A tutorial about SBML and

SBML Level 2 Version 2

Michael Hucka

Biological Network Modeling Center,
Beckman Institute
California Institute of Technology
Pasadena, California, USA

Sarah Keating

Science & Technology Research Institute
University of Hertfordshire
Hatfield, Hertfordshire, UK

Tutorial outline

1. Some background about SBML
2. SBML language basics
3. LibSBML and other SBML software infrastructure
4. Additional SBML features and SBML Level 2 Version 2 differences
5. A brief survey of SBML-compatible software
6. Closing comments and discussions

The background of SBML

Computational modeling is essential

Computational modeling is essential

- Computational modeling is increasingly crucial to biology

Computational modeling is essential

- Computational modeling is increasingly crucial to biology
- Not a new idea!
  - Experiencing heightened interest thanks to systems biology

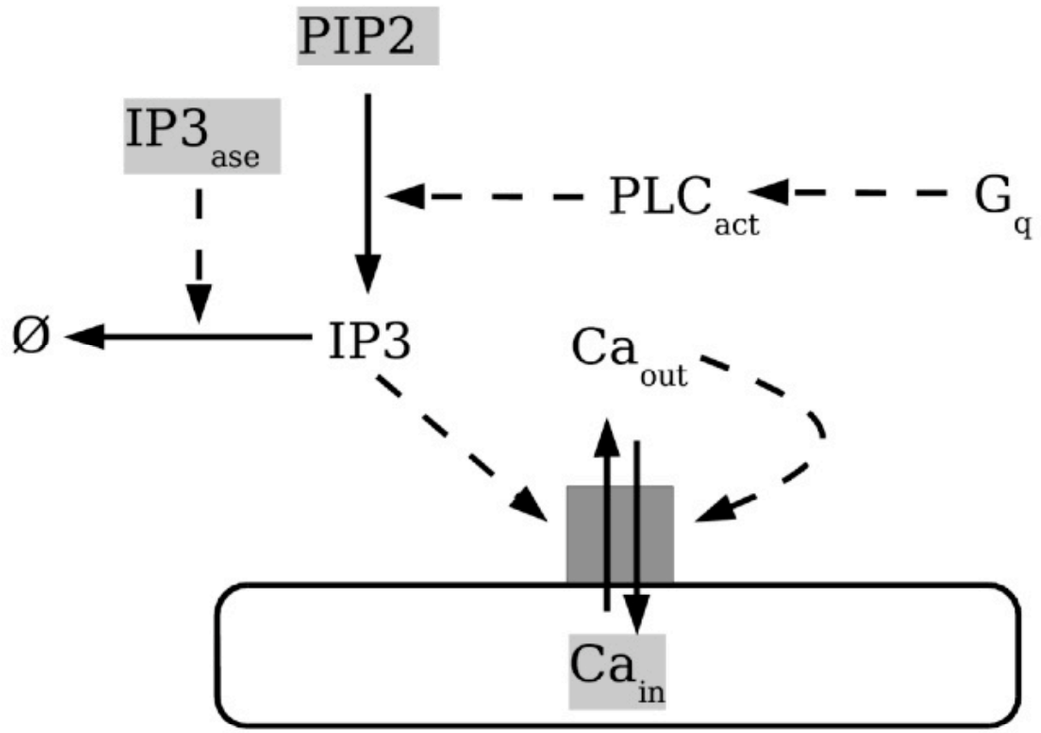
Computational modeling is essential

- Computational modeling is increasingly crucial to biology
- Not a new idea!
  - Experiencing heightened interest thanks to systems biology
- Serves the scientific method
  - Enables quantitative hypothesis testing
    - Forces you to quantify every assumption & make it testable
  - Enables precise knowledge transfer
    - Reduces ambiguities

Computational modeling is essential

- Computational modeling is increasingly crucial to biology
- Not a new idea!
  - Experiencing heightened interest thanks to systems biology
- Serves the scientific method
  - Enables quantitative hypothesis testing
    - Forces you to quantify every assumption & make it testable
  - Enables precise knowledge transfer
    - Reduces ambiguities
- Sign of rising popularity: new journals starting up
  - E.g., *PLoS Computational Biology*

Computational models



$$k_1 = k_2 = k_3 = 1\ s^{-1}$$

$$Km_1 = 10^{-7}\ M, Km_2 = 10^{-8}, Km_3 = 2.10^{-6}\ M$$

$$K_A = 10^{-11}, m = 4, n = 3, \alpha = 0.001$$

$$[Ca_{in}] = [IP3R] = [PLC_{tot}] = [PIP2] = [IP3_{ase}] = 0.001\ M$$

$$[G_q] = 0.01\ M, [Ca_{out}] = [IP3] = [PLC_{act}] = 0\ M$$

$$\frac{d[Ca_{out}]}{dt} = \frac{k_1[IP3R] * ([Ca_{in}] - [Ca_{out}])}{Km_1 + |[Ca_{in}] - [Ca_{out}]|} * \frac{[IP3]^m}{K_A + [IP3]^m}$$

$$\frac{d[IP3]}{dt} = \frac{k_2[PLC_{act}] * [PIP2]}{Km_2 + [PIP2]} - \frac{k_3[IP3_{ase}] * [IP3]}{Km_3 + [IP3]}$$

$$\frac{d[PLC_{act}]}{dt} = \frac{[G_q]^n}{\alpha + [G_q]^n} * [PLC_{tot}]$$

Software tools to aid modeling

Software tools to aid modeling

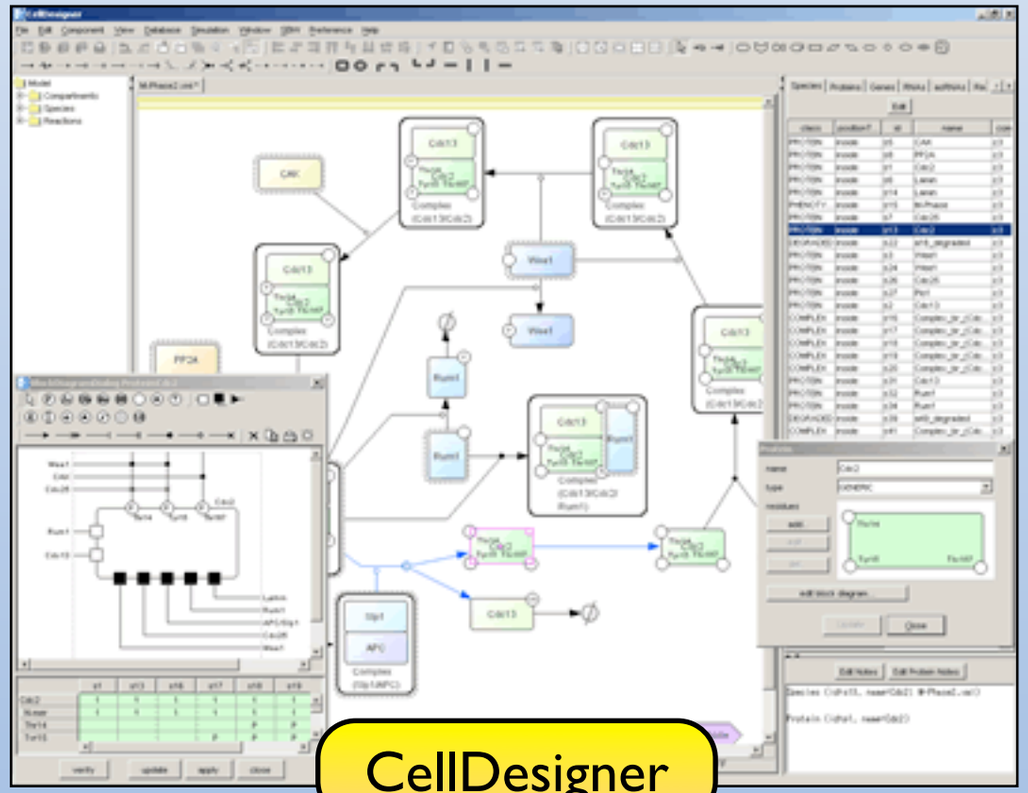
- General-purpose environments
  - Mathematica, MATLAB, etc.

Software tools to aid modeling

- General-purpose environments
  - Mathematica, MATLAB, etc.
- Special-purpose software
  - model editing
  - simulation
  - analysis
  - visualization

Software tools to aid modeling

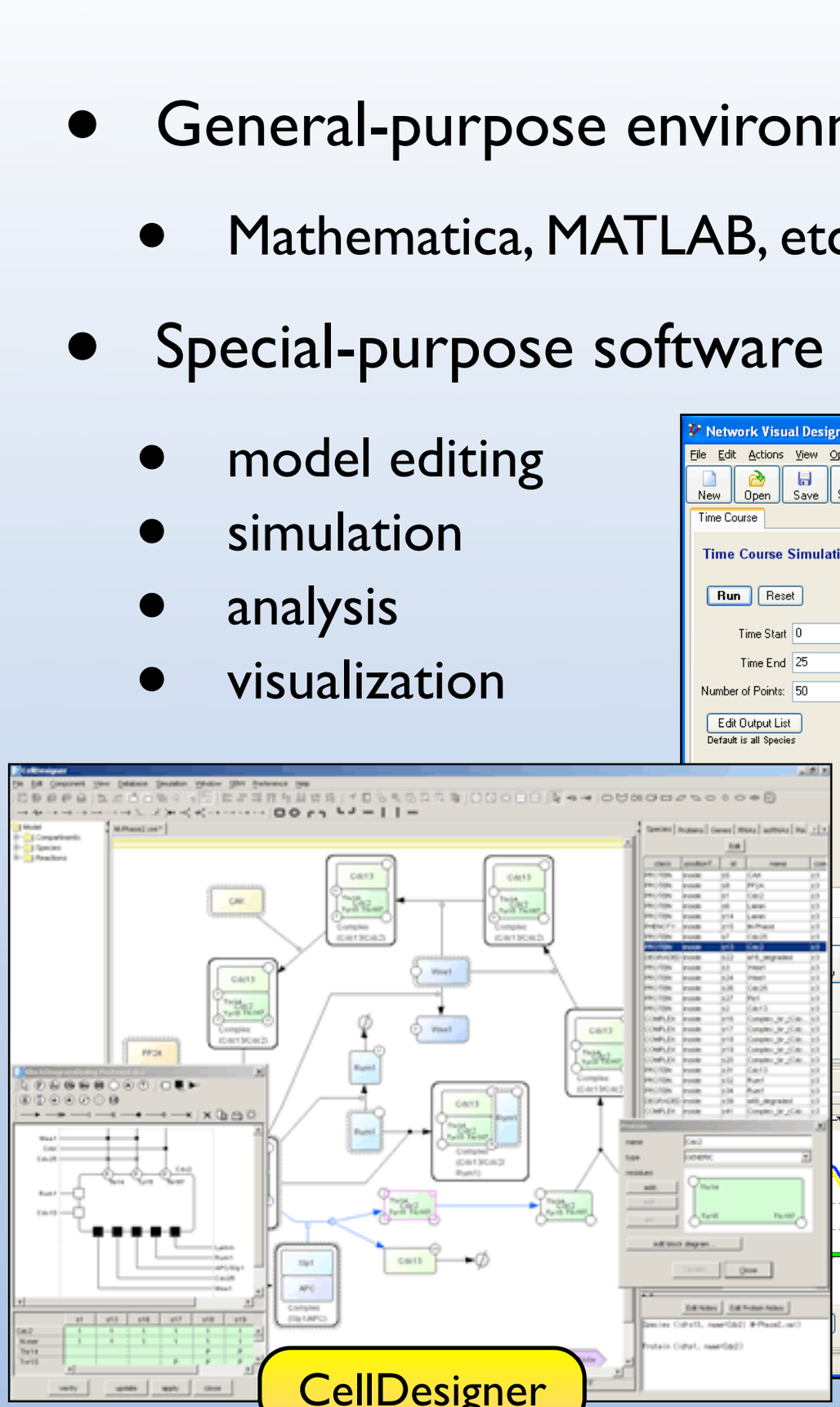
- General-purpose environments
  - Mathematica, MATLAB, etc.
- Special-purpose software
  - model editing
  - simulation
  - analysis
  - visualization

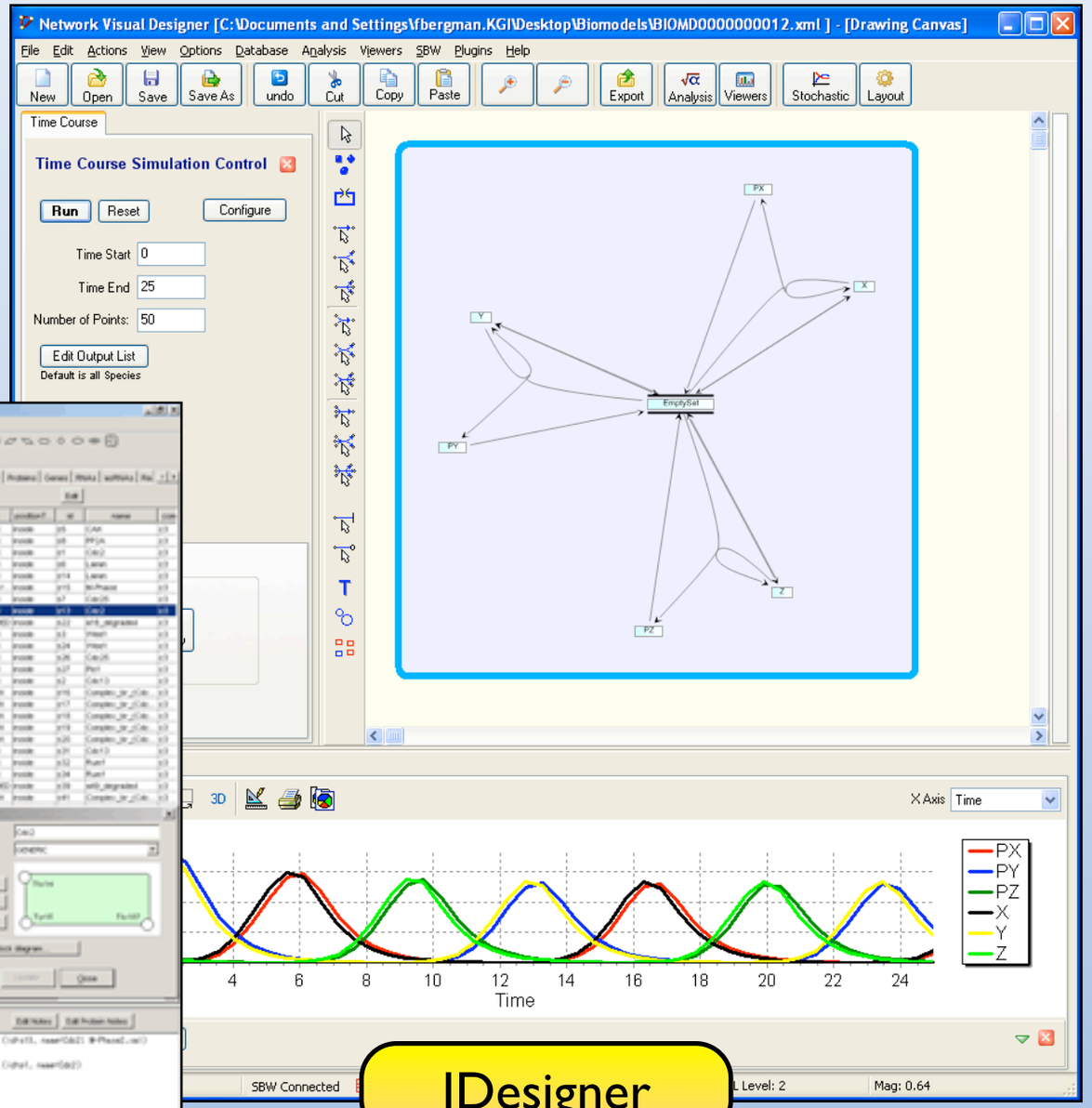


CellDesigner

Software tools to aid modeling

- General-purpose environments
  - Mathematica, MATLAB, etc.
- Special-purpose software
  - model editing
  - simulation
  - analysis
  - visualization

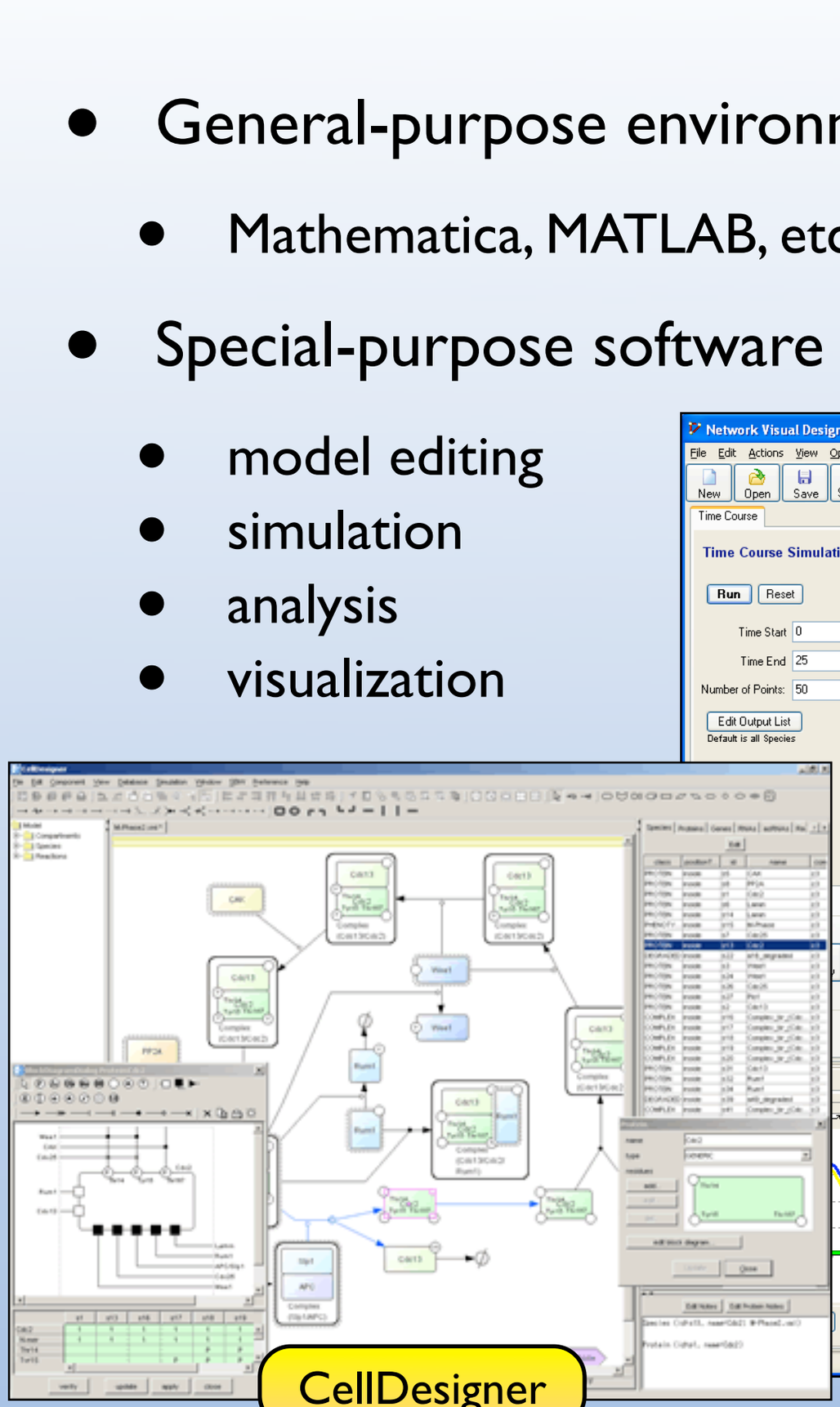


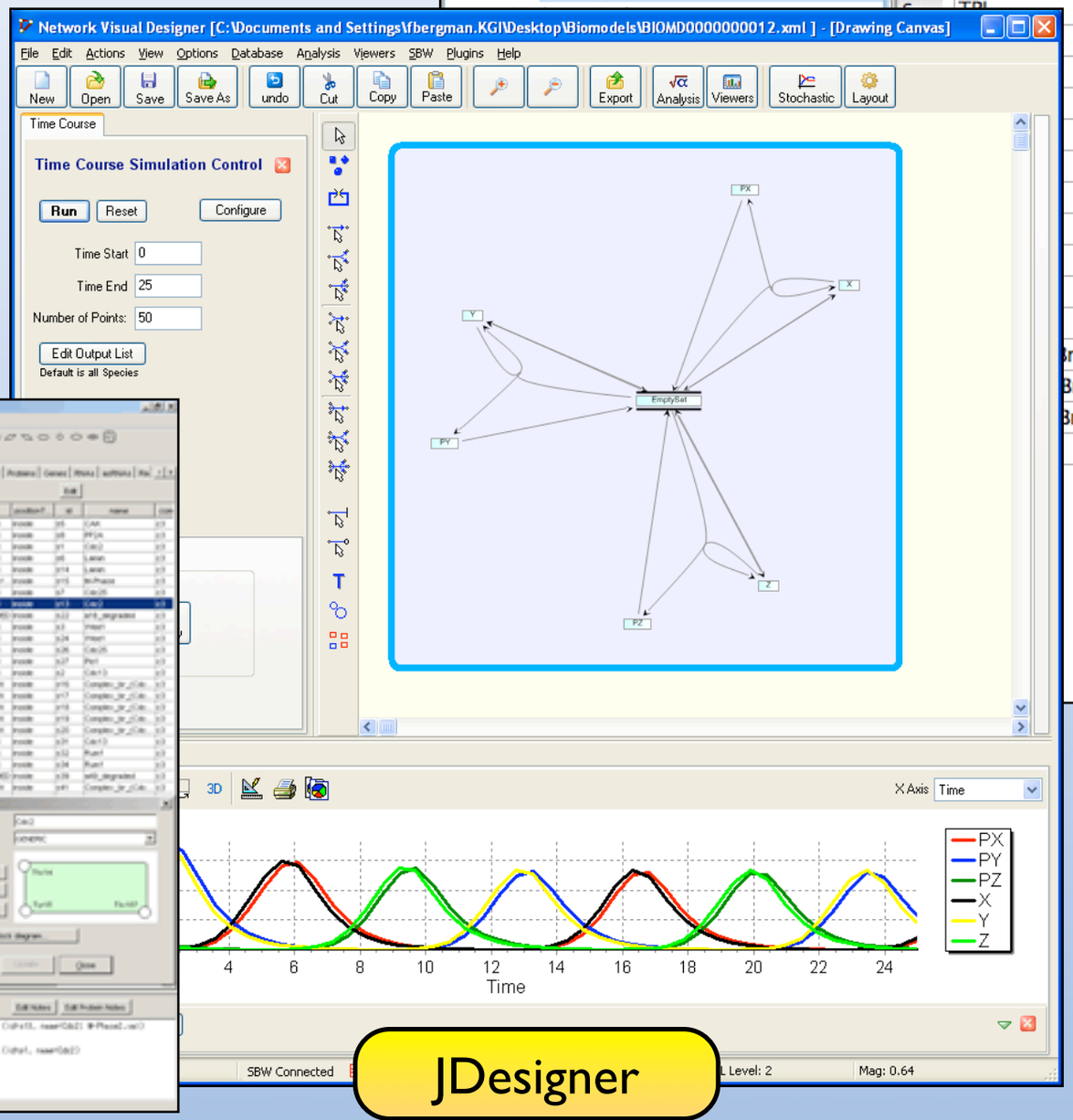


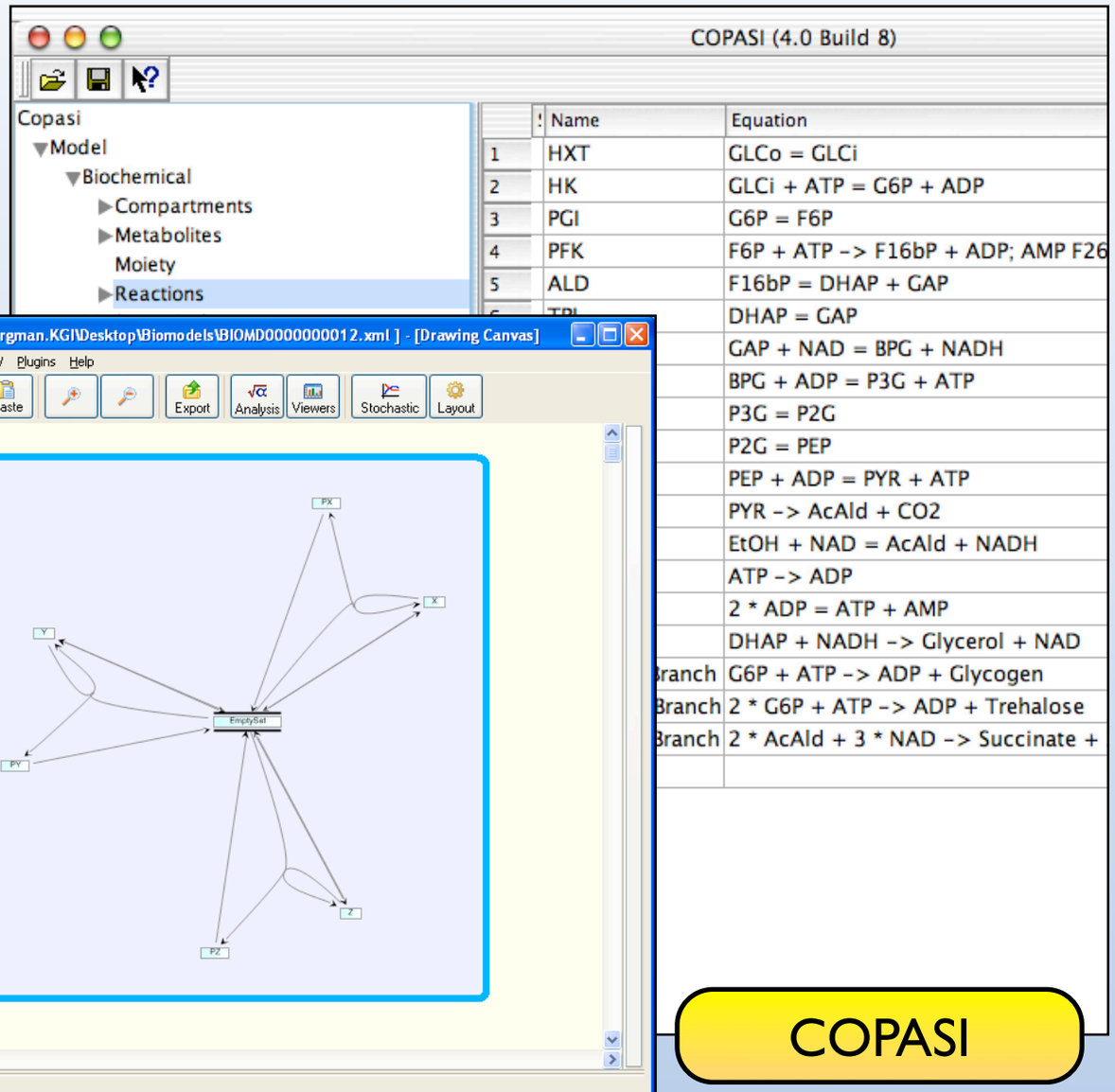
JDesigner

Software tools to aid modeling

- General-purpose environments
  - Mathematica, MATLAB, etc.
- Special-purpose software
  - model editing
  - simulation
  - analysis
  - visualization







Ability to exchange models is critical

Ability to exchange models is critical

- Simply publishing equations is not enough
  - Don’t want to transcribe equations from papers
  - Want a common file format

Ability to exchange models is critical

- Simply publishing equations is not enough
  - Don't want to transcribe equations from papers
  - Want a common file format
- Not an earth-shattering idea!
  - But curiously, such a format hadn't existed before year 2000
    - Each tool had its own unique proprietary format

SBML = Systems Biology Markup Language

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
- Suitable for reaction networks
  - Arbitrary rate functions

$$2\,A + B \rightarrow C$$

$$C \rightleftharpoons D + F$$

...

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
- Suitable for reaction networks
  - Arbitrary rate functions
- Declarative, not procedural

$$2\,A + B \rightarrow C$$

$$C \rightleftharpoons D + F$$

…

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
- Suitable for reaction networks
  - Arbitrary rate functions
- Declarative, not procedural
- Models can also include
  - Compartments (i.e., where chemical substances are located)
  - Mathematical "extras" (assignments, explicit different eq's)
  - Discontinuous events with arbitrary triggers

$$2\,A + B \rightarrow C$$

$$C \rightleftharpoons D + F$$

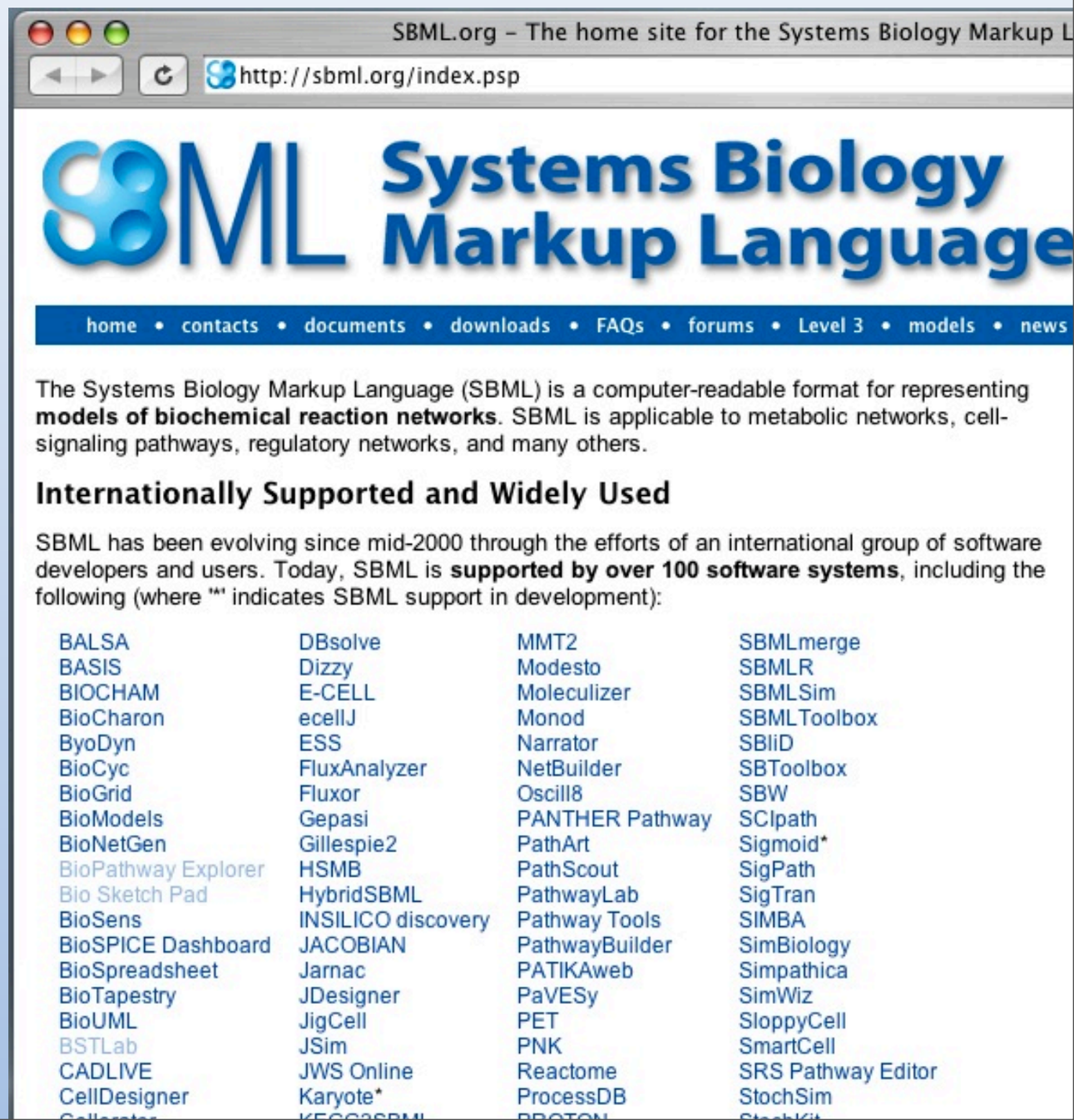
…

Fundamentally an exchange format

- For transferring models between software tools
- Not meant as a system's *internal* format
- Not suited for representing—
  - Experimental results
  - Numerical simulation results

Now the *de facto* standard

- Supported by >100 systems
  - Simulators, databases, analysis tools, editing tools
- Accepted by journals
  - Nature
  - PLoS Computational Biology
  - BMC
- Used in texbooks & courses

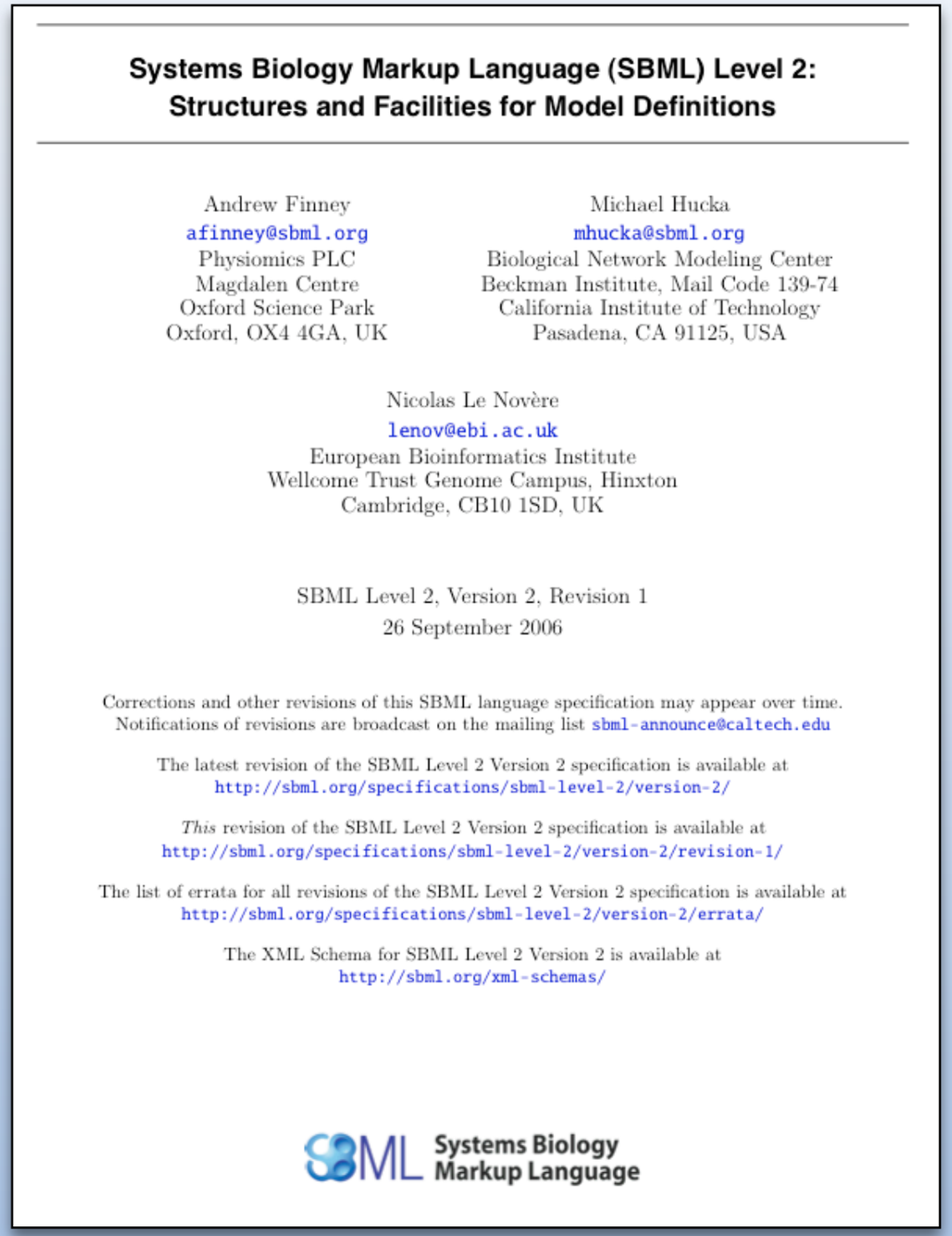


SBML "Levels"

- Levels are meant to coexist
- *Level 1*: mostly basic compartmental modeling
- *Level 2*: significantly more features—e.g.:
  - User-defined functions
  - Events
  - "Types" for chemical species and compartments
  - Initial conditions, constraints, other "fiddly bits"
- *Level 3*: now (back) in development

Latest: SBML Level 2 Version 2

- Final version released September 26
  - Embodies years of discussions and practical experience
  - Most software currently only supports L2V1, not L2V2 yet
    - But L2V2 support will come soon



Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions

Andrew Finney
afinney@sbml.org
Physiomics PLC
Magdalen Centre
Oxford Science Park
Oxford, OX4 4GA, UK

Michael Hucka
mhucka@sbml.org
Biological Network Modeling Center
Beckman Institute, Mail Code 139-74
California Institute of Technology
Pasadena, CA 91125, USA

Nicolas Le Novère
lenov@ebi.ac.uk
European Bioinformatics Institute
Wellcome Trust Genome Campus, Hinxton
Cambridge, CB10 1SD, UK

SBML Level 2, Version 2, Revision 1
26 September 2006

Corrections and other revisions of this SBML language specification may appear over time. Notifications of revisions are broadcast on the mailing list sbml-announce@caltech.edu

The latest revision of the SBML Level 2 Version 2 specification is available at
http://sbml.org/specifications/sbml-level-2/version-2/

This revision of the SBML Level 2 Version 2 specification is available at
http://sbml.org/specifications/sbml-level-2/version-2/revision-1/

The list of errata for all revisions of the SBML Level 2 Version 2 specification is available at
http://sbml.org/specifications/sbml-level-2/version-2/errata/

The XML Schema for SBML Level 2 Version 2 is available at
http://sbml.org/xml-schemas/

SBML Systems Biology Markup Language

A general overview of SBML

How is SBML used?

How is SBML used?

- Meant to provide an exchange language for software tools

How is SBML used?

- Meant to provide an exchange language for software tools
- Don't write SBML by hand if you can help it
  - (SBML is in XML, so you *could* write it by hand if you had to)

```
        <listOfReactants>
          <speciesReference species="MKKK"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="MKKK_P"/>
        </listOfProducts>
        <listOfModifiers>
          <modifierSpeciesReference species="MAPK_PP"/>
        </listOfModifiers>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <divide/>
              <apply>
                <times/>
                <ci> V1 </ci>
                <ci> MKKK </ci>
              </apply>
              <apply>
                <times/>
                <apply>
```

How is SBML used?

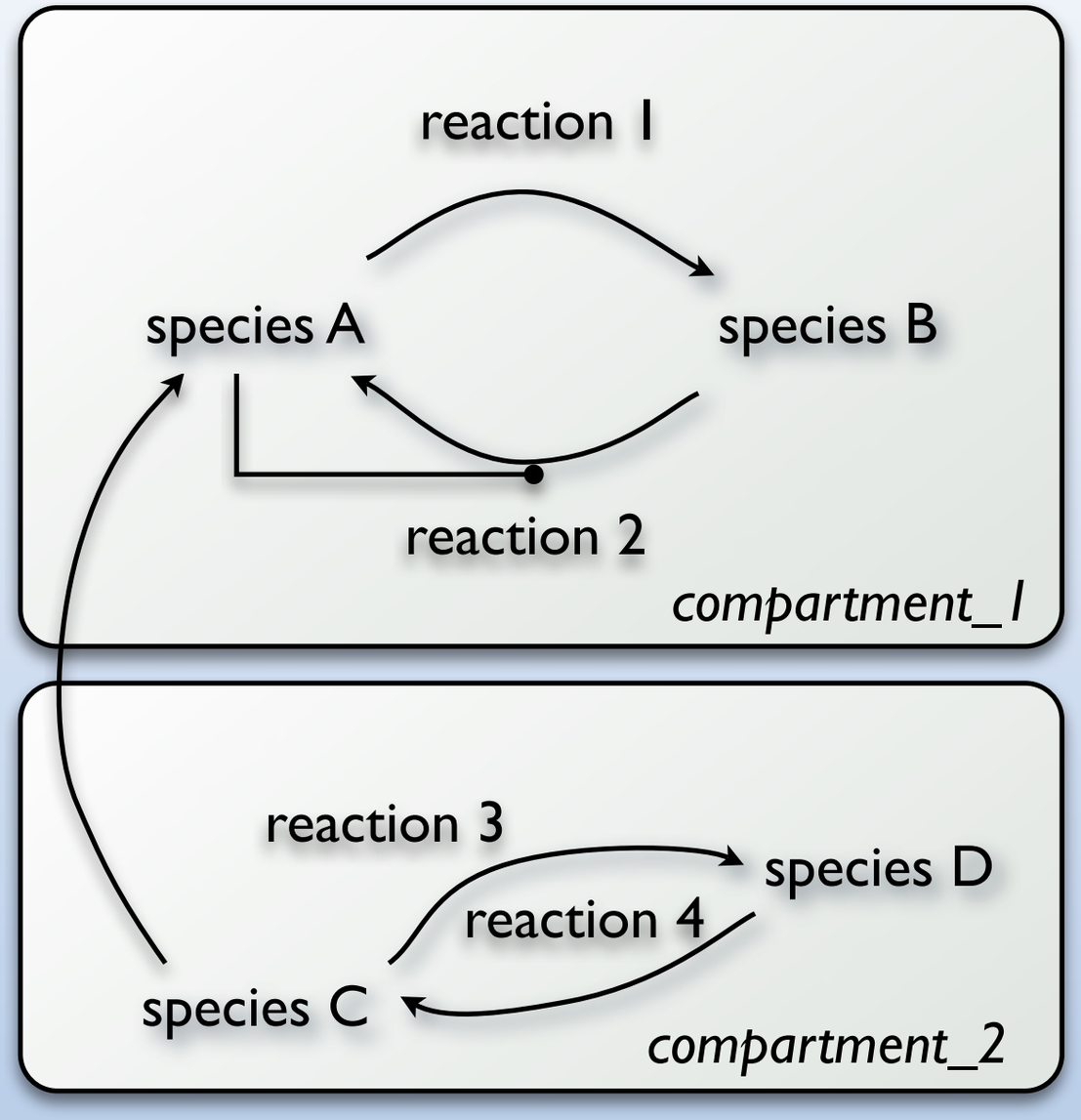
- Meant to provide an exchange language for software tools
- Don't write SBML by hand if you can help it
  - (SBML is in XML, so you *could* write it by hand if you had to)
- Software tools that "speak" SBML provide a higher-level interface

How is SBML used?

- Meant to provide an exchange language for software tools
- Don't write SBML by hand if you can help it
  - (SBML is in XML, so you *could* write it by hand if you had to)
- Software tools that "speak" SBML provide a higher-level interface
- Applications usually have their own native format
  - Import/export SBML rather than natively save as SBML

Some important SBML constructs

- Basic elements
  - species
  - compartment
  - parameter
  - reaction
- Additional useful elements
  - unit definition
  - “rule”
  - function definition
  - event
  - initial assignment
  - constraint



Example #1

- One reaction, 2A B + C, where rate is given as $k[A(t)]^2$
  - Initial conditions: [A(0)] = 3, [B(0)] = 0, [C(0)] = 0.

Basic SBML document structure

Sbml
level: positiveInteger { use="required" fixed="2" }
version: positiveInteger { use="required" fixed="2" }
model: Model

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version2"
      level="2" version="2">
   ...
</sbml>
```

- Format: plain text (technically UTF-8)
- Extension: usually .xml (not .sbml)

The **Model** container

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

Overall structure of the XML rendition of a **Model** instance

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

```
<model id="m" name="Example">
  <listOfFunctionDefinitions>
    ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
    ...
  </listOfUnitDefinitions>
  <listOfCompartmentTypes>
    ...
  </listOfCompartmentTypes>
  <listOfSpeciesTypes>
    ...
  </listOfSpeciesTypes>
  <listOfSpecies>
    ...
  </listOfSpecies>
    ...
```

Overall structure of the XML rendition of a **Model** instance

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

```
<model id="m" name="Example">
  <listOfFunctionDefinitions>
    ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
    ...
  </listOfUnitDefinitions>
  <listOfCompartmentTypes>
    ...
  </listOfCompartmentTypes>
  <listOfSpeciesTypes>
    ...
  </listOfSpeciesTypes>
  <listOfSpecies>
    ...
  </listOfSpecies>
    ...
```

Overall structure of the XML rendition of a **Model** instance

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

```
<model id="m" name="Example">
  <listOfFunctionDefinitions>
    ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
    ...
  </listOfUnitDefinitions>
  <listOfCompartmentTypes>
    ...
  </listOfCompartmentTypes>
  <listOfSpeciesTypes>
    ...
  </listOfSpeciesTypes>
  <listOfSpecies>
    ...
  </listOfSpecies>
    ...
```

Overall structure of the XML rendition of a **Model** instance

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

```
<model id="m" name="Example">
  <listOfFunctionDefinitions>
    ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
    ...
  </listOfUnitDefinitions>
  <listOfCompartmentTypes>
    ...
  </listOfCompartmentTypes>
  <listOfSpeciesTypes>
    ...
  </listOfSpeciesTypes>
  <listOfSpecies>
    ...
  </listOfSpecies>
    ...
```

Overall structure of the XML rendition of a **Model** instance

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

```
<model id="m" name="Example">
  <listOfFunctionDefinitions>
    ...
  </listOfFunctionDefinitions>
  <listOfUnitDefinitions>
    ...
  </listOfUnitDefinitions>
  <listOfCompartmentTypes>
    ...
  </listOfCompartmentTypes>
  <listOfSpeciesTypes>
    ...
  </listOfSpeciesTypes>
  <listOfSpecies>
    ...
  </listOfSpecies>
    ...
```

order is significant

`listOf_____s`

- Lists like **`Parameter[0..*]`** in the definition are translated into

```
<listOfParameters>
  <parameter ... />
  <parameter ... />
  ...
</listOfParameters>
```

- **`listOf_____s`** are derived from ***SBase***
  - Therefore, can have **`metaid`**, **`<notes>`**, **`<annotation>`**

SBase

SBase
metaid: ID { use="optional" } notes: (any : { namespace="http://www.w3.org/1999/xhtml" }) { minOccurs="0" maxOccurs="1" } annotation: (any) { minOccurs="0" maxOccurs="1" }

- Abstract type
- Most object structures in SBML are derived from ***SBase***
- **`<notes>`** allows human-readable annotations to be added
  - Format is XHTML
- **`<annotation>`** allows machine-readable annotations to be added
  - Applications can put their own data into it—just needs to be XML
  - Guidelines are discussed later in this tutorial
- **`metaid`** is for references by annotations

Common feature: identifiers and names

- Most elements have both an **`id`** and a **`name`** field
  - Identifier field has restricted syntax: abc123 or _abc123 or a_b_c_1 etc.
    - The **`id`** is what you use in expressions

Identifier	*Meaning*
species id	quantity of the species
compartment id	size of the compartment
parameter id	numerical value
function id	a call to that function
reaction id (L2v2)	rate of the reaction

  - Value of **`name`** is unrestricted (exception: no newlines or carriage returns)
  - Must assign a value to **`id`** on most objects, but **`name`** is always optional
- Some tools let you use **`name`**s & auto-generate **`id`**'s (e.g., COPASI)

Representation of mathematical expressions in SBML

- SBML Level 1: mathematical expressions encoded as text strings
  - In the XML, have **`formula="2*S1"`**
- SBML Level 2: math expressions encoded using MathML 2.0
  - Standard XML format for encoding mathematical expressions
  - SBML uses only a subset of the *content* portion of MathML
- MathML content always must be placed in a **`<math>`** element

MathML operators in SBML

- Most common operators available:
  - **`plus`**, **`minus`**, **`power`**, **`exp`**, etc.
  - relational operators: **`eq`**, **`neq`**, **`gt`**, **`lt`**, **`geq`**, **`leq`**
  - **`piecewise`**
  - **`pi, exponentiale`**
  - many others
- Complete list on p. 21 of L2V2 specification document

References in MathML

- References to numbers are through the **<cn>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn type="integer"> 42 </cn>
    <cn type="real"> 3.3 </cn>
  </apply>
</math>
```

- References to identifiers are through the **<ci>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn> 0.014 </cn>
    <ci> S1 </ci>
  </apply>
</math>
```

References in MathML

- References to numbers are through the **<cn>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn type="integer"> 42 </cn>
    <cn type="real"> 3.3 </cn>
  </apply>
</math>
```

- References to identifiers are through the **<ci>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn> 0.014 </cn>
    <ci> S1 </ci>
  </apply>
</math>
```

References in MathML

- References to numbers are through the **<cn>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn type="integer"> 42 </cn>
    <cn type="real"> 3.3 </cn>
  </apply>
</math>
```

- References to identifiers are through the **<ci>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn> 0.014 </cn>
    <ci> S1 </ci>
  </apply>
</math>
```

References in MathML

- References to numbers are through the **<cn>** element:

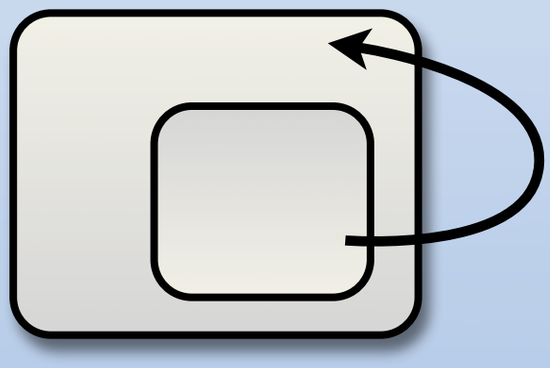
```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn type="integer"> 42 </cn>
    <cn type="real"> 3.3 </cn>
  </apply>
</math>
```

- References to identifiers are through the **<ci>** element:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <times/>
    <cn> 0.014 </cn>
    <ci> S1 </ci>
  </apply>
</math>
```

Compartment basics

- **`id & name`**
- **`spatialDimensions`**
  - `3, 2, 1, 0` (default: 3)
- **`size`**
  - Floating-point number giving the compartment size (volume or other)
- **`units`**
  - Identifier of the units of measure for the size
- **`outside`**
  - Identifier of the compartment outside of this one



- **`constant`**
  - Boolean: is the compartment size constant? (Default: true)

Compartment example

```
...
<listOfCompartments>
  <compartment id="cytoplasm" size="5"/>
  <compartment id="nucleus" size="1"
               outside="cytoplasm" />
</listOfCompartments>
...
<listOfSpecies>
  <species id="X" compartment="nucleus"
           initialAmount="1" />
  <species id="Y" compartment="cytoplasm"
           initialAmount="1" />
</listOfSpecies>
...
```

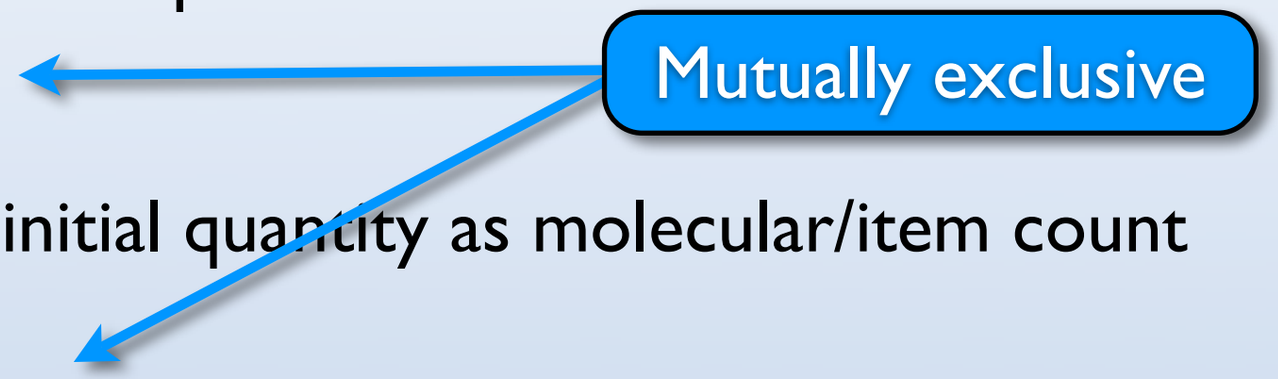
Species basics

Species basics

- **`id & name`**
- **`compartment`**
  - Identifier of compartment where species is located
- **`initialAmount`**
  - Floating-point number giving initial quantity as molecular/item count
- **`initialConcentration`**
  - Floating-point number giving initial quantity as concentration
    - More precisely: (*units of substance*)/(*units of size*)

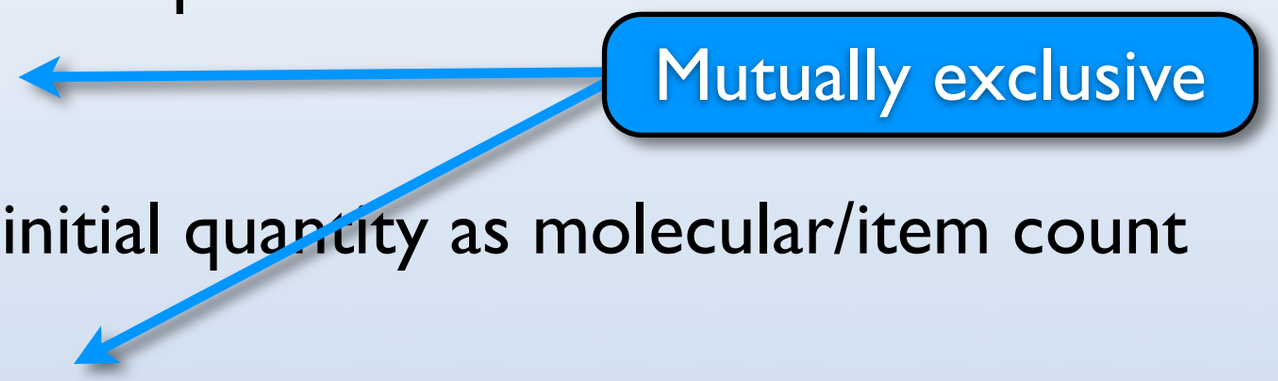
Species basics

- **`id & name`**
- **`compartment`**
  - Identifier of compartment where species is located
- **`initialAmount`**
  - Floating-point number giving initial quantity as molecular/item count
- **`initialConcentration`**
  - Floating-point number giving initial quantity as concentration
    - More precisely: (*units of substance*)/(*units of size*)



Species basics

- **`id & name`**
- **`compartment`**
  - Identifier of compartment where species is located
- **`initialAmount`**
  - Floating-point number giving initial quantity as molecular/item count
- **`initialConcentration`**
  - Floating-point number giving initial quantity as concentration
    - More precisely: (*units of substance*)/(*units of size*)
- **`boundaryCondition`**
  - Boolean: should a rate of change equation be constructed for the species based on the system of reactions? (Default: false)
- **`constant`**
  - Boolean: is the species quantity constant? (Default: false)



More about species

- Think of species as a pool of molecules of the same type
- Species must be located in *some* compartment
  - If have the *same* species in *multiple* compartments, must have *separate* species definitions for each
    - ... but in L2V2, can use **SpeciesType** to relate them together
- If doing stochastic model, best define species in terms of amounts

Species example

```
<listOfSpecies>
  <species id="S1" initialConcentration="0"
       compartment="c1" boundaryCondition="true" />
  <species id="S3" initialConcentration="10"
       compartment="c2" constant="true"/>
  <species id="S4" initialConcentration="4.5"
       compartment="c2"/>
</listOfSpecies>
```

Parameter basics

- **`id` & `name`**
- **`value`**
  - The floating-point value of the parameter
- **`units`**
  - Identifier of the units of measure for the value
- **`constant`**
  - Boolean: is the value of the parameter constant? (Default: true)

Parameter example

```
<listOfParameters>
  <parameter id="k1" value="0.5"/>
  <parameter id="k2" value="0.1"/>
</listOfParameters>
```

Reaction basics

} *References to defined species*

Reaction basics

- **`id` & `name`**
- **`<listOfReactants>`** (optional)
- **`<listOfProducts>`** (optional)
- **`<listOfModifiers>`** (optional)

} *References to defined species*

- **`<kineticLaw>`** (optional)

Reaction basics

- **`id` & `name`**
- **`<listOfReactants>`** (optional)
- **`<listOfProducts>`** (optional)
- **`<listOfModifiers>`** (optional)

} *References to defined species*

- **`<kineticLaw>`** (optional)

- **`reversible`**
  - Boolean: is the reaction reversible? (Default: true)

About reactants, products and modifiers

About reactants, products and modifiers

- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions

About reactants, products and modifiers

- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions
- Reaction can have any number of reactants or products, but *must* have at least one reactant *or* product

About reactants, products and modifiers

- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions
- Reaction can have any number of reactants or products, but *must* have at least one reactant *or* product
- A species can appear in both the list of reactants & list of products
  - Effective stoichiometry is then: (stoich.-as-react.) - (stoich.-as-prod.)
    - E.g.: 2A A + B ==> effective stoichiometry of A is +1

About reactants, products and modifiers

- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions
- Reaction can have any number of reactants or products, but *must* have at least one reactant *or* product
- A species can appear in both the list of reactants & list of products
  - Effective stoichiometry is then: (stoich.-as-react.) - (stoich.-as-prod.)
    - E.g.: 2A A + B ==> effective stoichiometry of A is +1
- A species labeled as modifier may also appear in list of reactants and/or products—not an either/or situation

About reactants, products and modifiers

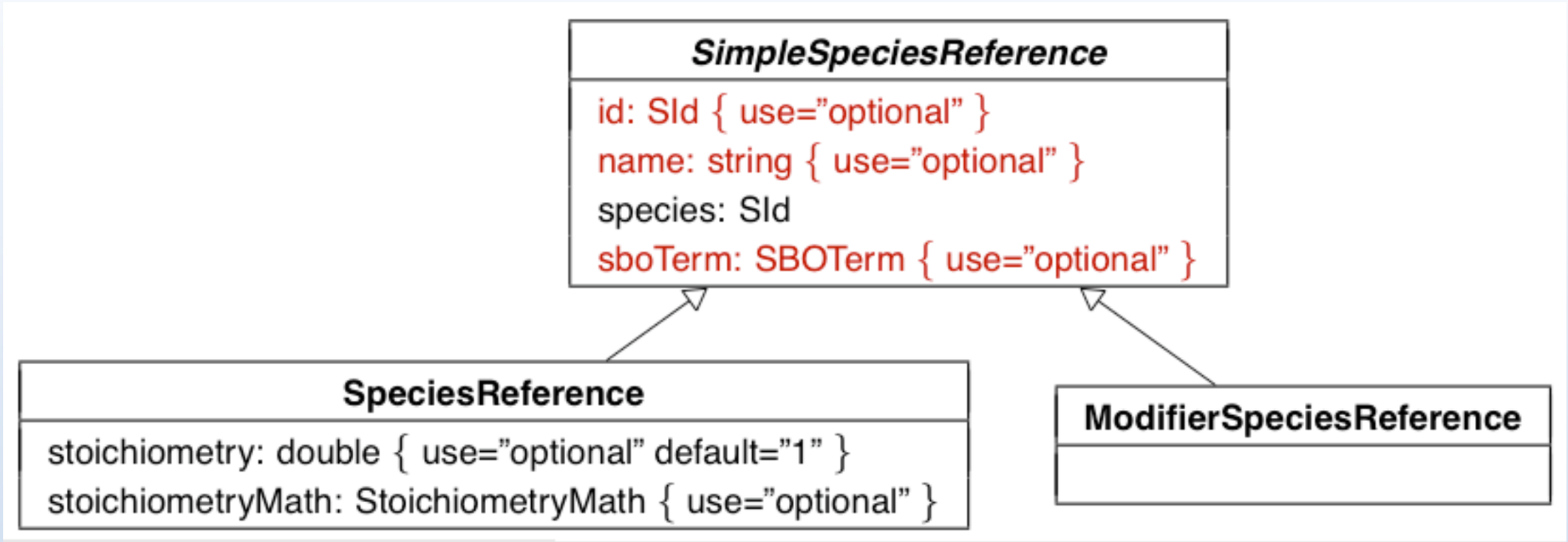
- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions
- Reaction can have any number of reactants or products, but *must* have at least one reactant *or* product
- A species can appear in both the list of reactants & list of products
  - Effective stoichiometry is then: (stoich.-as-react.) - (stoich.-as-prod.)
    - E.g.: 2A A + B ==> effective stoichiometry of A is +1
- A species labeled as modifier may also appear in list of reactants and/or products—not an either/or situation
- A "modifier" species appears in the rate expression but is neither created nor destroyed in that reaction

About reactants, products and modifiers

- All species must be defined in the model's top-level list of species
  - Reactions simply refer back to the species definitions
- Reaction can have any number of reactants or products, but *must* have at least one reactant *or* product
- A species can appear in both the list of reactants & list of products
  - Effective stoichiometry is then: (stoich.-as-react.) - (stoich.-as-prod.)
    - E.g.: 2A A + B ==> effective stoichiometry of A is +1
- A species labeled as modifier may also appear in list of reactants and/or products—not an either/or situation
- A "modifier" species appears in the rate expression but is neither created nor destroyed in that reaction

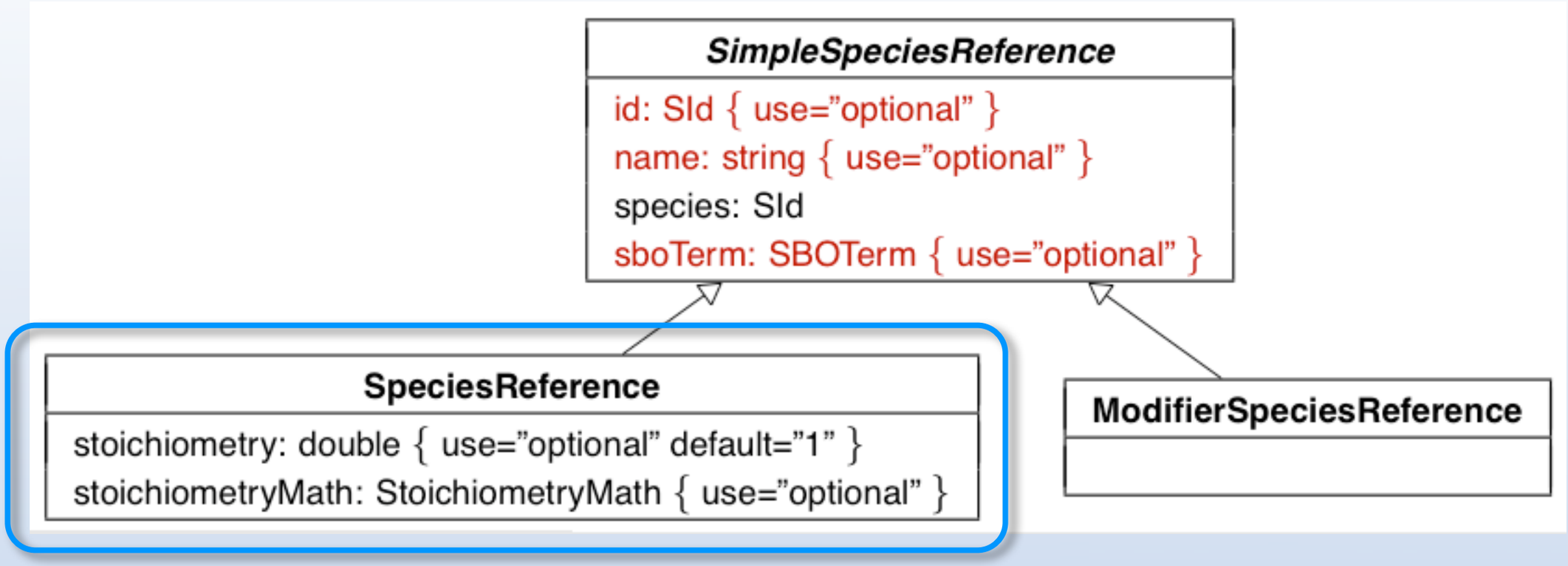
no specific role assumed for modifiers

Lists of reactants, products and modifiers



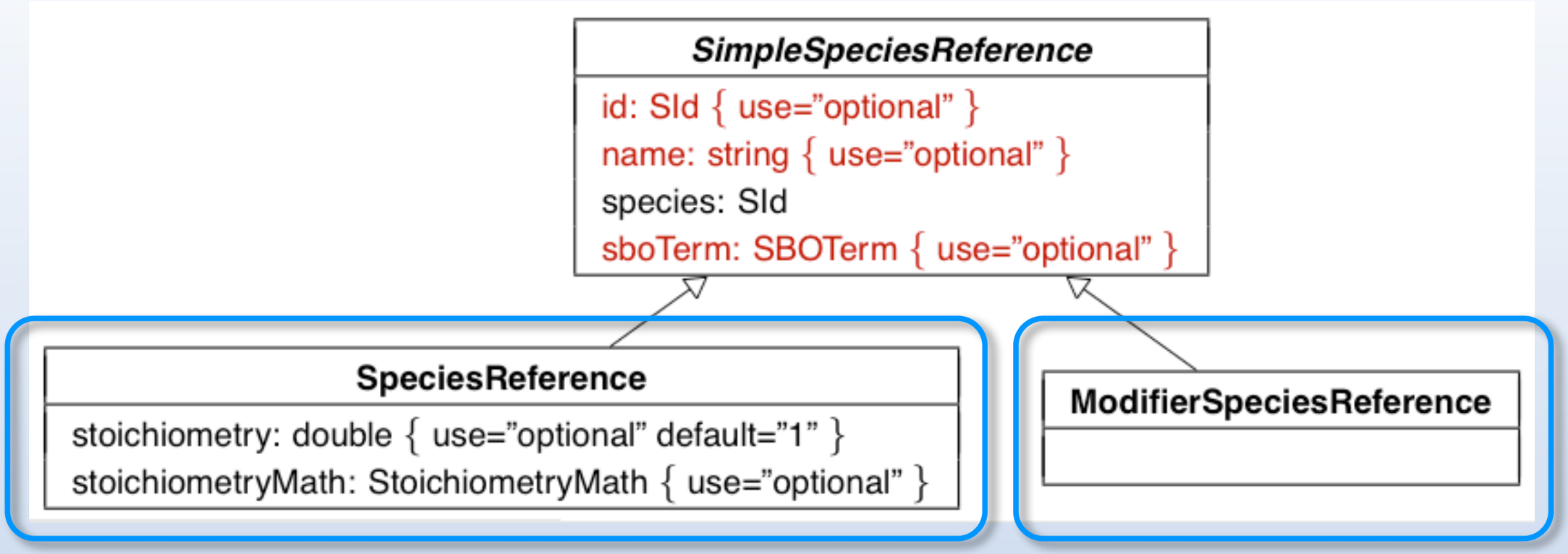
- Most import common field: **`species`**
  - **`species`** value must be id of existing species defined in the model
- **ModifierSpeciesReference** does not add any more fields
- **SpeciesReference** adds fields for stoichiometry

Lists of reactants, products and modifiers



- Most import common field: **`species`**
  - **`species`** value must be id of existing species defined in the model
- **ModifierSpeciesReference** does not add any more fields
- **SpeciesReference** adds fields for stoichiometry

Lists of reactants, products and modifiers



- Most import common field: **`species`**
  - **`species`** value must be id of existing species defined in the model
- **ModifierSpeciesReference** does not add any more fields
- **SpeciesReference** adds fields for stoichiometry

Stoichiometries

- Normally a stoichiometry is an integer or floating-point scalar value
  - Default value is "**1**"

```
<reaction id="Dimerization" reversible="false">
  <listOfReactants>
    <speciesReference species="P" stoichiometry="2"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="P2" />
  </listOfProducts>
  ...
```

- For more complicated stoichiometries, use **`stoichiometryMath`**
  - MathML expression
  - Mutually exclusive with **`stoichiometry`**, use one or the other

Stoichiometries

- Normally a stoichiometry is an integer or floating-point scalar value
  - Default value is "**1**"

```
<reaction id="Dimerization" reversible="false">
  <listOfReactants>
    <speciesReference species="P" stoichiometry="2"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="P2" />
  </listOfProducts>
  ...
```

- For more complicated stoichiometries, use **`stoichiometryMath`**
  - MathML expression
  - Mutually exclusive with **`stoichiometry`**, use one or the other

Stoichiometries

- Normally a stoichiometry is an integer or floating-point scalar value
  - Default value is "**1**"

```
<reaction id="Dimerization" reversible="false">
  <listOfReactants>
    <speciesReference species="P" stoichiometry="2"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="P2" />
  </listOfProducts>
  ...
```

- For more complicated stoichiometries, use **`stoichiometryMath`**
  - MathML expression
  - Mutually exclusive with **`stoichiometry`**, use one or the other

KineticLaw basics

- **`<math>`**
  - MathML expression for the speed of the reaction
  - Units must be substance/time
- **`<listOfParameters>`**
  - Defines parameters whose identifiers have scope local to the reaction *only*
    - Not visible from any other reaction, or rest of model
    - Identifiers shadow global identifiers
- Data type is same **Parameter** as for global parameters

KineticLaw basics

- **`<math>`**
  - MathML expression for the speed of the reaction
  - Units must be substance/time

not concentration/time

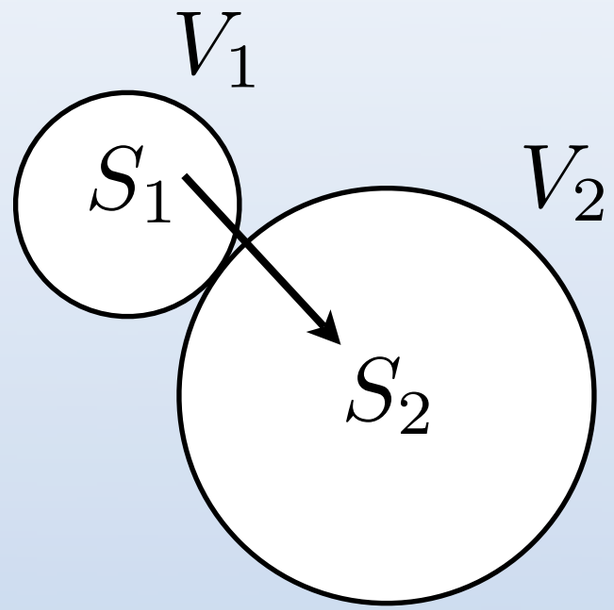
- **`<listOfParameters>`**
  - Defines parameters whose identifiers have scope local to the reaction *only*
    - Not visible from any other reaction, or rest of model
    - Identifiers shadow global identifiers
  - Data type is same **Parameter** as for global parameters

Interpreting reactions

- Why are SBML rate expression not identical to rate laws?
  - Consider simple example:

$$S_1 \rightarrow S_2$$

$$\text{rate law} = k \cdot [S_1]$$



  - What does this mean?

$$\frac{d[S_2]}{dt} = -\frac{d[S_1]}{dt} = k \cdot [S_1]$$

- But what if $V_1 \neq V_2$? For example, what if $V_2 = 3V_1$?
  - Look at number of molecules of each species in each compartment:

$$n_{S_1} = [S_1] \cdot V_1 \qquad n_{S_2} = [S_2] \cdot V_2$$

  - How molecules leave & enter each compartment?
    - $k \cdot [S_1] \cdot V_1$ molecules leave the first compartment
    - $3 \cdot k \cdot [S_1] \cdot V_1$ molecules enter the second compartment

"Kinetic law" in SBML

- Rate expressions are substance/time, **not** substance/size/time
- Conversion for basic cases is simple:
  - Multiply by volume of compartment where reactants are located:

$$\text{rate} = k \cdot [S_1] \cdot V_1$$

  - Express rates of changes of reactants & products in terms of substances:

$$\frac{dn_{S_1}}{dt} = -k_1 \cdot [S_1] \cdot V_1$$

$$\frac{dn_{S_2}}{dt} = k_1 \cdot [S_1] \cdot V_1$$

  - Can easily recover concentrations:

$$[S_1] = \frac{n_{S_1}}{V_1} \qquad [S_2] = \frac{n_{S_2}}{V_2}$$

"Kinetic law" in SBML

- Rate expressions are substance/time, **not** substance/size/time
- Conversion for basic cases is simple:
  - Multiply by volume of compartment where reactants are located:

$$\text{rate} = k \cdot [S_1] \cdot V_1$$

  - Express rates of changes of reactants & products in terms of substances:

$$\frac{dn_{S_1}}{dt} = -k_1 \cdot [S_1] \cdot V_1$$

$$\frac{dn_{S_2}}{dt} = k_1 \cdot [S_1] \cdot V_1$$

moles
items
mass
"dimensionless"

- Can easily recover concentrations:

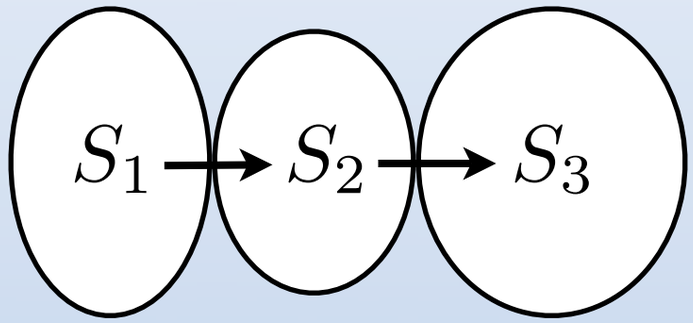
$$[S_1] = \frac{n_{S_1}}{V_1} \qquad [S_2] = \frac{n_{S_2}}{V_2}$$

Example #2

$$S_1 \rightarrow S_2 \rightarrow S_3$$

Rate laws: $r_1 = k_1 \cdot [S_1]$ $r_2 = k_2 \cdot [S_2]$

Compartments:



Compartment volumes: V_1 V_2 V_3

Initial concentrations: $S_1 = 4$ $S_2 = 10$ $S_3 = 0$

Volumes: $V_1 = 1$ $V_2 = 0.5$ $V_3 = 2$

Constants: $k_1 = 0.2$ $k_2 = 0.7$

Example #2: interpretation of differential equations

Express rates of change of species quantities (as amounts, not concentrations):

$$\frac{dn_{S_1}}{dt} = -\, r_1 \qquad\quad = -k_1 \cdot [S_1] \cdot V_1$$

$$\frac{dn_{S_2}}{dt} = +\, r_1 - r_2 \quad = +k_1 \cdot [S_1] \cdot V_1 - \, k_2 \cdot [S_2] \cdot V_2$$

$$\frac{dn_{S_3}}{dt} = \qquad +\, r_2 \quad = \qquad\qquad\qquad\qquad +\, k_2 \cdot [S_2] \cdot V_2$$

LibSBML and other SBML software infrastructure

Software infrastructure supporting use of SBML

http://sbml.org/wiki/icsb2006_tutorial

libSBML

Ben Bornstein and Sarah Keating

Application Programming Interface

- read
- write
- validate
- manipulate
- translate

Languages

- ISO C and C++
- java
- python
- perl
- lisp
- MATLAB
- octave (coming in libSBML-3)

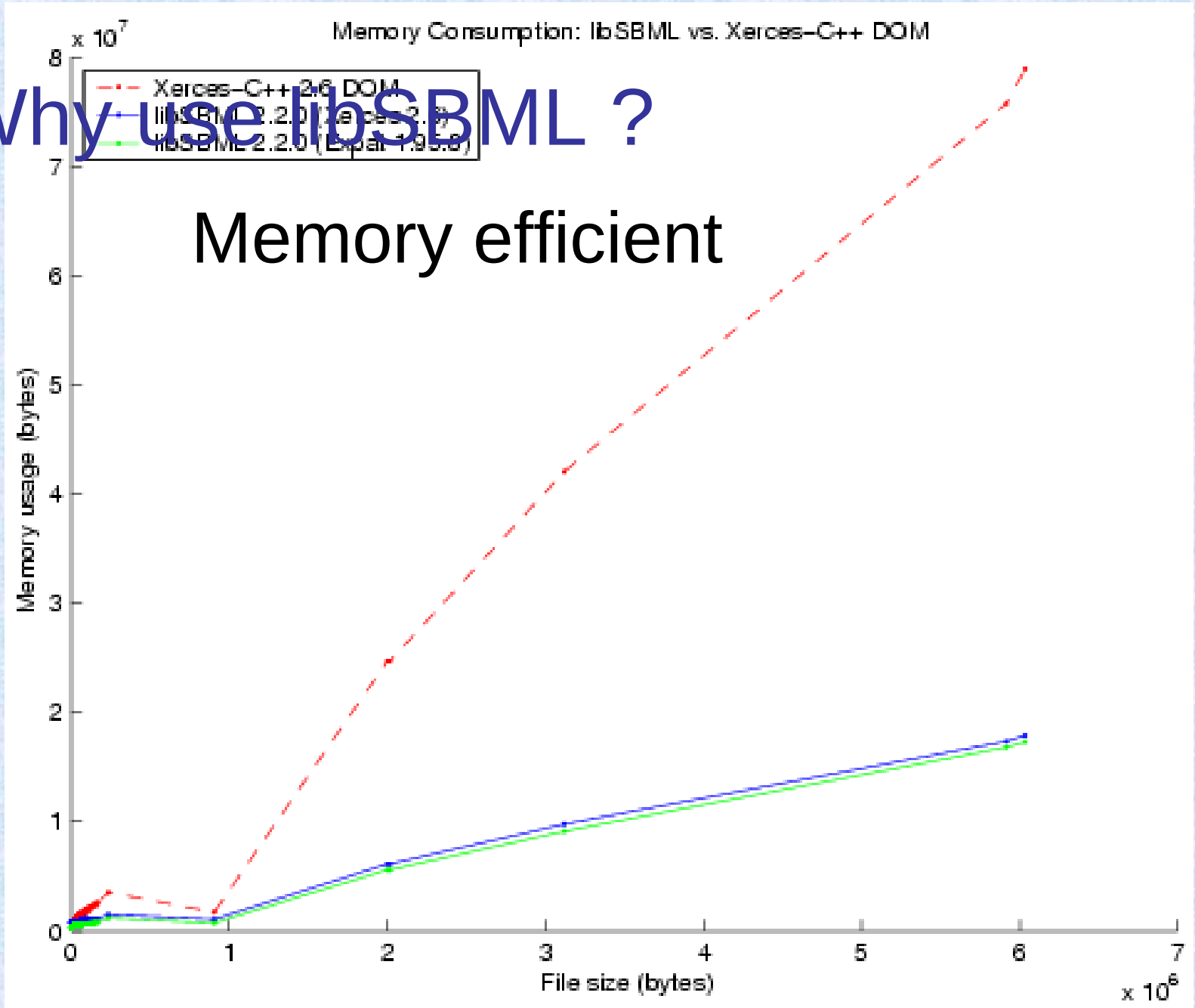
XML Parsers

- Xerces
- Expat
- libXML (coming in libSBML-3)

Platforms

- Linux
- Windows
- Mac OS

Why use libSBML ?

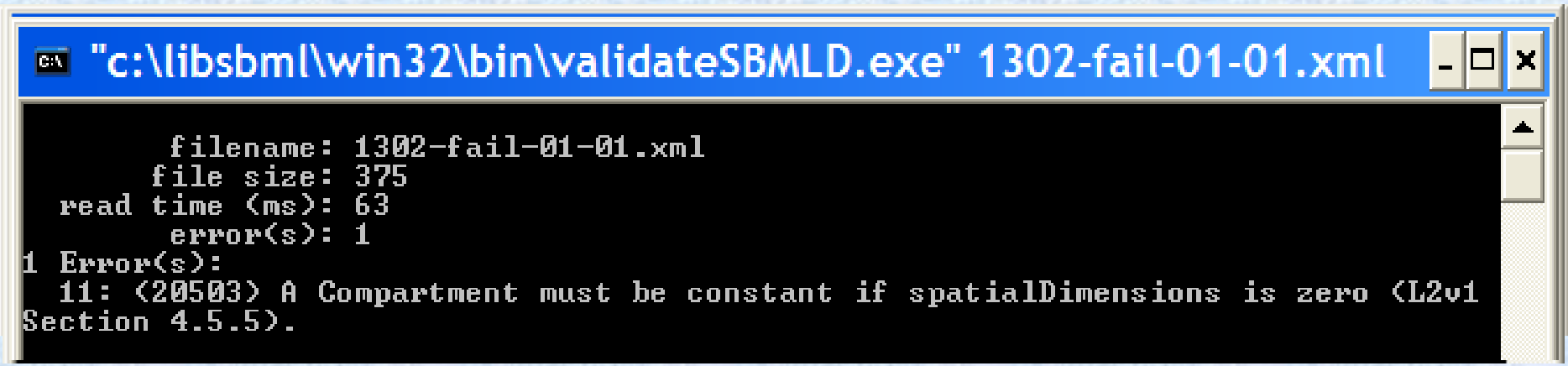


Memory efficient

Why use libSBML ?

SBML validation

- XML checks
- ordering checks
- syntax checks
- consistency checks



Why use libSBML ?

High level interface

Query a model

Model->getNumReactions()

Compartment->isSetSize()

Retrieve information from a model

Model->getSpecies(“s1”)

Species->getInitialConcentration()

Why use libSBML ?

High level interface

Add information to a model

Species->setInitialConcentration(2.0)

Compartment->setSize(5.6)

Create a model

Model->create(“my_model”)

Model->addCompartment(Compartment&)

Why use libSBML ?

Math support

Infix (Level 1)

```
k * A * B
```

MathML (Level 2)

```
<apply>
  <times/>
    <ci> k </ci>
    <ci> A </ci>
    <ci> B </ci>
</apply>
```

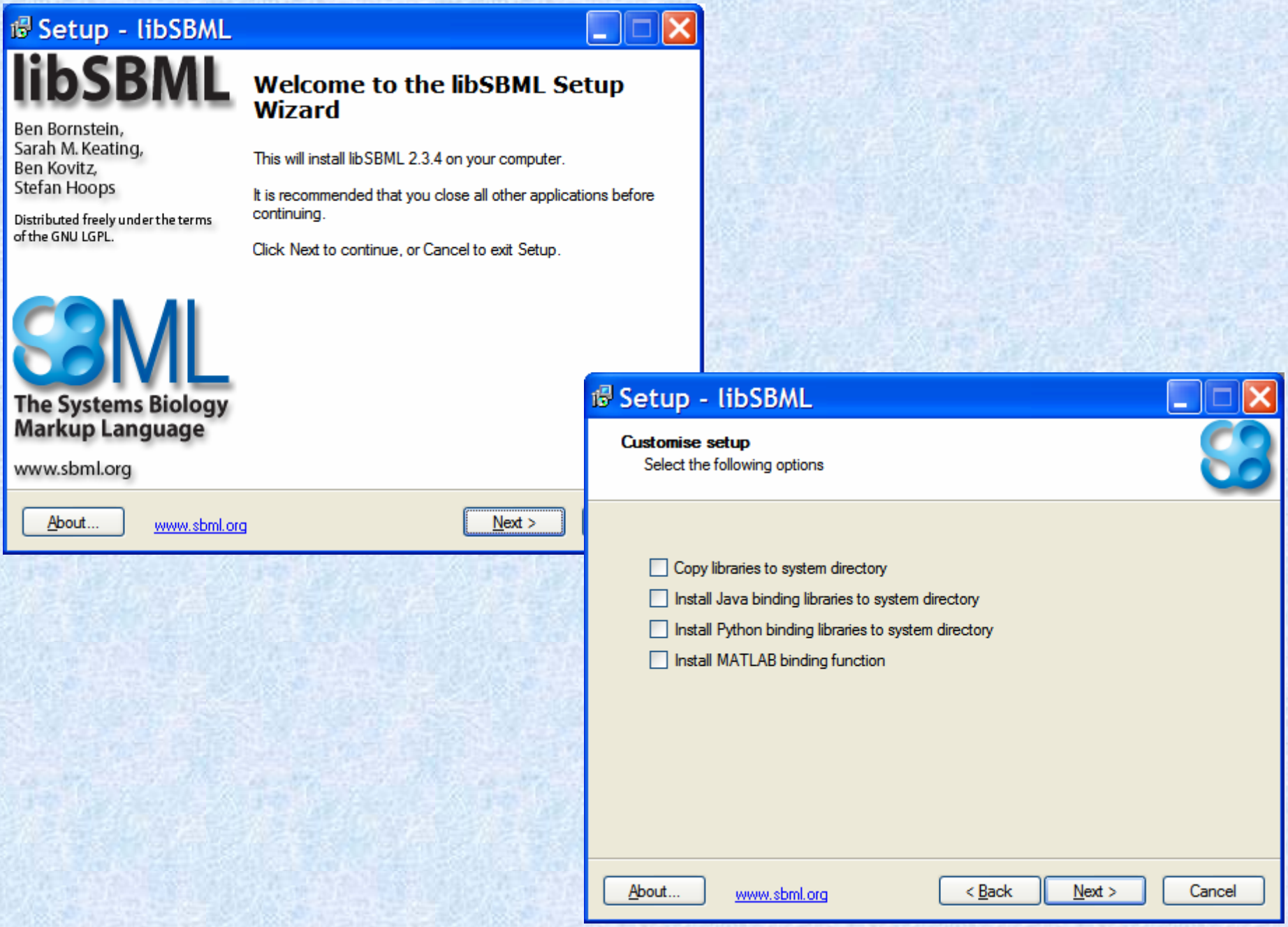
Getting started …

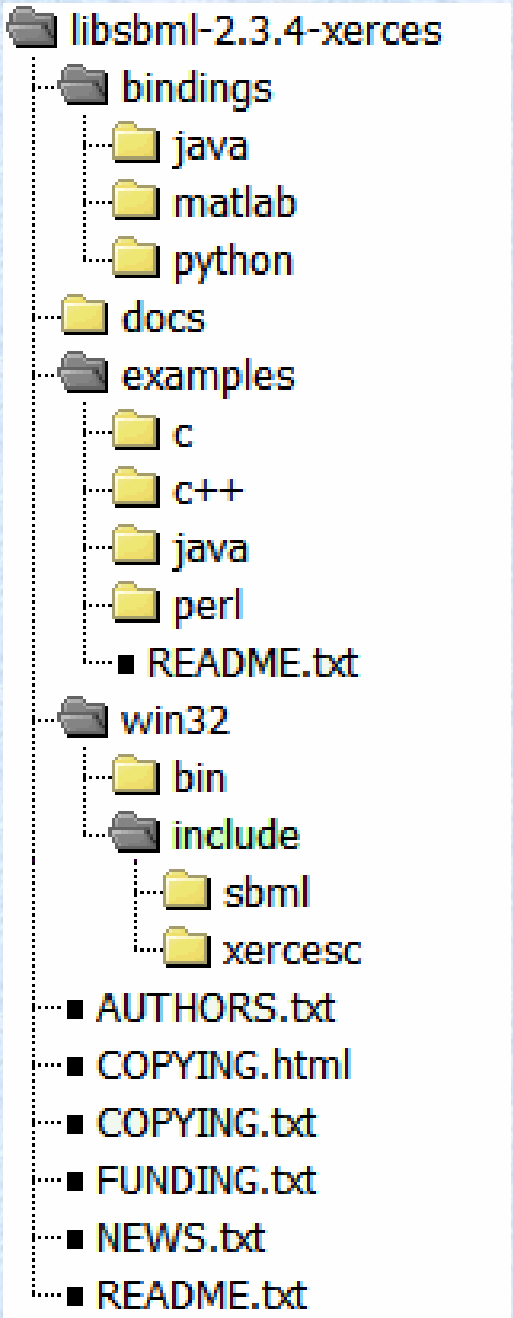
http://sourceforge.net/project/showfiles.php?group_id=71971&package_id=71670

Package	Release (date)	Filename	Size (bytes)	Downloads	Architecture	Type
libsbml						
Latest	2.3.4 [Notes] (2005-10-03 08:48)					
		libsbml-2.3.4-docs.zip	8618311	540	Platform-Independent	.zip
		libsbml-2.3.4-win-expat.exe	1548945	456	i386	.exe (32-bit Windows)
		libsbml-2.3.4-win-xerces.exe	4588356	599	i386	.exe (32-bit Windows)
		libsbml-2.3.4.zip	7211202	1392	Platform-Independent	Source .zip
Totals:	1	4	21966814	2987		

http://sbml.org/wiki/icsb2006_tutorial

Download libSBML


Setup - libSBML
libSBML
Ben Bornstein,
Sarah M. Keating,
Ben Kovitz,
Stefan Hoops
Distributed freely under the terms of the GNU LGPL.
Welcome to the libSBML Setup Wizard
This will install libSBML 2.3.4 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
SBML
The Systems Biology Markup Language
www.sbml.org
About...
www.sbml.org
Next >
Setup - libSBML
Customise setup
Select the following options
Copy libraries to system directory
Install Java binding libraries to system directory
Install Python binding libraries to system directory
Install MATLAB binding function
About...
www.sbml.org
< Back
Next >
Cancel



libsbml - Microsoft Visual C++ [design] - validateSBML.cpp

File Edit View Project Build Debug Tools Window Help

Debug SBMLReadErrorMessages

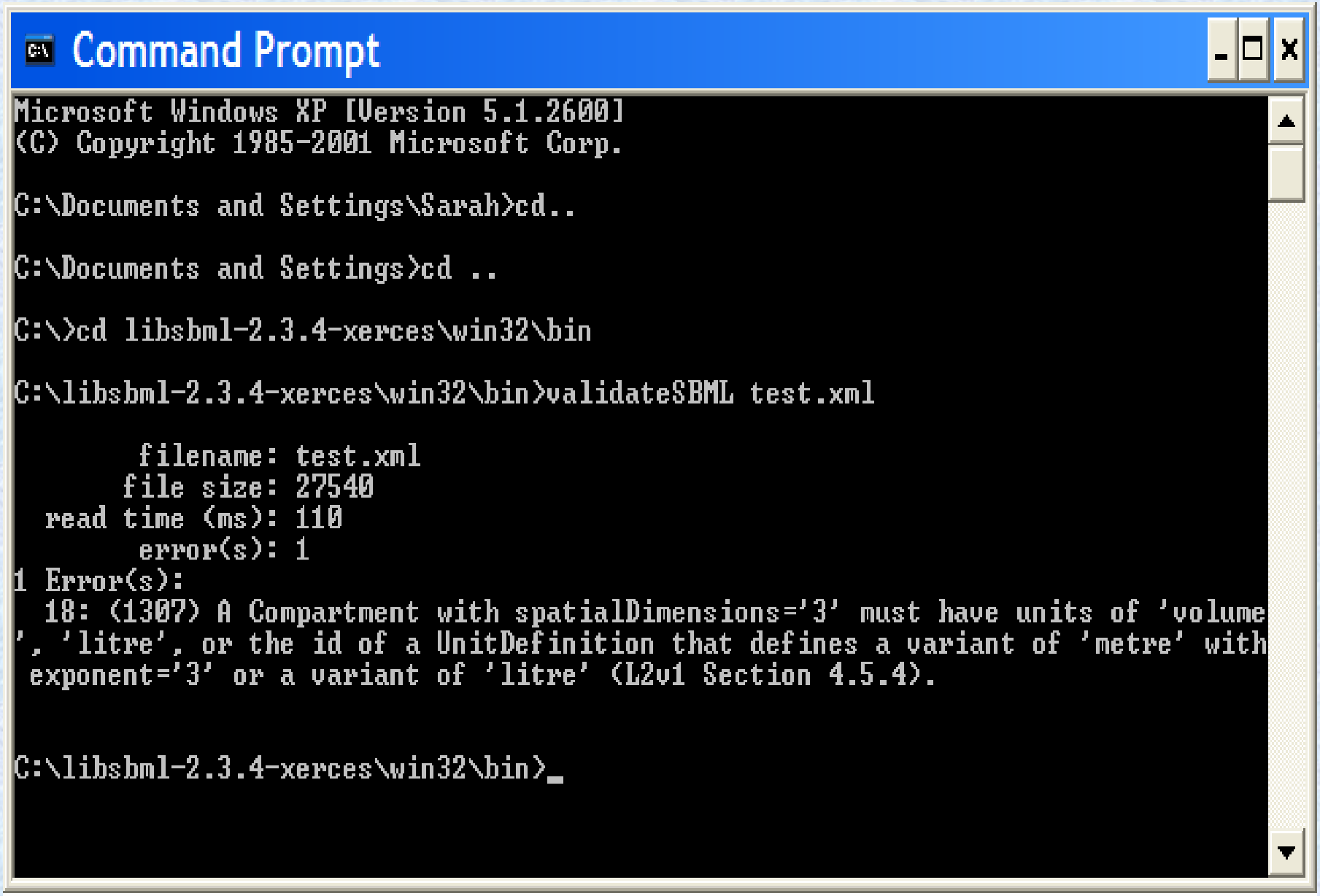
Solution Explorer

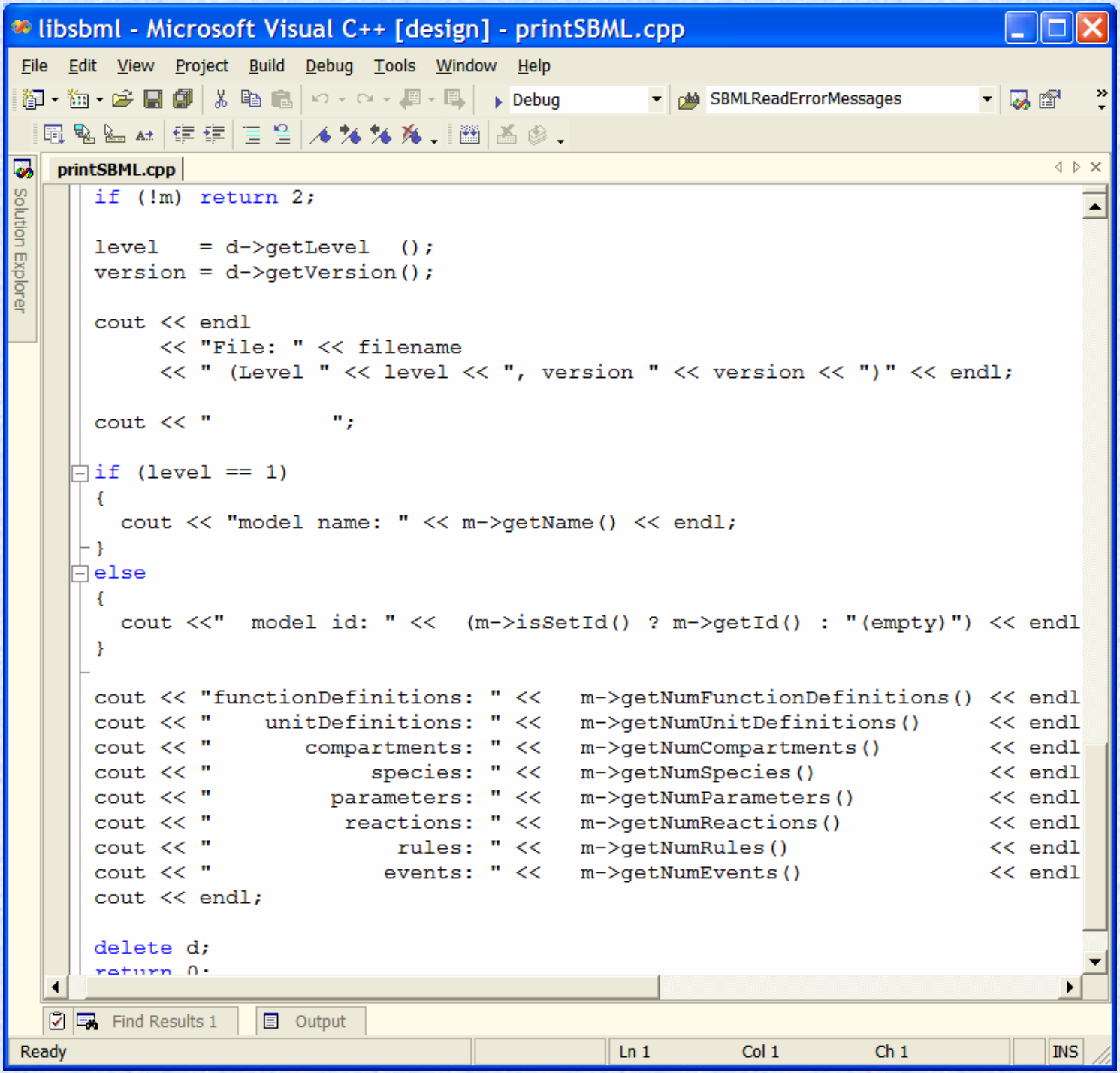
validateSBML.cpp

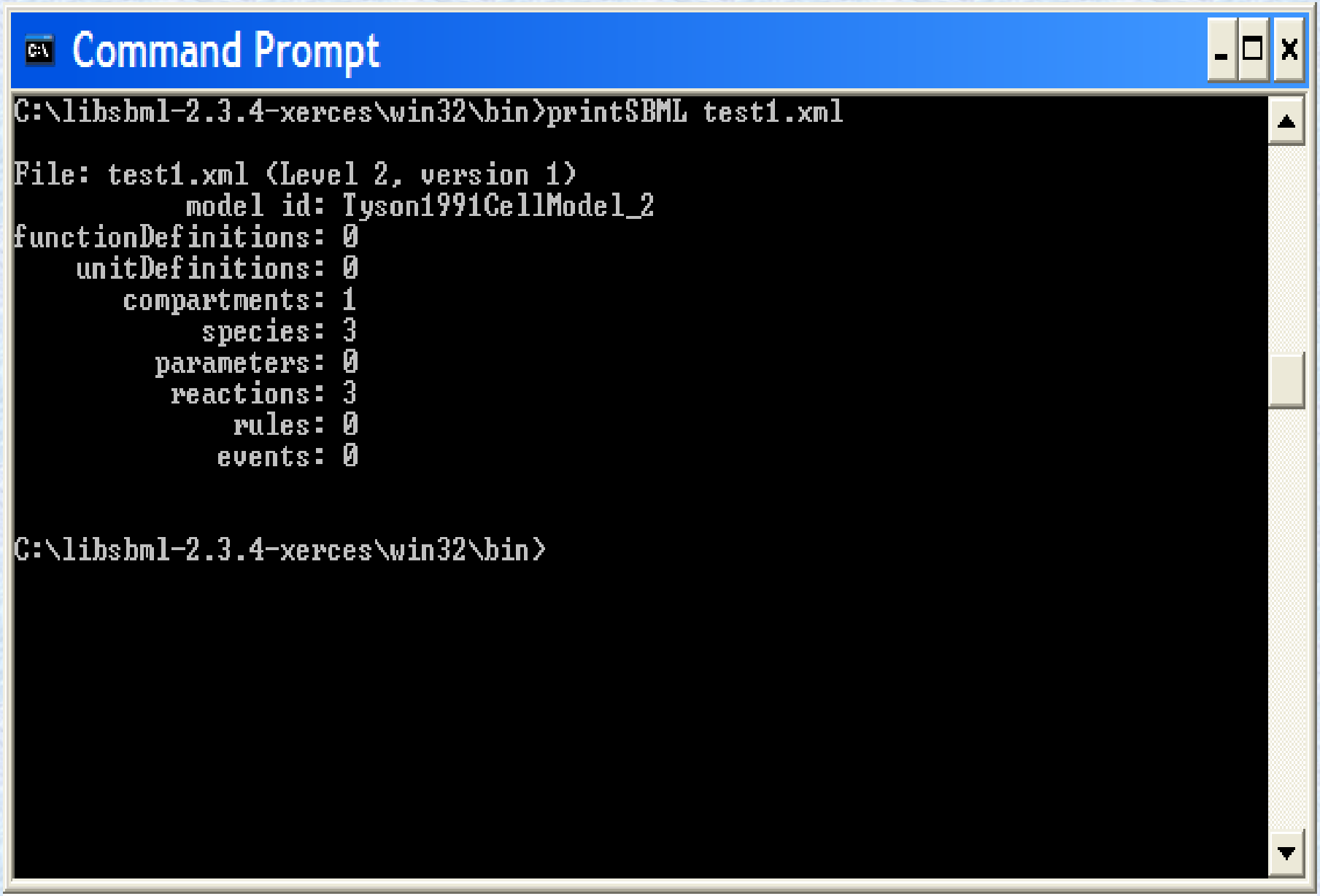
```
  filename = argv[1];

  start = getCurrentMillis();
  d     = sr.readSBML(filename);
  stop  = getCurrentMillis();

  errors  = d->getNumWarnings() + d->getNumErrors() + d->getNumFatals();
  errors += d->checkConsistency();

  size = getFileSize(filename);

  cout << endl;
  cout << "         filename: " << filename     << endl;
  cout << "        file size: " << size         << endl;
  cout << "  read time (ms): " << stop - start << endl;
  cout << "         error(s): " << errors       << endl;

  if (errors > 0)
  {
    d->printWarnings(cout);
    d->printErrors  (cout);
    d->printFatals  (cout);
  }

  cout << endl;

  delete d;
  return errors;
}
```

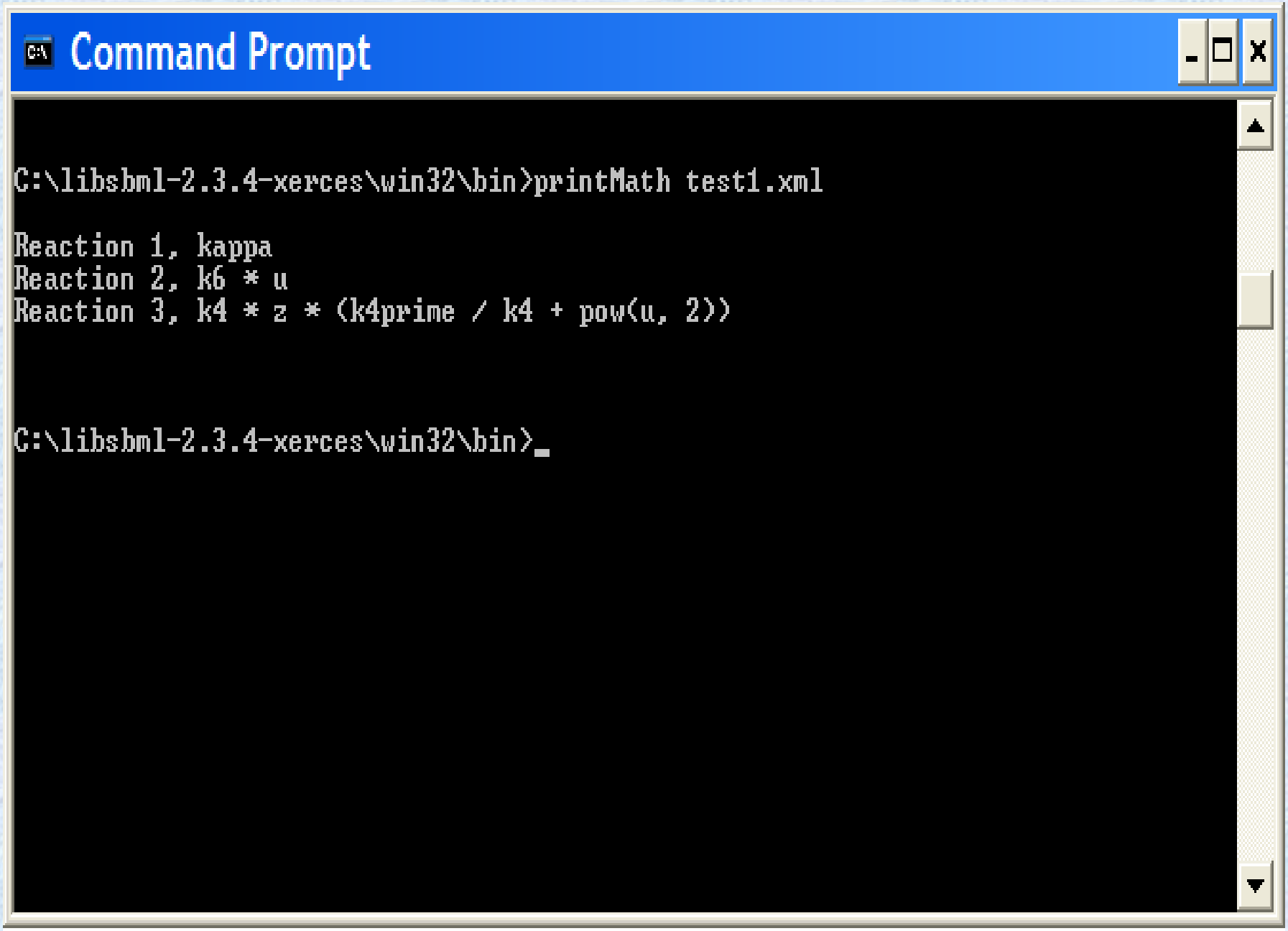
Find Results 1 Output

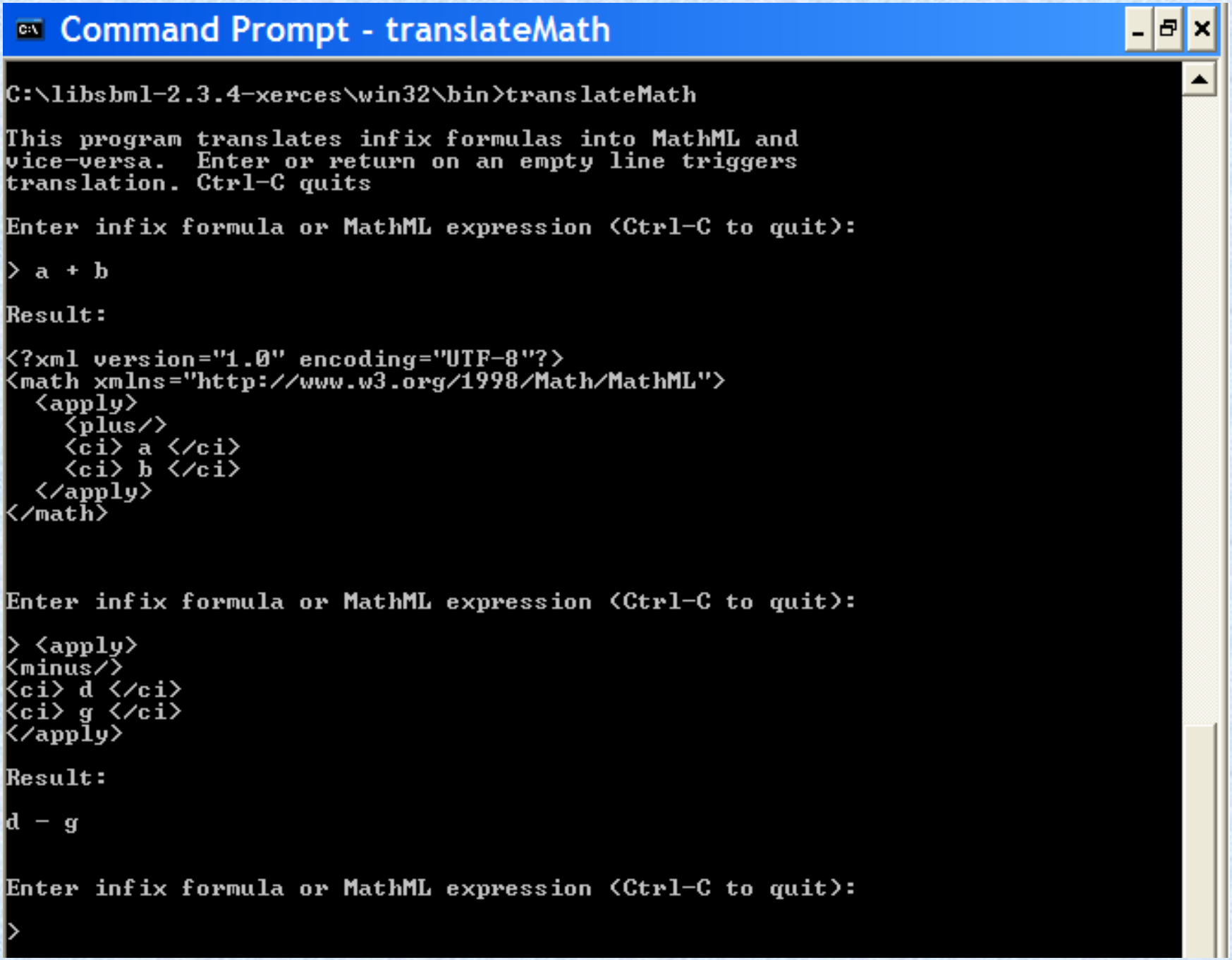
Ready Ln 98 Col 1 Ch 1 INS


Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Sarah>cd..
C:\Documents and Settings>cd ..
C:\>cd libsbml-2.3.4-xerces\win32\bin
C:\libsbml-2.3.4-xerces\win32\bin>validateSBML test.xml
filename: test.xml
file size: 27540
read time (ms): 110
error(s): 1
1 Error(s):
18: (1307) A Compartment with spatialDimensions='3' must have units of 'volume', 'litre', or the id of a UnitDefinition that defines a variant of 'metre' with exponent='3' or a variant of 'litre' (L2v1 Section 4.5.4).
C:\libsbml-2.3.4-xerces\win32\bin>


libsbml - Microsoft Visual C++ [design] - printSBML.cpp
File Edit View Project Build Debug Tools Window Help
Debug
SBMLReadErrorMessages
Solution Explorer
printSBML.cpp
if (!m) return 2;
level = d->getLevel ();
version = d->getVersion();
cout << endl
<< "File: " << filename
<< " (Level " << level << ", version " << version << ")" << endl;
cout << " ";
if (level == 1)
{
cout << "model name: " << m->getName() << endl;
}
else
{
cout <<" model id: " << (m->isSetId() ? m->getId() : "(empty)") << endl
}
cout << "functionDefinitions: " << m->getNumFunctionDefinitions() << endl
cout << " unitDefinitions: " << m->getNumUnitDefinitions() << endl
cout << " compartments: " << m->getNumCompartments() << endl
cout << " species: " << m->getNumSpecies() << endl
cout << " parameters: " << m->getNumParameters() << endl
cout << " reactions: " << m->getNumReactions() << endl
cout << " rules: " << m->getNumRules() << endl
cout << " events: " << m->getNumEvents() << endl
cout << endl;
delete d;
Find Results 1
Output
Ready
Ln 1
Col 1
Ch 1
INS


Command Prompt
C:\libsbml-2.3.4-xerces\win32\bin>printSBML test1.xml
File: test1.xml (Level 2, version 1)
model id: Tyson1991CellModel_2
functionDefinitions: 0
unitDefinitions: 0
compartments: 1
species: 3
parameters: 0
reactions: 3
rules: 0
events: 0
C:\libsbml-2.3.4-xerces\win32\bin>

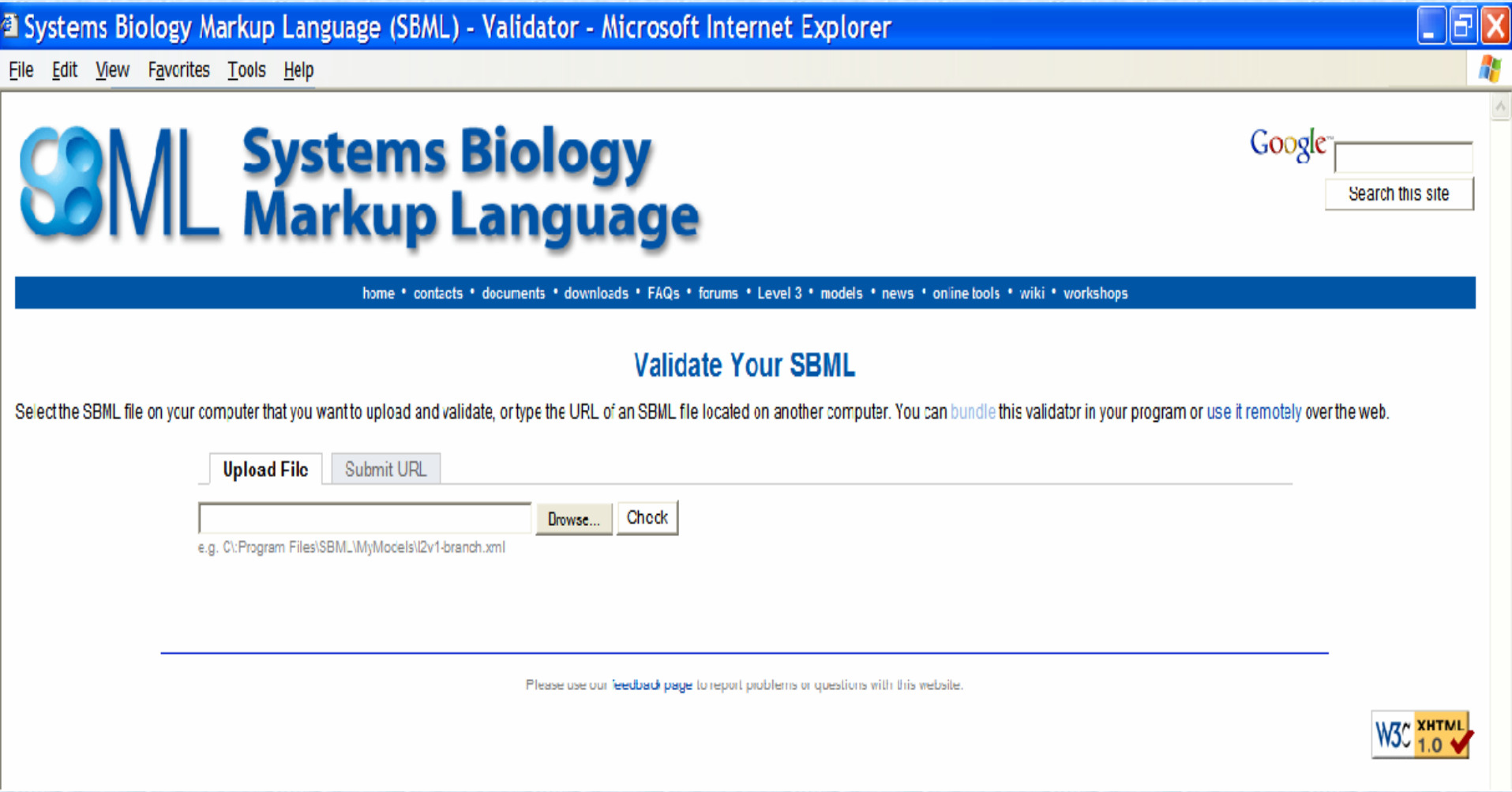

Command Prompt
C:\libsbml-2.3.4-xerces\win32\bin>printMath test1.xml
Reaction 1, kappa
Reaction 2, k6 * u
Reaction 3, k4 * z * (k4prime / k4 + pow(u, 2))
C:\libsbml-2.3.4-xerces\win32\bin>


Command Prompt - translateMath
C:\libsbml-2.3.4-xerces\win32\bin>translateMath
This program translates infix formulas into MathML and
vice-versa. Enter or return on an empty line triggers
translation. Ctrl-C quits
Enter infix formula or MathML expression (Ctrl-C to quit):
> a + b
Result:
<?xml version="1.0" encoding="UTF-8"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<apply>
<plus/>
<ci> a </ci>
<ci> b </ci>
</apply>
</math>
Enter infix formula or MathML expression (Ctrl-C to quit):
> <apply>
<minus/>
<ci> d </ci>
<ci> g </ci>
</apply>
Result:
d - g
Enter infix formula or MathML expression (Ctrl-C to quit):
>

Online validator

Ben Bornstein

http://sbml.org/validator/


Systems Biology Markup Language (SBML) - Validator - Microsoft Internet Explorer
File Edit View Favorites Tools Help
SBML Systems Biology Markup Language
Google
Search this site
home • contacts • documents • downloads • FAQs • forums • Level 3 • models • news • online tools • wiki • workshops
Validate Your SBML
Select the SBML file on your computer that you want to upload and validate, or type the URL of an SBML file located on another computer. You can bundle this validator in your program or use it remotely over the web.
Upload File
Submit URL
Browse...
Check
e.g. C:\Program Files\SBML\MyModels\l2v1-branch.xml
Please use our feedback page to report problems or questions with this website.
W3C XHTML 1.0

Validate Your SBML

Select the SBML file on your computer that you want to upload and validate, or type the URL of an SBML file located on another computer. You can bundle this validator in your program or use it remotely over the web.

Upload File | Submit URL

Browse... Check

e.g. C:\Program Files\SBML\MyModels\l2v1-branch.xml

Results

1301-fail-01-01.xml

This document is not valid SBML!

1 Error

1. Error *Line 11 Column 61:* **Compartment units must not be set if spatialDimensions is zero (L2v1 Section 4.5.4).**

```
<compartment id="c" spatialDimensions="0" units="ml"/>
```

Document Listing

```
1.  <?xml version="1.0" encoding="UTF-8"?>
2.
3.  <!--
4.  Fail: (1301) Compartment units must not be set if spatialDimensions is
5.  zero.
6.  -->
7.
8.  <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1">
9.  <model>
10.    <listOfCompartments>
11.    <compartment id="c" spatialDimensions="0" units="ml"/>
```

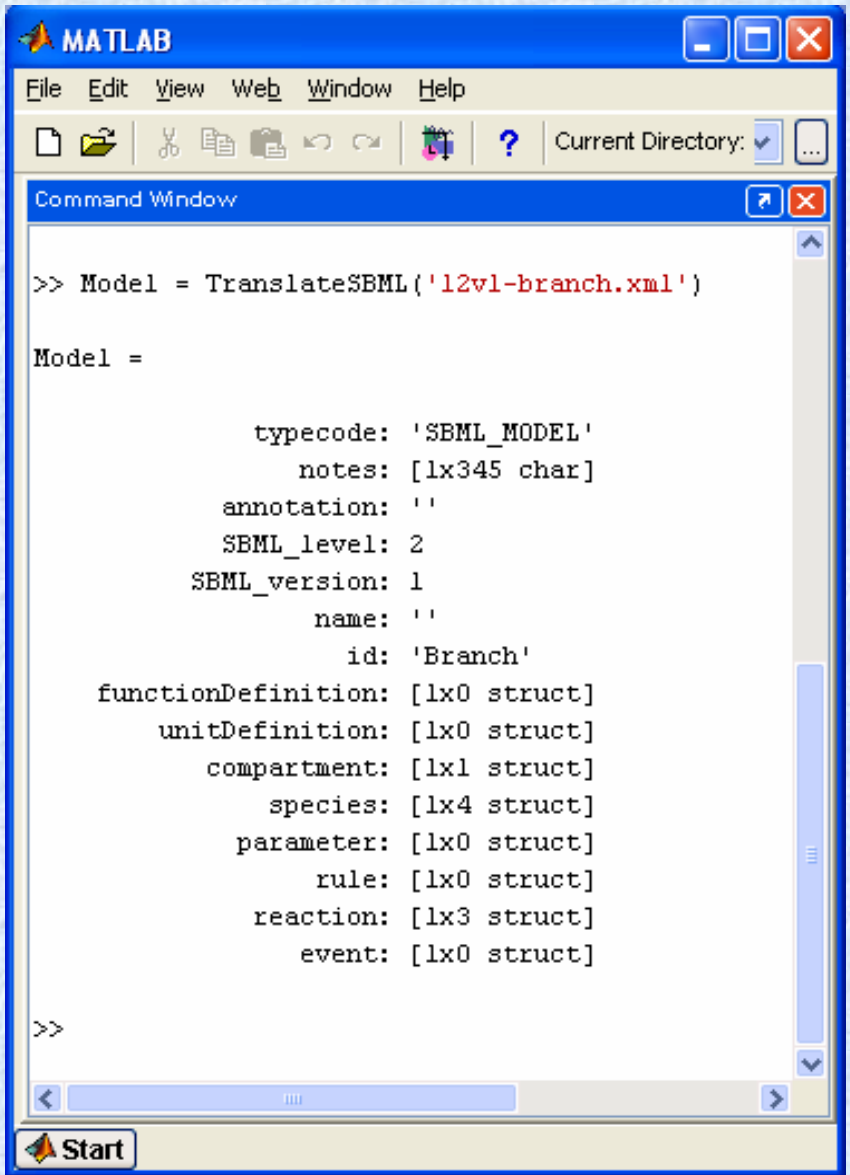
SBMLToolbox

Sarah Keating

An SBML toolbox for MATLAB users

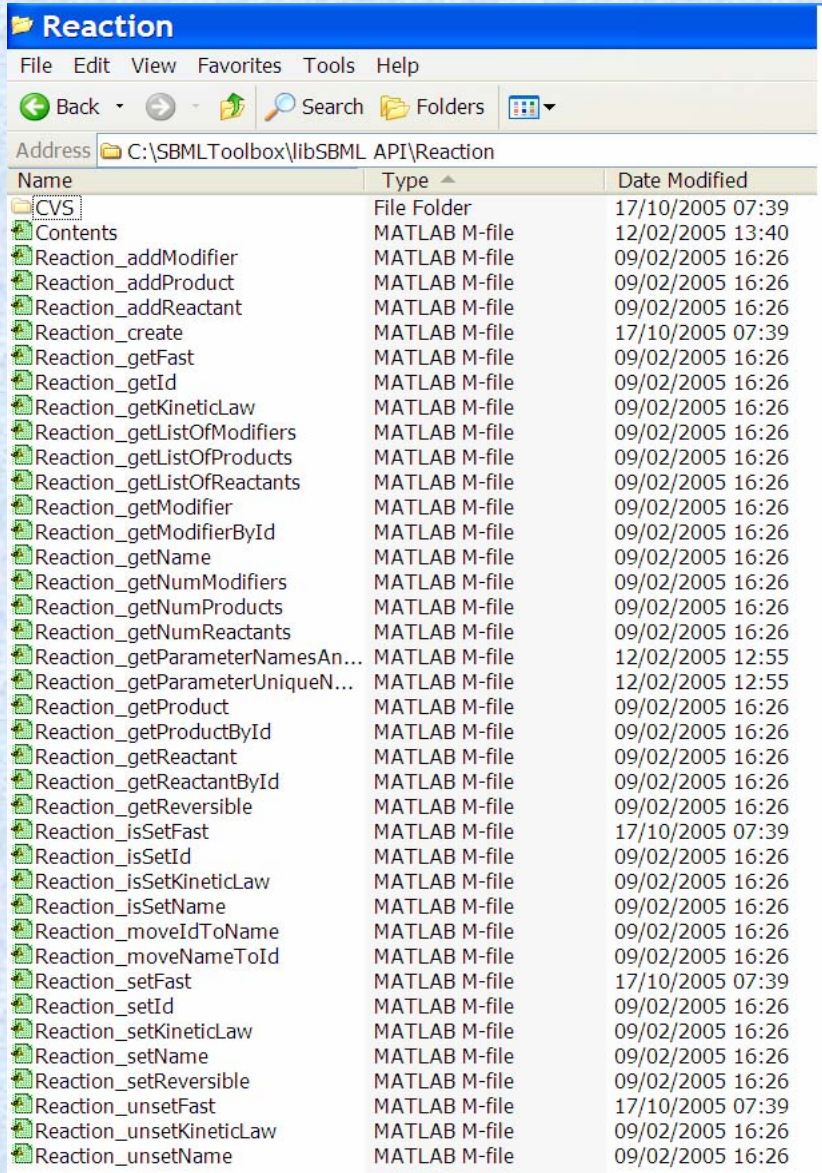
Why use SBMLToolbox ?

- import SBML into MATLAB
- represent models as MATLAB structures



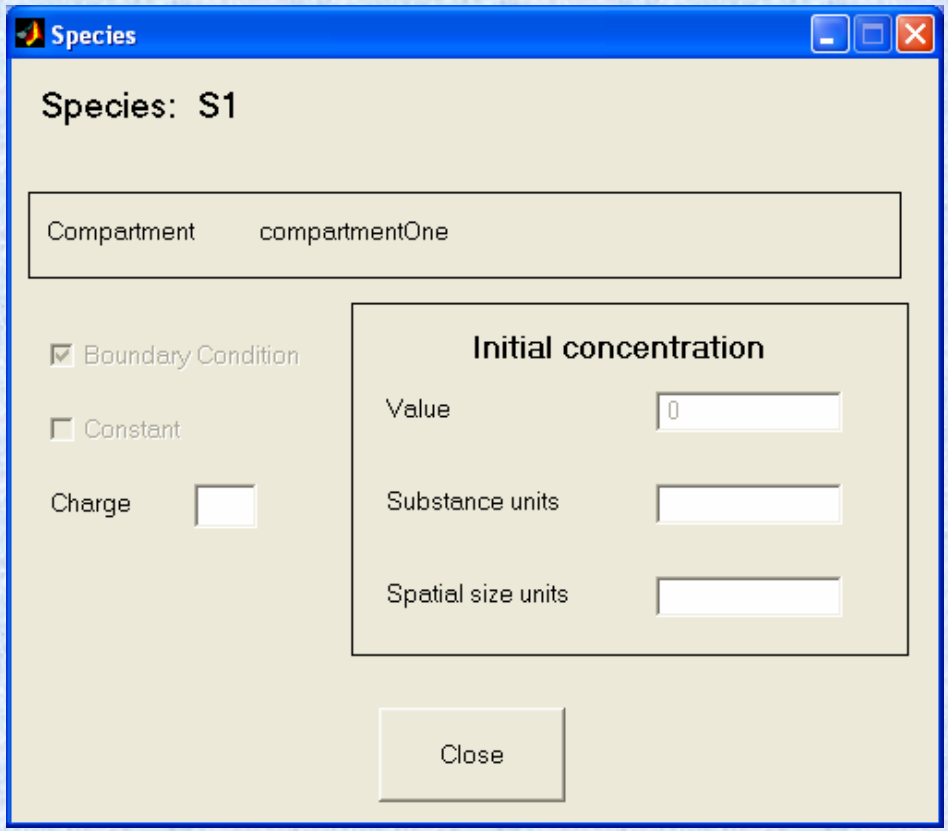
Why use SBMLToolbox ?

- mimic libSBML API
- examples of simulation



Why use SBMLToolbox ?

- GUI Model inspector/creator
- save/load models to MATLAB data files



- NOT a systems biology toolbox

- provides import/export between SBML and MATLAB

- provide examples of how MATLAB functionality can be applied to SBML models

Acknowledgements

- Ben Kovitz
- Stefan Hoops
- Christoph Flamm
- Rainer Machne
- Martin Ginkel
- Mike Hucka

Anyone who caught bugs, made suggestions, discussed …

••• Break •••

Additional SBML features and SBML Level 2 Version 2 differences

Units in SBML

- All mathematical entities can have units defined or implied. 2 ways:
  - Key object structures have explicit fields for setting units:

Structure	**Units fields**
Compartment	**`units`**
Species	**`substanceUnits, spatialSizeUnits`**
Parameter	**`units`**
Event	**`timeUnits`**
KineticLaw	***`substanceUnits, timeUnits`***

removed in L2V2

  - Built-in default units

Identifer	**Default**	**Possible scalable units**
`substance`	mole	mole, item, gram, kilogram, dimensionless
`volume`	litre	litre, cubic metre, dimensionless
`area`	square metre	square metre, dimensionless
`length`	metre	metre, dimensionless
`time`	second	second, dimensionless

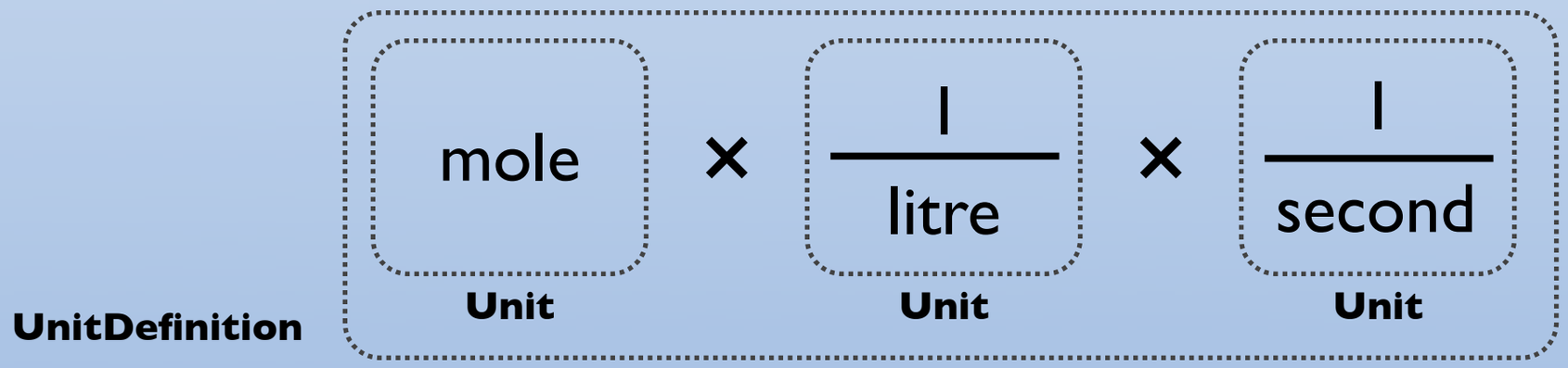
Redefining units

- Same mechanism for defining new units and redefining the built-ins:

UnitDefinition
id: UnitSId
name: string { use="optional" }
unit: Unit[1..*]

Unit
kind: UnitKind
exponent: int { use="optional" default="1" }
scale: int { use="optional" default="0" }
multiplier: double { use="optional" default="1" }

- Unit definition creates a new unit identifier (the value of `id`)
  - Unit identifier namespace is global but different from space of other id's
- Approach is multiplicative composition
  - E.g., moles/(litre • second) =



The meaning of fields in **Unit**

$$y_b\left\{u_b\right\} = y\left\{u\right\}\left(\frac{w\left\{u_b\right\}}{\left\{u\right\}}\right)$$

The meaning of fields in **Unit**

$$y_b\{u_b\} = y\{u\}\left(\frac{w\{u_b\}}{\{u\}}\right)$$

Quantity y in original units “u”

The meaning of fields in **Unit**

$$y_b\{u_b\} = y\{u\}\left(\frac{w\{u_b\}}{\{u\}}\right)$$

Units you want to convert into

Quantity y in original units “u”

The meaning of fields in **Unit**

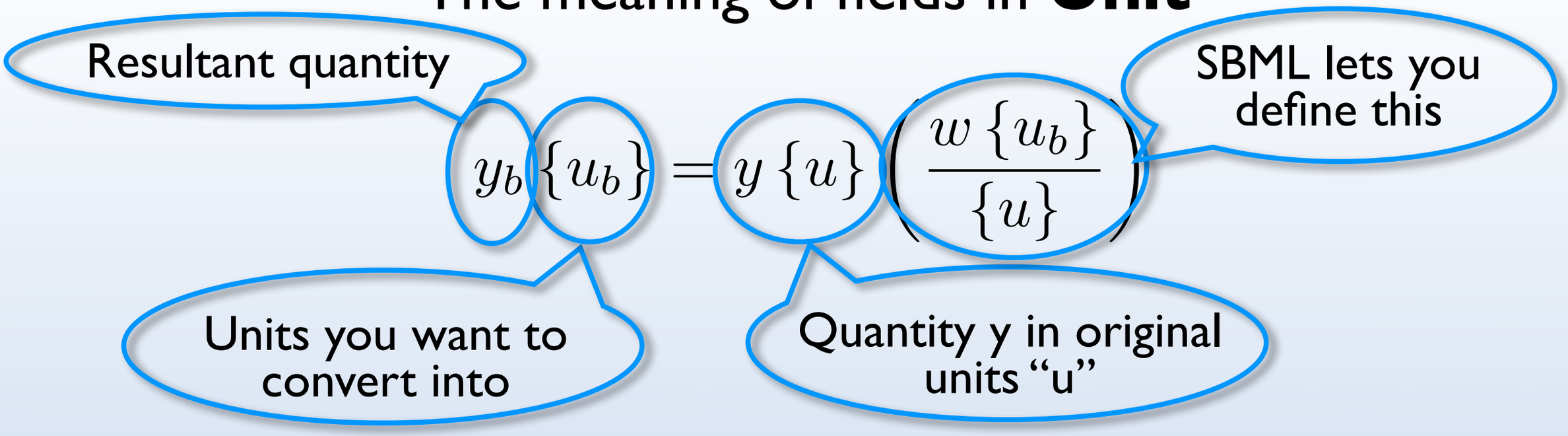
Resultant quantity

$$y_b\{u_b\} = y\{u\}\left(\frac{w\{u_b\}}{\{u\}}\right)$$

Units you want to convert into

Quantity y in original units “u”

The meaning of fields in **Unit**



The meaning of fields in **Unit**

$$y_b\left\{u_b\right\} = y\left\{u\right\}\left(\frac{w\left\{u_b\right\}}{\left\{u\right\}}\right)$$

The meaning of fields in **Unit**

$$y_b\left\{u_b\right\} = y\left\{u\right\}\left(\frac{w\left\{u_b\right\}}{\left\{u\right\}}\right)$$

$$\left\{u\right\} = \left(\texttt{multiplier}\cdot 10^{\texttt{scale}}\left\{u_b\right\}\right)^{\texttt{exponent}}$$

The meaning of fields in **Unit**

$$y_b\left\{u_b\right\} = y\left\{u\right\}\left(\frac{w\left\{u_b\right\}}{\left\{u\right\}}\right)$$

$$\left\{u\right\} = \left(\texttt{multiplier} \cdot 10^{\texttt{scale}}\left\{u_b\right\}\right)^{\texttt{exponent}}$$

UnitDefinition
id: UnitSId
name: string { use="optional" }
unit: Unit[1..*]

Unit
kind: UnitKind
exponent: int { use="optional" default="1" }
scale: int { use="optional" default="0" }
multiplier: double { use="optional" default="1" }

- **UnitKind** is an enumeration of base units (SI + a few extras)
  - mole, kelvin, second, metre, litre, gram, kilogram, item, dimensionless, etc.

Example unit definition

- Definition of "foot":

$$foot = (0.3048 \bullet 10^{0} \bullet metre)^{1}$$

$$y_b\ metres = 0.3048 \bullet y\ feet$$

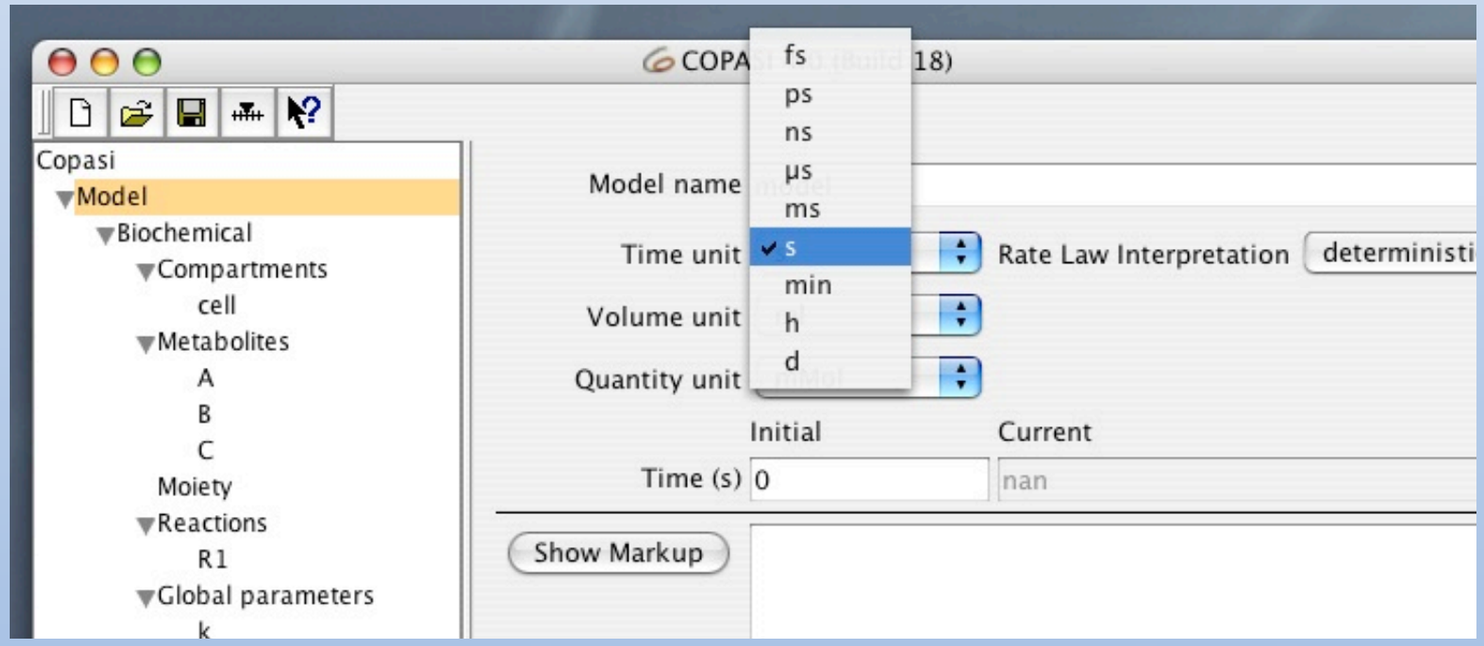
```
<listOfUnitDefinitions>
  <unitDefinition id="foot">
    <listOfUnits>
      <unit kind="metre" multiplier="0.3048"
                         exponent="1"
                         scale="0"/>
    </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
```

More on using units

- Default model-wide units
  - Redefine the special units “time”, “substance”, “volume”, “length”, “area”

```
<listOfUnitDefinitions>
  <unitDefinition id="volume">
    <listOfUnits>
      <unit kind="litre" scale="-3"/>
    </listOfUnits>
  </unitDefinition>
  <unitDefinition id="substance">
    <listOfUnits>
      <unit kind="mole" scale="-3"/>
    </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
```

- Tools usually provide a more direct way



Changes to unit system in L2V2

- No **`offset`** on UnitDefinition
- No predefined unit "Celsius"
  - Modeler needs to include necessary conversions explicitly
    - Could use a function definition
    - Could use an assignment rule

AssignmentRule, RateRule, AlgebraicRule

- "Rules" in SBML define extra mathematical expressions
  - E.g.: if need to express additional mathematical relationships beyond what is implied by the system of reactions
- 3 subtypes:

Rule type	General form	Example
algebraic	$0 = f(\mathbf{W})$	0 = S1 + S2
assignment	$x = f(\mathbf{V})$	x = y + z
rate	$dx/dt = f(\mathbf{W})$	dS/dt = 10.5

- Rules define relationships that hold at all times

Rules in the context of the overall system

$$
\begin{aligned}
dS_1/dt &= r_1 + r_2 + r_3 + \dots \\
dS_2/dt &= -r_1 + r_5 + \dots \\
&\dots \\
0 &= f_1(\mathbf{W}) \\
0 &= f_2(\mathbf{W}) \\
&\dots \\
&\dots \\
x &= g_1(\mathbf{W}) \\
y &= g_2(\mathbf{W}) \\
&\dots \\
dm/dt &= h_1(\mathbf{W}) \\
dq/dt &= h_2(\mathbf{W}) \\
&\dots
\end{aligned}
$$

Rules in the context of the overall system

Equations derived from reaction definitions

$$dS_1/dt = r_1 + r_2 + r_3 + \dots$$
$$dS_2/dt = -r_1 + r_5 + \dots$$

$$\dots$$
$$0 = f_1(\mathbf{W})$$
$$0 = f_2(\mathbf{W})$$
$$\dots$$
$$\dots$$
$$x = g_1(\mathbf{W})$$
$$y = g_2(\mathbf{W})$$
$$\dots$$
$$dm/dt = h_1(\mathbf{W})$$
$$dq/dt = h_2(\mathbf{W})$$
$$\dots$$

Rules in the context of the overall system

$$dS_1/dt = r_1 + r_2 + r_3 + \dots$$
$$dS_2/dt = -r_1 + r_5 + \dots$$
$$\dots$$

Algebraic rules

$$0 = f_1(\mathbf{W})$$
$$0 = f_2(\mathbf{W})$$

$$\dots$$
$$\dots$$
$$x = g_1(\mathbf{W})$$
$$y = g_2(\mathbf{W})$$
$$\dots$$
$$dm/dt = h_1(\mathbf{W})$$
$$dq/dt = h_2(\mathbf{W})$$
$$\dots$$

Rules in the context of the overall system

$$dS_1/dt = r_1 + r_2 + r_3 + \ldots$$
$$dS_2/dt = -r_1 + r_5 + \ldots$$
$$\ldots$$
$$0 = f_1(\mathbf{W})$$
$$0 = f_2(\mathbf{W})$$
$$\ldots$$

Assignment rules

$$x = g_1(\mathbf{W})$$
$$y = g_2(\mathbf{W})$$

$$\ldots$$
$$dm/dt = h_1(\mathbf{W})$$
$$dq/dt = h_2(\mathbf{W})$$
$$\ldots$$

Rules in the context of the overall system

$$dS_1/dt = r_1 + r_2 + r_3 + \dots$$
$$dS_2/dt = -r_1 + r_5 + \dots$$
$$\dots$$
$$0 = f_1(\mathbf{W})$$
$$0 = f_2(\mathbf{W})$$
$$\dots$$
$$x = g_1(\mathbf{W})$$
$$y = g_2(\mathbf{W})$$
$$\dots$$
$$dm/dt = h_1(\mathbf{W})$$
$$dq/dt = h_2(\mathbf{W})$$
$$\dots$$

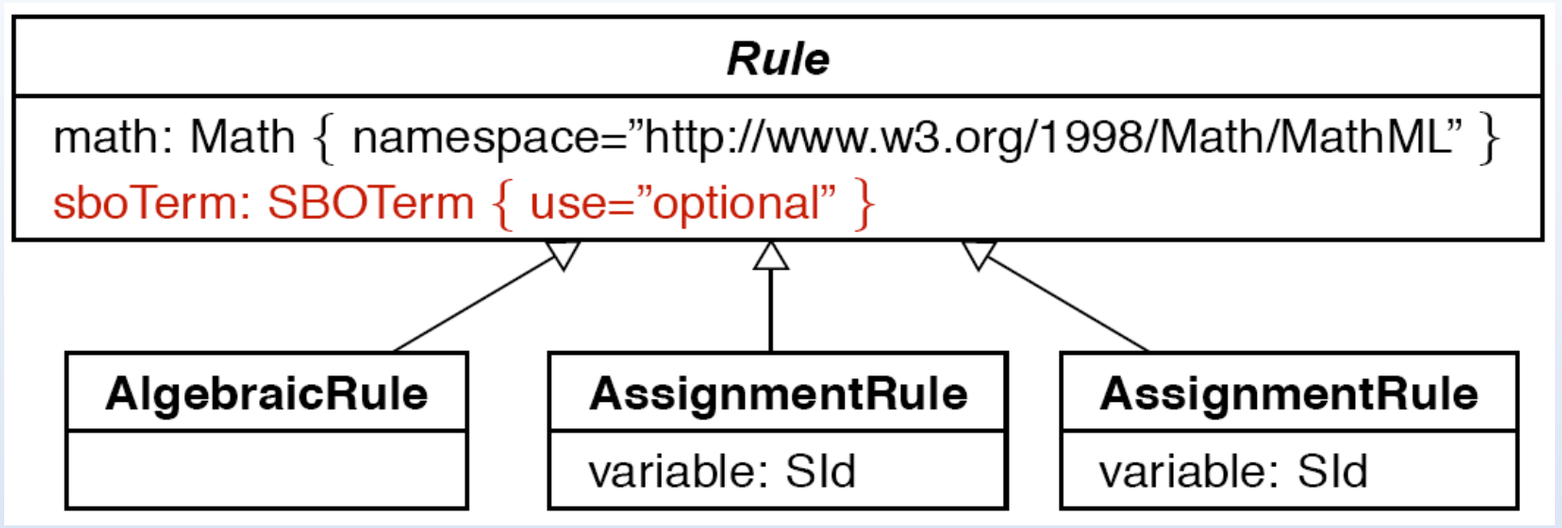
Rate rules

Rules in the context of the overall system

$$
\begin{aligned}
dS_1/dt &= r_1 + r_2 + r_3 + \dots \\
dS_2/dt &= -r_1 + r_5 + \dots \\
&\dots \\
0 &= f_1(\mathbf{W}) \\
0 &= f_2(\mathbf{W}) \\
&\dots \\
x &= g_1(\mathbf{W}) \\
y &= g_2(\mathbf{W}) \\
&\dots \\
dm/dt &= h_1(\mathbf{W}) \\
dq/dt &= h_2(\mathbf{W}) \\
&\dots
\end{aligned}
$$

Rules and equations from reactions are taken together

SBML object structures for rules



- E.g.:

```
<assignmentRule variable="k">
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
      <divide/>
      <ci> k3 </ci>
      <ci> k2 </ci>
    </apply>
  </math>
</assignmentRule>
```

Rule semantics and restrictions

- AssignmentRule
  - Assignment to a species, compartment or parameter overrides the initial value given in the definition
  - Variable must not be set by both an AssignmentRule and a RateRule
  - Variable must not be set by both an AssignmentRule and an InitialAssignment
- RateRule
  - Variable must not be set by both an AssignmentRule and a RateRule
- Overall restrictions on the whole model:
  - The model must not contain algebraic loops
  - The model must not be overdetermined
    - Can only happen if model contains algebraic rules
    - Can be assessed statically (L2V2 spec contains one possible approach)

User-defined functions

FunctionDefinition
id: SId
name: string { use="optional" }
math: (lambda:Lambda) { namespace="http://www.w3.org/1998/Math/MathML" }
sboTerm: SBOTerm { use="optional" }

- All references to model variables must be passed as arguments
- References to other user-defined functions must be for functions defined earlier in the file
  - I.e., no forward references to functions
  - No recursive or mutually-recursive functions
- (Point of all this: can implement functions as macro substitutions)

User-defined functions: example

```
<functionDefinition id="pow3">
   <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
         <bvar><ci> x </ci></bvar>
         <apply>
            <power/>
            <ci> x </ci>
            <cn> 3 </cn>
         </apply>
      </lambda>
   </math>
</functionDefinition>
...
<listOfReactions>
   <reaction id="reaction_1">
   ...
   <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
            <ci> pow3 </ci>
            <ci> S1 </ci>
         </apply>
      </math>
   </kineticLaw>
   ...
   </reaction>
</listOfReactions>
```

User-defined functions: example

```
<functionDefinition id="pow3">
   <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
         <bvar><ci> x </ci></bvar>
         <apply>
            <power/>
            <ci> x </ci>
            <cn> 3 </cn>
         </apply>
      </lambda>
   </math>
</functionDefinition>
...
<listOfReactions>
   <reaction id="reaction_1">
   ...
   <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
            <ci> pow3 </ci>
            <ci> S1 </ci>
         </apply>
      </math>
   </kineticLaw>
   ...
   </reaction>
</listOfReactions>
```

User-defined functions: example

```
<functionDefinition id="pow3">
   <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
         <bvar><ci> x </ci></bvar>
         <apply>
            <power/>
            <ci> x </ci>
            <cn> 3 </cn>
         </apply>
      </lambda>
   </math>
</functionDefinition>
...
<listOfReactions>
   <reaction id="reaction_1">
   ...
   <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
            <ci> pow3 </ci>
            <ci> S1 </ci>
         </apply>
      </math>
   </kineticLaw>
   ...
   </reaction>
</listOfReactions>
```

Uses <ci> to reference function identifier

User-defined functions: example

```
<functionDefinition id="pow3">
   <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
         <bvar><ci> x </ci></bvar>
         <apply>
            <power/>
            <ci> x </ci>
            <cn> 3 </cn>
         </apply>
      </lambda>
   </math>
</functionDefinition>
...
<listOfReactions>
   <reaction id="reaction_1">
   ...
   <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
            <ci> pow3 </ci>
            <ci> S1 </ci>
         </apply>
      </math>
   </kineticLaw>
   ...
   </reaction>
</listOfReactions>
```

Uses <bvar> to pass arguments

Events

Event
id: SId name: string { use="optional" } trigger: (math : Math { namespace="http://www.w3.org/1998/Math/MathML" }) delay: (math : Math { namespace="http://www.w3.org/1998/Math/MathML" }) { minOccurs="0" maxoccurs="1" } timeUnits: UnitSId { use="optional" } eventAssignment: EventAssignment[1..*] sboTerm: SBOTerm { use="optional" }

EventAssignment
variable: SId math: Math { namespace="http://www.w3.org/1998/Math/MathML" } sboTerm: SBOTerm { use="optional" }

- Defines discontinuous changes in model variables
- **EventAssignment** `variable` can be
  - species
  - compartment
  - parameter
- Trigger and delay conditions are full mathematical expressions

Usage points: events

- Triggered on transition from false to true
  - Not possible for event to trigger at t=0 — no transition in variable values
- Assignment 'math' expression evaluated *when event is fired*
  - Effect can be delayed (expressed using 'delay' expression)
- Cannot create or destroy species/compartments/reactions/etc.
  - But you could use a variable that acts as a switch: x*switch
- Warning: not well supported by most software tools (yet)

Compartments

Compartment
id: SId
name: string { use="optional" }
compartmentType: SId { use="optional" }
spatialDimensions: int { maxInclusive="3" minInclusive="0" use="optional" default="3" }
size: double { use="optional" }
units: UnitSId { use="optional" }
outside: SId { use="optional" }
constant: boolean { use="optional" default="true" }

- There is no default size!
- Extremely good practice to always set the size
  - Can use **`size`** field, assignment rules, or (in L2V2) initial assignments
- **`size`** value only allowed if **`spatialDimensions`** != 0
- Units of size must be consistent

Species

Species

id: SId
name: string { use="optional" }
speciesType: SId { use="optional" }
compartment: SId
initialAmount: double { use="optional" }
inintialConcentration: double { use="optional" }
substanceUnits: UnitSId { use="optional" }
spatialSizeUnits: UnitSId { use="optional" }
hasOnlySubstanceUnits: boolean { use="optional" default="false" }
boundaryCondition: boolean { use="optional" default="false" }
charge: int { use="optional" } *deprecated*
constant: boolean { use="optional" default="false" }

- Units of substance and spatial size can be set separately
- **`hasOnlySubstanceUnits`**
  - Boolean: should species quantity always be as substance or substance/size?
  - Important: setting **`initialAmount`** or **`initialConcentration`** does **not** imply units will be substance or substance/size, respectively

Species

Species

id: SId
name: string { use="optional" }
speciesType: SId { use="optional" }
compartment: SId
initialAmount: double { use="optional" }
inintialConcentration: double { use="optional" }
substanceUnits: UnitSId { use="optional" }
spatialSizeUnits: UnitSId { use="optional" }
hasOnlySubstanceUnits: boolean { use="optional" default="false" }
boundaryCondition: boolean { use="optional" default="false" }
charge: int { use="optional" } *deprecated*
constant: boolean { use="optional" default="false" }

- Units of substance and spatial size can be set separately
- **`hasOnlySubstanceUnits`**
  - Boolean: should species quantity always be as substance or substance/size?
  - Important: setting **`initialAmount`** or **`initialConcentration`** does **not** imply units will be substance or substance/size, respectively

interpretation is concentration unless **`hasOnlySubstanceUnits`**=true

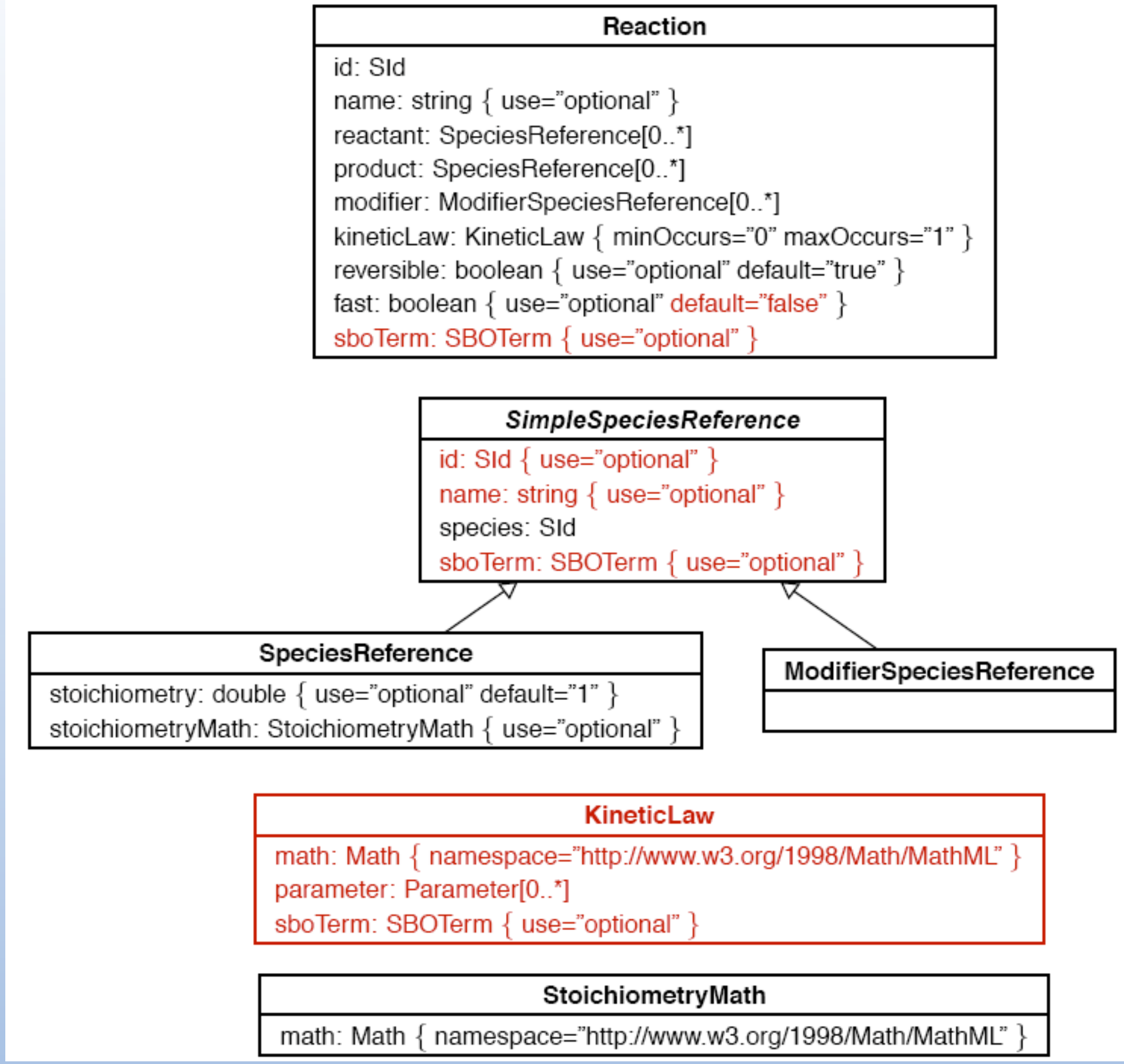
Species: "boundary conditions" and "constant" species

- Some intuitive explanations:
  - "Boundary condition": when a species is a reactant or product in one or more reactions, but its quantity is not changed by those reactions
    - E.g., when a chemical is buffered in an experimental set-up, or the modeler wants to assume there is an infinite quantity of the species
    - ODE (or equivalent) shouldn't be constructed based on the reactions
      - But SBML rules and other constructs may still change it
  - "Constant": if a species' quantity is constant
    - Different from being a boundary condition—**`boundaryCondition`** flag says whether an ODE should be constructed

Species: **`constant`** and **`boundaryCondition`**

`constant` value	**`boundaryCondition`** value	can have assignment or rate rule	can be reactant or product	species' quantity can be changed by
true	true	no	yes	(never changes)
false	true	yes	yes	rules and events
true	false	no	no	(never changes)
false	false	yes	yes	reactions *or* rules (but not both), and events

Reactions


Reaction
id: SId
name: string { use="optional" }
reactant: SpeciesReference[0..*]
product: SpeciesReference[0..*]
modifier: ModifierSpeciesReference[0..*]
kineticLaw: KineticLaw { minOccurs="0" maxOccurs="1" }
reversible: boolean { use="optional" default="true" }
fast: boolean { use="optional" default="false" }
sboTerm: SBOTerm { use="optional" }
SimpleSpeciesReference
id: SId { use="optional" }
name: string { use="optional" }
species: SId
sboTerm: SBOTerm { use="optional" }
SpeciesReference
stoichiometry: double { use="optional" default="1" }
stoichiometryMath: StoichiometryMath { use="optional" }
ModifierSpeciesReference
KineticLaw
math: Math { namespace="http://www.w3.org/1998/Math/MathML" }
parameter: Parameter[0..*]
sboTerm: SBOTerm { use="optional" }
StoichiometryMath
math: Math { namespace="http://www.w3.org/1998/Math/MathML" }

Usage points: reactions (specific)

- Reversibility:
  - Because "kinetic law" is optional, it's useful to have separate flag
    - Some types of analyses can be done without simulation or kinetic law
  - **`reversible`** flag should be true only if rate expression represents combined effect of forward and backward reaction
    - Converse: if false, it's a statement that the reaction only proceeds in the forward direction
  - *Caution*: not clear how to define reversible reactions for stochastic simulation. Best practice: always define 2 reactions explicitly in that case.

New in SBML Level 2 Version 2

Model
id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

Added in Level 2 Version 2

Compartment types and species types

- L2V2 only
- Does not affect mathematical interpretation

```
<listOfSpeciesTypes>
  <speciesType id="ATP"/>
</listOfSpeciesTypes>
...
<listOfSpecies>
  <species id="ATP_in_cytosol" speciesType="ATP"
           compartment="cytosol"/>
  <species id="ATP_in_nucleus" speciesType="ATP"
           compartment="nucleus"/>
</listOfSpecies>
```

Compartment types and species types

- L2V2 only
- Does not affect mathematical interpretation

```
<listOfSpeciesTypes>
  <speciesType id="ATP"/>
</listOfSpeciesTypes>
...
<listOfSpecies>
  <species id="ATP_in_cytosol" speciesType="ATP"
           compartment="cytosol"/>
  <species id="ATP_in_nucleus" speciesType="ATP"
           compartment="nucleus"/>
</listOfSpecies>
```

Usage points: how do you assign initial values?

- Multiple approaches:

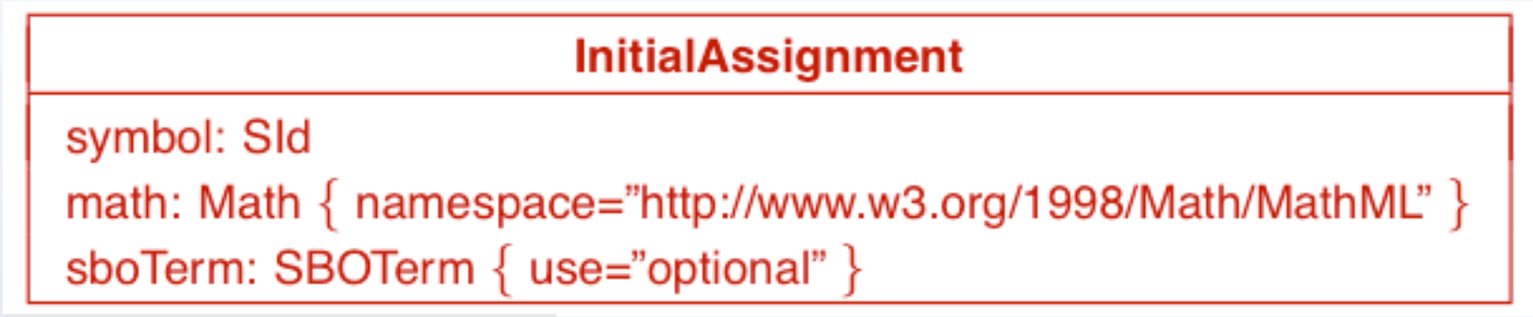
1. Use the appropriate value field on an element (most portable approach)

Element	*Initial value field(s)*
`species`	**`initialConcentration`** **`initialAmount`**
`compartment`	**`size`**
`parameter`	**`value`**

   - Limitation: scalar values only, not a mathematical expression

2. Use assignment rules (somewhat less portable)
   - Mathematical expression
3. Use *Initial Assignment* in Level 2 Version 2
   - Mathematical expression

Initial assignments



- Overrides value given in object definition
  - But the object definition must still be provided
- Applies for t <= 0
- Cannot have both an initial assignment and an assignment rule for the same identifier

Constraints

Constraint
math: Math { namespace="http://www.w3.org/1998/Math/MathML" } message: (any: { namespace="http://www.w3.org/1999/xhtml" }) { minOccurs="0" maxOccurs="1" } sboTerm: SBOTerm { use="optional" }

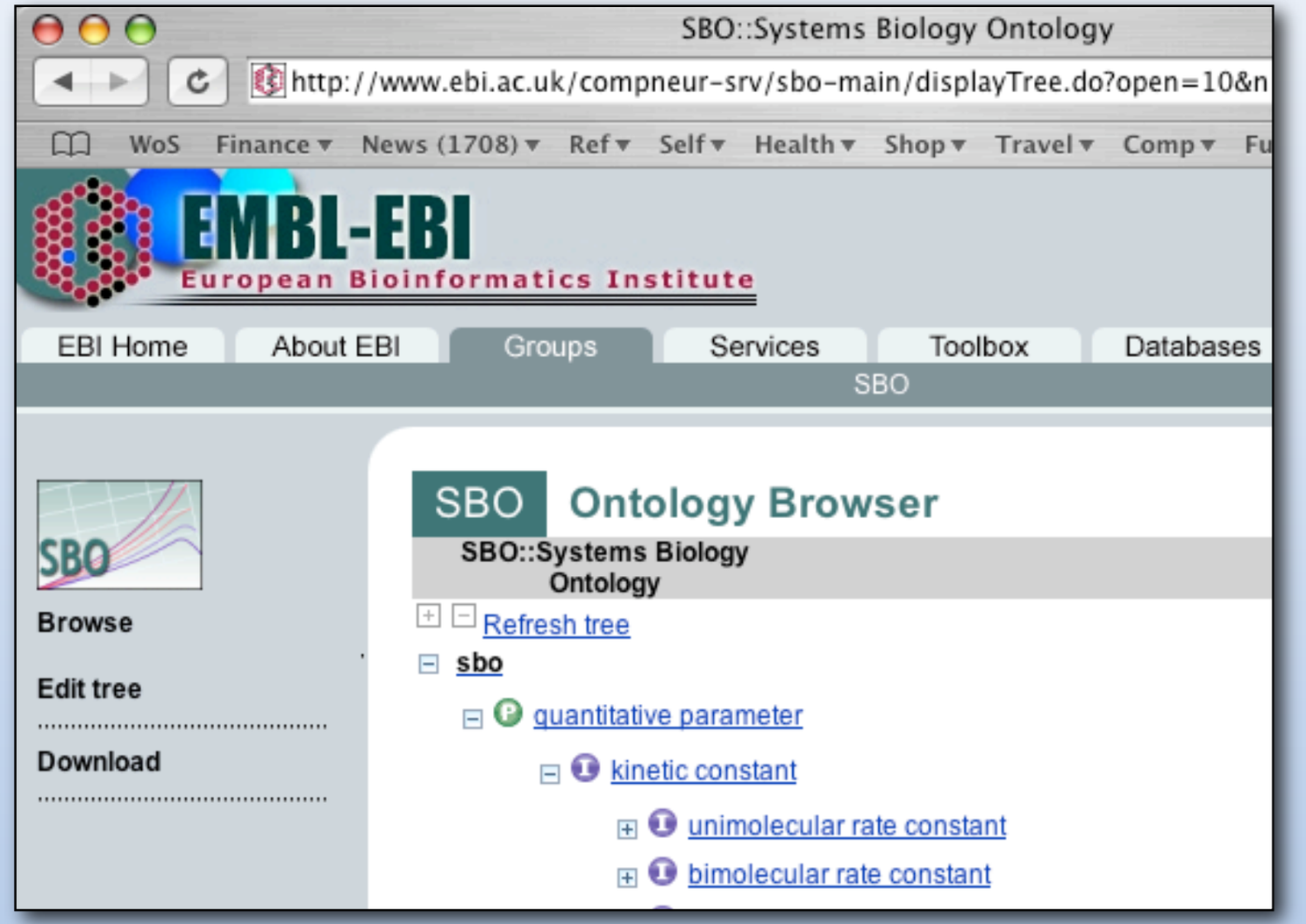
- Allows stating assumptions under which simulation is valid
- Intuitively: if the math expression is true, model is in a valid state
  - As soon as the expression evalues to false, the model is in an *invalid* state
  - SBML does not require a particular action to be taken
    - But it would be a good idea for the software system to stop or at least indicate the model has failed a constraint

`sboTerm`

- SBO = Systems Biology Ontology
  - Independent, international, open effort of BioModels.net Consortium
  - Occupies a space not filled by other ontologies
    - Primarily for describing rate laws and constituents
      1. Classification of rate laws
         - Each term includes a mathematical function definition
      2. Controlled vocabulary for the roles of reaction participants
         - E.g.: “substrate”, “catalyst”, “competitive inhibitor”, etc.
      3. Controlled vocabulary for the roles of quantitative parameters
         - E.g.: “Hill coefficient”

SBO Browser

- http://biomodels.net/SBO



by Melanie Curtot at EBI

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ! kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

Terms are machine-readable

```
[Term]
id: SBO:0000015
name: Briggs-Haldane equation
def: "Rate-law presented in G. E. Briggs and J. B. S. Haldane (1925) A note on the
kinetics of enzyme action, Biochem. J., 19, 339-339. It is a general rate equation
that does not require the restriction of equilibrium of Michaelis-Menten or irreversible
reactions of Van Slyke, but instead make the hypothesis that the complex enzyme-substrate
is in steady-state. Although of the same form than the Henri-Michaelis-Menten equation,
it is semantically different since Km now represents a pseudo-equilibrium constant, and
is equal to the ratio between the rate of consumption of the complex (sum of dissociation
of substrate and generation of product) and the association rate of the enzyme and the
substrate.
is_a: SBO:0000011 ; kinetics of unireactant enzymes
math: <math xmlns="http://www.w3.org/1998/Math/MathML">
      <lambda>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000004">S</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000006">E</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000019">kp</ci></bvar>
      <bvar><ci definitionURL="http://www.biomodels.net/SBO/#SBO:0000008">Km</ci></bvar>
         <apply>
           <divide/>
           <apply>
             <times/><ci>E</ci><ci>kp</ci><ci>S</ci>
           </apply>
           <apply>
             <plus/><ci>Km</ci><ci>S</ci>
           </apply>
         </apply>
       </lambda>
    </math>
```

can insert in SBML

Select SBML constructs have **sboTerm**

Model

id: SId { use="optional" }
name: string { use="optional" }
sboTerm: SBOTerm { use="optional" }
functionDefinition: FunctionDefinition[0..*]
unitDefinition: UnitDefinition[0..*]
compartmentType: CompartmentType[0..*]
speciesType: SpeciesType[0..*]
compartment: Compartment[0..*]
species: Species[0..*]
parameter: Parameter[0..*]
initialAssignment: InitialAssignment[0..*]
rule: Rule[0..*]
constraint: Constraint[0..*]
reaction: Reaction[0..*]
event: Event[0..*]

Initial

symbol: SId
math: Math { namespace="htt
sboTerm: SBOTerm { use="op

Reaction

id: SId
name: string { use="optional" }
reactant: SpeciesReference[0..*]
product: SpeciesReference[0..*]
modifier: ModifierSpeciesReference[0..*]
kineticLaw: KineticLaw { minOccurs="0" maxOccurs="1" }
reversible: boolean { use="optional" default="true" }
fast: boolean { use="optional" default="false" }
sboTerm: SBOTerm { use="optional" }

SimpleSpeciesReference

id: SId { use="optional" }
name: string { use="optional" }
species: SId
sboTerm: SBOTerm { use="optional" }

SpeciesReference

stoichiometry: double { use="optional" default="1" }
stoichiometryMath: StoichiometryMath { use="optional" }

ModifierSpeciesReference

KineticLaw

math: Math { namespace="http://www.w3.org/1998/Math/MathML" }
parameter: Parameter[0..*]
sboTerm: SBOTerm { use="optional" }

StoichiometryMath

math: Math { namespace="http://www.w3.org/1998/Math/MathML" }

Values of **`sboTerm`** for different SBML constructs

SBML Component	SBO Vocabulary	Parent SBO Identifier
Model	modeling framework	SBO:0000004
Reaction	modeling framework	SBO:0000004
Parameter	quantitative parameter	SBO:0000002
SpeciesReference	participant role	SBO:0000003
ModifierSpeciesReference	participant role	SBO:0000003
FunctionDefinition	mathematical expression	SBO:0000064
KineticLaw	mathematical expression	SBO:0000064
InitialAssignment	mathematical expression	SBO:0000064
AlgebraicRule	mathematical expression	SBO:0000064
AssignmentRule	mathematical expression	SBO:0000064
RateRule	mathematical expression	SBO:0000064
Constraint	mathematical expression	SBO:0000064
Event	mathematical expression	SBO:0000064
EventAssignment	mathematical expression	SBO:0000064

Annotations

- Applications can add data to each element in SBML
- Must have only one element enclosing an application's data

```
<annotation>
    <mysim:nodecolors xmlns:mysim="http://www.mysim.org/ns"
        mysim:bgcolor="green"
        mysim:fgcolor="white"/>
</annotation>
```

- Each application should put its data under a separate element

Format for annotations

- Content of <annotation> must be single top-level namespace
- There's a recommended format for
  1. referring to controlled vocabulary terms and database identifiers for describing biological and biochemical entities
  2. describing the creator of a model and its modification history
- Uses RDF & a restricted set of Dublin Core to encode relationships
- Specific BioModels qualifier names
  - http://biomodels.net
- See Sec. 6 in L2V2 specification

••• Break •••

A brief survey of SBML-compatible software

Summary of several software tools: general features

Package	*Win*	*Mac*	*Lin*	*Web*	*Interface*	*Environment*	*ODE*	*Stoch*
MathSBML	x	x	x		Text	Mathematica	x	
COPASI	x	x	x		Forms		x	x
CellDesigner	x	x	x		Graphical, Forms		x	
SBML ODE Solver	x	x	x	x	API & cmd line		x	
Jarnac/ JDesigner	x				Text, Graphical		x	x
SBToolbox	x	x	x		Text	MATLAB	x	x
SimBiology	x	x	x		Text, Graphical	MATLAB	x	x
Dizzy	x	x	x		Text, Forms		x	x

Summary of several software tools: SBML details

Package	*Units?*	*Events?*	*Algebraic Rules?*	*Delays?*	*Functions?*	*Special Features*
MathSBML	x	x	x	x	x	all of Mathematica
COPASI	x				x	sensitivity analysis, parameter scan., MCA, optimization
CellDesigner	x	part.	x		x	parameter scan., sensitivity analysis
SBML ODE Solver	x	part.	x		x	parameter scan., sensitivity analysis
Jarnac/JDesigner		part.	part.	x	x	compact language, MCA
SBToolbox		part.		x	x	optimization, sensitivity analysis, all of MATLAB
SimBiology	x		x		x	full-blown MATLAB product

Where to find the software

- **MathSBML**: http://sbml.org/software/mathsbml
  - Author: Bruce Shapiro (Caltech/JPL/BNMC)
  - Most current documentation is online, not in download!
- **COPASI**: http://copasi.org
  - Authors: P. Mendes, S.Hoops (Virginia Tech), S. Sahle, R.Gauges (EML Heidelberg)
- **SBML ODE Solver**: http://www.tbi.univie.ac.at/~raim/odeSolver/
  - Authors: R. Machne, C. Flamm (U.Vienna)
- **Jarnac, JDesigner**: http://sbw.kgi.edu
  - Author: Herbert Sauro (KGI)
- **SBToolbox**: http://www.sbtoolbox.org/
  - Author: Henning Schmidt (Chalmers, Sweden)
- **CellDesigner**: http://celldesigner.org
- **SimBiology**: http://mathworks.com
- **Dizzy**: http://www.systemsbiology.org/Technology/Data_Visualization_and_Analysis/Dizzy

Closing comments and discussions

SBML = Systems Biology Markup Language

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
- Declarative, not procedural

$$2\,A + B \rightarrow C$$

$$C \rightleftharpoons D + F$$

...

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
- Declarative, not procedural

$$2\,A + B \rightarrow C$$

$$C \rightleftharpoons D + F$$

- Models can also include
  - Compartments (i.e., where chemical substances are located)
  - Mathematical "extras" (assignments, explicit different eq's)
  - Discontinuous events with arbitrary triggers

Current de facto standard

Current de facto standard

- Successful as a model exchange language
  - 100+ software systems worldwide
    - Including commercial developers: MathWorks, TERANODE, etc.
  - Used by international consortia, industry, academia
  - Used in educational materials

Current de facto standard

- Successful as a model exchange language
  - 100+ software systems worldwide
    - Including commercial developers: MathWorks, TERANODE, etc.
  - Used by international consortia, industry, academia
  - Used in educational materials
- Successful as “glue”
  - Coalesced a community of software developers
  - Allows interchange where none existed before
    - Between people, between software, between researchers at many different levels

Current de facto standard

- Successful as a model exchange language
  - 100+ software systems worldwide
    - Including commercial developers: MathWorks, TERANODE, etc.
  - Used by international consortia, industry, academia
  - Used in educational materials
- Successful as "glue"
  - Coalesced a community of software developers
  - Allows interchange where none existed before
    - Between people, between software, between researchers at many different levels
- Of course, SBML isn't without problems

SBML evolution continues

SBML evolution continues

- Current development process is informal
  - SBML Editors seek consensus and created integrated specification
  - Polling & voting by community for major decisions

SBML evolution continues

- Current development process is informal
  - SBML Editors seek consensus and created integrated specification
  - Polling & voting by community for major decisions
- New SBML process coming
  - Borrow ideas from World Wide Web Consortium & other orgs
  - Architectural board with larger membership
  - SBML Editors voted in for limited terms
  - Implement better-defined, regimented process
    - Calls for proposals, etc.
    - Voting, etc.

Upcoming SBML meetings

- SBML Forum Meeting 2006 Oct. 12-13 (after ICSB, in Tokyo)
  - See http://sbml.org/workshops for more info
- SBML Hackathon, June 2007, University of Newcastle (UK)
- SBML Forum Meeting 2007, Oct. 5-6, Long Beach, California (USA)
  - See http://icbs-2007.org

Keep informed!

- Join **sbml-announce**@caltech.edu (low-volume, broadcast-only)
- Join **sbml-discuss** (higher volume, discussions & debates)
- Join **libsbml-discuss** (discussions & help about libsbml)
- See **http://sbml.org/forums**

Thank you for attending!

- And a big thanks to our funding agencies:
  - NIH National Institute of General Medical Sciences (USA)
  - National Science Foundation (USA)
  - International Joint Research Program of NEDO (Japan)
  - JST ERATO Kitano Symbiotic Systems Project (Japan)
  - JST ERATO-SORST Program (Japan)
  - Japanese Ministry of Agriculture
  - Japanese Ministry of Education, Culture, Sports, Science and Tech.
  - BBSRC e-Science Initiative (UK)
  - DARPA IPTO Bio-Computation Program (USA)
  - Air Force Office of Scientific Research (USA)
  - STRI, University of Hertfordshire (UK)
  - Beckman Institute, Caltech