotation data appropriately in an SBML document. Avoid, for example, encoding all annotation data in a top level attribute. The data associated with, for example, an individual species should be encoded in the annotation element enclosed within the SBML species element representing that species.

6.8 How should I attach standard database identifiers, for example GO terms, to SBML elements?

Database identifier annotations can be created using CellML metadata in SBML Level 2; see the specification for CellML Metadata, in particular Section 4.10, "Biological Entity". SBML elements derived from `SBase` have a `metaid` attribute, which should contain a value unique to the SBML document. RDF elements can use this value as a reference to the specific element.

Its possible to use a simpler form of database identifier annotation than that shown in the CellML specification. For example here's an SBML species annotated with a GO term:

```
<species id="foo" metaid="x1"/>
...
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:cmeta="http://www.cellml.org/metadata/1.0#">
  <rdf:Description about="#x1">
    <cmeta:bio_entity>
         <cmeta:identifier rdf:parseType="Resource">
            <cmeta:identifier_scheme>GO Consortium</cmeta:identifier_scheme>
            <rdf:value>GO:0048038</rdf:value>
         </cmeta:identifier>
    </cmeta:bio-entity>
  </rdf:Description>
</rdf:RDF>
...
```

6.9 What should I do when I encounter an incorrect SBML file or stream?

Whilst it is not expected that an application will detect all errors in an parsed SBML document it is expected that a parser will not simply ignore or fudge discrepancies between the document and the relevant SBML standard. The parser should report the line number and a description of the error to the user i.e. a parser should not assume that an incorrect SBML file is an extremely unusual event or that the details of a parsing error are irrelevant to the user.

If you encounter consistent differences between the standard specification and documents that claim to be compliant then please report this to the sbml-discuss mailing list.

6.10 There are several different ways in which I could encode my models in SBML. Which forms are more interoperable?

It is true that there are several ways in which to encode a given model in SBML, deciding which form to use in various contexts is a big subject. Although this section describes which SBML features are commonly used tools should support as wide a set of features of SBML when reading SBML as possible. Here is a rough guide:

6.10.1 Identifiers

Although `id` fields have restricted syntax do assume that tools could potentially display the contents of this field to users even when `name` attributes are available. This means that if you capture names from a UI that this data is placed in both the `name` and `id` fields (with the `id` field value perhaps slightly mangled). Avoid using crude automatically generated values in the `id` field.

6.10.2 Stoichiometry

Whilst it is possible to encode complex math expressions to specify stoichiometry it is best to restrict these expressions to being just positive integer values.

6.10.3 Species initial value

A species initial value can be specified using `initialAmount` or `initialConcentration`. The attribute most commonly used is `initialConcentration`.

6.10.4 Compartment spatial dimensions

Common practice is to allow the `spatialDimensions` attribute to default to 3. In fact the majority of simulators only support the value 3 for this attribute.

6.10.5 Species References

A reaction's product and reactant lists can contain more than one species reference structures referring to the same species. This is not however good practice for interoperability.

6.10.6 Rules verses Reactions

In all cases where possible a reaction structure is preferable to the equivalent rule structures even if this means the use of reactions which have no reactants or products.

6.10.7 Algebraic Rules verses Assignment Rules

If it is possible to avoid using algebraic rules then do not use them.

6.10.8 Discontinuities

Avoid the use of discontinuous operators like `piecewise`.

6.10.9 Delay

Avoid the use of the built-in delay operator.

6.10.10 Events

Avoid the use of the events.

6.10.11 Units

Do not expect tools to interpret units. Ensure your tool can parse models which use entirely default units.

6.11 How much effort should I invest in preserving the SBML form when round-tripping models through my software?

The first priority should be to support as much of the SBML standard as possible both for reading and writing. You should write using the most interoperable form as possible as described in Question 6.10. To maximize interoperability beyond this requires trying to include as much of an imported SBML model as possible when rewriting it in SBML. This includes preserving annotation data and avoiding mangling `id` and `name` fields. The order of structures, other than rules, and the white space between elements do not require preservation.

7 Organization

7.1 What is the overall SBML development process?

SBML development has been and continues to be motivated and directed by the systems biology community. The process is managed by the *SBML Editors* (see next question), but they do so under the control of the community. The editors collect proposals for changes to SBML from the SBML Working Groups and from other groups and individuals, and then seek to establish a consensus in the community about how to proceed with the proposals. With this information, the editors assemble some of the proposals into a draft specification for a new edition of SBML. After this draft has been reviewed by the community, it becomes a final specification for the new edition of SBML. ("Edition" in this context can be either a new SBML Level, or a new version of an existing level.)

7.2 Who are the "SBML Editors"?

Currently, the editors are Andrew Finney and Mike Hucka

7.3 What do SBML Editors do?

The following are the roles of the SBML editors:

- assemble proposals into SBML Level specifications
- organize SBML forum meetings
- maintain the `http://www.sbml.org` website
- publicize and document the SBML standard
- support developers who wish to work with SBML
- support the SBML Working Groups in developing proposals

7.4 What are “SBML Forum” Meetings?

These are biannual face-to-face meetings organized by the SBML editors. The formal title of the meetings is the *Workshops on Software Platforms for Systems Biology*. These are held in early summer and early winter. The winter meeting is normally held in conjunction with the International Conference for Systems Biology (ICSB). SBML Forum meetings allow for significant discussion of new SBML proposals and interoperability issues. An archive of information on these meetings is available at `http://www.sbw-sbml.org/workshops/index.html`

7.5 What are the “Workshops on Software Platforms for Systems Biology”?

They are the same as the SBML forum meetings (see Question 7.4).

7.6 SBML doesn’t encode all the information that is being used in systems biology modeling. What is being done to address this problem?

The SBML community has established *Working Groups* to develop proposals for extending SBML in future levels. The current Working Groups are:

The model composition group Investigating SBML extensions to support the composition of models from submodels.

The diagrams/graphical layout group Investigating SBML extensions to support the recording of the graphical network diagrams of a model created by many contemporary network-oriented modeling tools.

The arrays group Investigating the extension of SBML data structures to permit arrays of items (e.g., species, compartments) to be grouped and manipulated *en masse*.

The complexes/multistate species group Investigating SBML extensions to enable the compact representation of species with multiple possible “states” (e.g., due to phosphorylation).

The parameter sets group Investigating the creation of a separate standard for defining alternative initial values for variables and parameters in a model.

The alternative reactions group Investigating the extension of SBML reactions to support more directly the expression of stochastic and other formalism for describing reactions.

The hybrid models group Investigating how SBML should support multiple formalisms, such as continuous and discrete deterministic within the same model.

The controlled vocabularies group Investigating the extension of SBML to indicate that a model, its reactions, or other components of a model are meant to be interpreted in certain ways (e.g., to be simulated in a stochastic framework).

The dynamic structures group Investigating the extension of SBML to enable model structures to vary during simulation. One aspect of aspect of this is allowing rules and reactions to have effects that depend on the state of the model system.

The spatial features group Investigating the extension of SBML to describe the 2-D and 3-D spatial characteristics of models, the geometry of compartments, the diffusion properties of species, and the specification of different species concentrations across different regions of a cell.

7.7 How do members of the working groups communicate?

Discussions within the group should, at least initially, take place on sbml-discuss, rather than on a separate mailing list. This will avoid the need for working group organizer(s) to create new archived mailing lists right from the start, and will also allow other people to see some of the discussions even if they do not want to be involved to the level of officially being in the working group. If the traffic for a particular group becomes too much for other people on sbml-discuss, that working group may create its own archived mailing list at that time.

7.8 I have identified a feature or features that are missing from SBML. How to do I start a working group to address this issue?

Any interested person or group can send mail to sbml-discuss to propose forming a working group. The announcement should include the following information:

- A short statement of the purpose or goal of the effort (typically, an extension to SBML).
- The initial contact person for the group. That person may or may not be the eventual chairperson.
- The list of the initial members of the group.
- A description of the expected form of the outcome from the effort. This will typically be a document describing some proposed changes to SBML.
- An estimate of how long the group expects to stay in existence (meaning, how long it will give itself to accomplish its goals before reevaluating its status and expectations.)

Being a member of a working group implies a desire and willingness to work on actual software underlying the focus of the group's effort. Proposals for SBML changes need to be accompanied by examples of one or more software implementations of the proposed changes. To make this possible, the implementations should add proposed XML extensions within the SBML `annotation` element. This will allow tools to exchange models with the proposed extensions in a semi-organized fashion. Once the proposed changes have been debugged to the satisfaction of the group(s) using them, the proposal can be introduced to the SBML community at large. When accepted, the new extensions will be migrated out of the 'annotation' element and made a part of SBML.

7.9 What is the relationship between SBML and CellML?

CellML is built around an approach of composing systems of equations by linking together the variables in those equations; this is augmented by features for declaring biochemical reactions explicitly, as well as encapsulating arbitrary components into modules. By contrast, SBML provides constructs that are more similar to the internal object models used in many simulation/analysis packages specialized for biochemical networks. This reflects its history of having been developed in cooperation with software developers who write these tools. In practice, SBML appears to be better suited to the purpose of enabling interoperability with existing simulation tools focused on the biochemical reaction level. These differences notwithstanding, the SBML and CellML efforts share much in common, and the development of SBML has benefited from discussions and interactions with the developers of CellML. (The CellML team has been an active participant in the SBML development process.) The editors view CellML and SBML as somewhat different approaches being investigated as solutions to the same general problems. One of the aims of the model composition working group is to ensure that the two representations can be integrated with each other to create a single standard in the future. The result will be compositional compatibility between CellML and SBML, such that models expressed in one language can be used as components or submodels in the other.

7.10 What is the relationship between SBML and BioPAX?

BioPax is a consortium that is developing a format for the exchange of pathway data between bioinformatics databases. The aim is to ensure that large subsets of data represented in one format can be converted into the other format. The BioPAX standard is designed to support a large set of biochemical entity types and types of relationships between these types. These types will be arranged into ontologies. The SBML editors have a good working relationship with the group doing the majority of the work on this standard: Chris Sander's group at MSKCC.

7.11 What is the relationship between SBML and I3C?

Interoperable Informatics Infrastructure Consortium (I3C) is an organization which promotes the use of standards in the life sciences research software. The I3C has a pathways/systems biology working group. The SBML editors are co-chairs of this working group.

The I3C doesn't create standards; instead, it encourages their development and then makes recommendations on standards to its members. I3C has a close relationship with OMG.

7.12 Why isn't SBML being developed under the auspices of a standards body like the OMG?

We are attempting to have the SBML development process meet the requirements of the community it serves.

The editors considered submitting SBML as a proposal to the Object Management Group (OMG) in response to a request for proposals (RFP) for pathways representations. However, the SBML community decided at the 7th Forum meeting in May, 2003, that while it would be useful to have the endorsement of a standards body like the OMG, our time would be better spent working on standards development rather conforming to all the standards requirements of the OMG process. The editors did not submit a response to the OMG RFP. However, Scott Markel of LION Bioscience and chair of the OMG LSR group, submitted an outline proposal suggesting SBML and CellML as a basis for an OMG standard as a response to this RFP.

7.13 Which research groups are involved in SBML development?

All the groups that have produced the software listed in Question 2 are involved to some extent in the development of SBML.

8 History

8.1 Who proposed the creation of SBML?

Dr. Hamid Bolouri initially suggested it at the First Workshop on Software Platforms for Systems Biology in April 2000. The delegates there supported the idea enthusiastically.

8.2 Who were the original SBML editors?

Michael Hucka, Herbert Sauro and Andrew Finney wrote the original SBML proposal in the second half of the year 2000.

8.3 What was the organizational context of the original editors?

This work was initially performed in John Doyle's group by Michael Hucka, Herbert Sauro, Andrew Finney and Hamid Bolouri in the Control and Dynamical Systems Department (CDS) at the California Institute of Technology (Caltech). John Doyle and Hiroaki Kitano were the joint Principle Investigators. Hamid Bolouri managed the process until May 2003.

8.4 Where are they now?

Hamid Bolouri is Professor of Computational Biology at the Institute of Technology. Herbert Sauro is an Assistant Professor at the Keck Graduate Institute. Mike Hucka remains in CDS and Caltech. Andrew Finney is a Senior Research Fellow at the Science and Technology Research Centre at the University of Hertfordshire. Mike and Andrew are now joint chairs/editors of the SBML process.

8.5 Who funded the initial editorial work?

The initial development of SBML was funded by the Japan Science And Technology Corporation's *Exploratory Research for Advanced Technology* program (JST ERATO).

8.6 Who funded many of the SBML Forum meetings?

UK's Biotechnology and Biological Sciences Research Council.

9 Help

9.1 My question is not answered in this FAQ list. Who should I contact?

You may contact any of the editors of this FAQ individually, or use the SBML project email address, sbml-team@caltech.edu. The editors of this FAQ are:

- Andrew Finney
- Mike Hucka