SBML-shorthand specification, Version 3.1.1

Darren J Wilkinson
email: darren.j.wilkinson@durham.ac.uk

February 24, 2024

Contents

1 Introduction

1.1 Overview

SBML-shorthand is a language for describing systems biology models. It is designed primarily for translation (compilation) into SBML. It is not associated with any particular modelling or simulation tool. SBML itself is not intended to be read by humans or written "by hand" — SBML-shorthand is. Note that SBML-shorthand is not intended to cover every feature in SBML — just the most commonly used aspects.

1.2 Version

The SBML-shorthand specification has a version number of the form $a.b.c$. The a refers to the SBML Level that the shorthand is a target for. The b refers to the SBML Version that the shorthand is a target for. The c refers to the version number of the SBML-shorthand specification for that particular SBML Level/Version combination. So, 3.1.1 is the first draft of a SBML-shorthand specification aimed at SBML Level 3, Version 1 (core).

1.3 Example

The following text is a complete SBML-shorthand description of a simple model for Michaelis-Menten enzyme kinetics.

```
@model:3.3.1=MichaelisMentenKinetics "Michaelis-Menten Kinetics"
@compartments
 cell=1
@species
 cell:Substrate=1000
 cell:Enzyme=100
 cell:Complex=0
 cell:Product=0
@parameters
 k1=1
 k1r=2
@reactions
@rr=SubstrateEnzymeBinding "Substrate-enzyme binding"
 Substrate+Enzyme -> Complex
 k1*Substrate*Enzyme-k1r*Complex
@r=Conversion
 Complex -> Product + Enzyme
 k2*Complex : k2=3
```

The "compiled" SBML corresponding to this shorthand description is given below.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1">
  <model id="MichaelisMentenKinetics" name="Michaelis-Menten Kinetics">
    <listOfCompartments>
      <compartment id="cell" size="1" constant="true"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="Substrate" compartment="cell" initialAmount="1000" hasOnlySubstanceUnits="false"
               boundaryCondition="false" constant="false"/>
      <species id="Enzyme" compartment="cell" initialAmount="100" hasOnlySubstanceUnits="false"
               boundaryCondition="false" constant="false"/>
      <species id="Complex" compartment="cell" initialAmount="0" hasOnlySubstanceUnits="false"
               boundaryCondition="false" constant="false"/>
```

```
      <species id="Product" compartment="cell" initialAmount="0" hasOnlySubstanceUnits="false"
               boundaryCondition="false" constant="false"/>
    </listOfSpecies>
    <listOfParameters>
      <parameter id="k1" value="1" constant="true"/>
      <parameter id="k1r" value="2" constant="true"/>
    </listOfParameters>
    <listOfReactions>
      <reaction id="SubstrateEnzymeBinding" name="Substrate-enzyme binding" reversible="true" fast="false">
        <listOfReactants>
          <speciesReference species="Substrate" stoichiometry="1" constant="false"/>
          <speciesReference species="Enzyme" stoichiometry="1" constant="false"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="Complex" stoichiometry="1" constant="false"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <minus/>
              <apply>
                <times/>
                <ci> k1 </ci>
                <ci> Substrate </ci>
                <ci> Enzyme </ci>
              </apply>
              <apply>
                <times/>
                <ci> k1r </ci>
                <ci> Complex </ci>
              </apply>
            </apply>
          </math>
        </kineticLaw>
      </reaction>
      <reaction id="Conversion" reversible="false" fast="false">
        <listOfReactants>
          <speciesReference species="Complex" stoichiometry="1" constant="false"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="Product" stoichiometry="1" constant="false"/>
          <speciesReference species="Enzyme" stoichiometry="1" constant="false"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci> k2 </ci>
              <ci> Complex </ci>
            </apply>
          </math>
          <listOfLocalParameters>
            <localParameter id="k2" value="3"/>
          </listOfLocalParameters>
        </kineticLaw>
      </reaction>
    </listOfReactions>
  </model>
</sbml>
```

It is hopefully immediately transparent that the SBML-shorthand notation is much easier to read and write than the corresponding SBML, but that there is a simple one-one correspondence between the model elements.

2 Specification

2.1 Basic structure

The description format is plain ASCII text. The suggested file extension is `.mod`, but this is not required. All whitespace other than carriage returns is insignificant (unless it is contained within a quoted "name" element). Carriage returns are significant. The description is case-sensitive. Blank lines are ignored. The comment character is `#` - all text from a `#` to the end of the line is ignored.

2.2 Model

The model description must begin with the characters `@model:3.3.1=` (the 3.3.1 corresponds to the version number of this specification). The text following the `=` on the first line is the model id. An optional model name may also be specified, following the id, enclosed in double-quotes.

The model line may be followed with an additional line specifying global model units. For example, the SBML-shorthand

```
@model:3.3.1=mymodel "My model"
 s=substance, t=second, v=litre
```

would be translated to

```
  <model id="MyName" name="My Model" substanceUnits="substance"
                     timeUnits="second" volumeUnits="litre">
```

The model attributes `substanceUnits`, `timeUnits`, `volumeUnits`, `areaUnits`, `lengthUnits`, `extentUnits`, and `conversionFactor` are represented by the letters `s`, `t`, `v`, `a`, `l`, `e` and `c`, respectively.

The model is completed with the specification of the 5 sections, `@units`, `@compartments`, `@species`, `@parameters`, `@rules`, `@reactions` and `@events`, corresponding to the SBML sections, `<listOfUnitDefinitions>`, `<listOfCompartments>`, `<listOfSpecies>`, `<listOfParame` `<listOfRules>`, `<listOfReactions>` and `<listOfEvents>`, respectively. The sections must occur in the stated order. Sections are optional, but if present, may not be empty. These are the only sections covered by this specification. Subsequent specifications will contain additional sections.

2.3 Units

The format of the individual sections will be explained mainly by example. The following SBML-shorthand

```
@units
 substance=item
 mmls=mole:s=-3; litre:e=-1; second:e=-1
```

would be translated to

```
    <listOfUnitDefinitions>
      <unitDefinition id="substance" name="item (substance default)">
        <listOfUnits>
          <unit kind="item" exponent="1" scale="0" multiplier="1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="mmls">
```

```
      <listOfUnits>
        <unit kind="mole" exponent="1" scale="-3" multiplier="1"/>
        <unit kind="litre" exponent="-1" scale="0" multiplier="1"/>
        <unit kind="second" exponent="-1" scale="0" multiplier="1"/>
      </listOfUnits>
    </unitDefinition>
  </listOfUnitDefinitions>
```

The unit attributes `exponent`, `multiplier`, and `scale` are denoted by the letters `e`, `m` and `s`, respectively. Note that as there is (currently) no way to refer to units elsewhere in SBML-shorthand, the only function for this section is to re-define the globally defined units declared in the model section.

2.4 Compartments

The following SBML-shorthand

```
@compartments
 cell=1
 cytoplasm=0.8
 nucleus=0.1
 mito "Mitochondria"
 cell2
```

would be translated to

```
  <listOfCompartments>
    <compartment id="cell" size="1" constant="true"/>
    <compartment id="cytoplasm" size="0.8" constant="true"/>
    <compartment id="nucleus" size="0.1" constant="true"/>
    <compartment id="mito" name="Mitochondria" constant="true"/>
    <compartment id="cell2" constant="true"/>
  </listOfCompartments>
```

Note that if a `name` attribute is to be specified, it should be specified at the end of the line in double-quotes. This is true for other SBML elements too.

2.5 Species

The following shorthand

```
@species
 cell:Gene = 10b "The Gene"
 cell:P2=0
 cell:S1=100 s
 cell:[S2]=20 sc
 cell:[S3]=1000 bc
 mito:S4=0 b
```

would be translated to

```
  <listOfSpecies>
    <species id="Gene" name="The Gene" compartment="cell" initialAmount="10"
             hasOnlySubstanceUnits="false" boundaryCondition="true" constant="false"/>
    <species id="P2" compartment="cell" initialAmount="0"
             hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"/>
    <species id="S1" compartment="cell" initialAmount="100"
```

```
            hasOnlySubstanceUnits="true" boundaryCondition="false" constant="false"/>
    <species id="S2" compartment="cell" initialConcentration="20"
            hasOnlySubstanceUnits="true" boundaryCondition="false" constant="true"/>
    <species id="S3" compartment="cell" initialConcentration="1000"
            hasOnlySubstanceUnits="false" boundaryCondition="true" constant="true"/>
    <species id="S4" compartment="mito" initialAmount="0"
            hasOnlySubstanceUnits="false" boundaryCondition="true" constant="false"/>
  </listOfSpecies>
```

Compartments are compulsory. An `initialConcentration` (as opposed to an `initialAmount`) is flagged by enclosing the species `id` in brackets. The boolean attributes `hasOnlySubstanceUnits`, `boundaryCondition` and `constant` can be set to `true` by appending the letters `s`, `b` and `c`, respectively. The order of the flags is not important.

2.6 Parameters

The section

```
@parameters
 k1=1  # a comment
 k2 = 10v "K2"
```

would be translated to

```
  <listOfParameters>
    <parameter id="k1" value="1" constant="true"/>
    <parameter id="k2" name="K2" value="10" constant="false"/>
  </listOfParameters>
```

2.7 Rules

Currently only assignment and rate rules are supported. Assignment rules are supported using a fairly obvious notation. Rate rules can be specified by prefixing the rule with `@rate:`. So

```
@rules
 PTot = P + 2*P2
 v = 1 + 0.5*t
 @rate:v2=1
```

would be translated to

```
  <listOfRules>
    <assignmentRule variable="PTot">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <plus/>
          <ci> P </ci>
          <apply>
            <times/>
            <cn type="integer"> 2 </cn>
            <ci> P2 </ci>
          </apply>
        </apply>
      </math>
    </assignmentRule>
    <assignmentRule variable="v">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
```

```
        <apply>
          <plus/>
          <cn type="integer"> 1 </cn>
          <apply>
            <times/>
            <cn> 0.5 </cn>
            <csymbol encoding="text"
                     definitionURL="http://www.sbml.org/sbml/symbols/time">
                     t </csymbol>
          </apply>
        </apply>
      </math>
    </assignmentRule>
    <rateRule variable="v2">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <cn type="integer"> 1 </cn>
      </math>
    </rateRule>
  </listOfRules>
```

2.8 Reactions

Each reaction is specified by exactly two or three lines of text. The first line declares the reaction name and whether the reaction is reversible (`@rr=` for reversible and `@r=` otherwise). The second line specifies the reaction itself using a fairly standard notation. If a modifier is to be included, it should occur at the end, separated from the rest of the reaction by a `:`. The (optional) third line specifies the full rate law for the kinetics. If local parameters are used, they should be declared on the same line in a comma-separated list, separated from the rate law using a `:`

So, for example,

```
@reactions
@r=RepressionBinding "Repression Binding"
 Gene + 2P -> P2Gene
 k2*Gene
@rr=Reverse
 P2Gene -> Gene+2P
 k1r*P2Gene : k1r=1,k2=3
@r=NoKL
 Harry->Jim : Fred
@r=Test
 Fred -> Fred2
 k4*Fred : k4=1
```

would translate to

```
  <listOfReactions>
    <reaction id="RepressionBinding" name="Repression Binding" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference species="Gene" stoichiometry="1" constant="false"/>
        <speciesReference species="P" stoichiometry="2" constant="false"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="P2Gene" stoichiometry="1" constant="false"/>
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
```

```
            <times/>
            <ci> k2 </ci>
            <ci> Gene </ci>
          </apply>
        </math>
      </kineticLaw>
    </reaction>
    <reaction id="Reverse" reversible="true" fast="false">
      <listOfReactants>
        <speciesReference species="P2Gene" stoichiometry="1" constant="false"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="Gene" stoichiometry="1" constant="false"/>
        <speciesReference species="P" stoichiometry="2" constant="false"/>
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <times/>
            <ci> k1r </ci>
            <ci> P2Gene </ci>
          </apply>
        </math>
        <listOfLocalParameters>
          <localParameter id="k1r" value="1"/>
          <localParameter id="k2" value="3"/>
        </listOfLocalParameters>
      </kineticLaw>
    </reaction>
    <reaction id="NoKL" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference species="Harry" stoichiometry="1" constant="false"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="Jim" stoichiometry="1" constant="false"/>
      </listOfProducts>
      <listOfModifiers>
        <modifierSpeciesReference species="Fred"/>
      </listOfModifiers>
    </reaction>
    <reaction id="Test" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference species="Fred" stoichiometry="1" constant="false"/>
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="Fred2" stoichiometry="1" constant="false"/>
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <times/>
            <ci> k4 </ci>
            <ci> Fred </ci>
          </apply>
        </math>
        <listOfLocalParameters>
          <localParameter id="k4" value="1"/>
        </listOfLocalParameters>
      </kineticLaw>
    </reaction>
  </listOfReactions>
```

2.9 Events

The following SBML-shorthand

```
@events
 reset= t>=25 : P=100;P2=0 "Reset event"
 flush= P>10 ; 20 : P=0
```

would be translated to

```
<listOfEvents>
  <event id="reset" name="Reset event">
    <trigger>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <geq/>
          <csymbol encoding="text" definitionURL="http://www.sbml.org/sbml/symbols/time"> t </csymbol>
          <cn type="integer"> 25 </cn>
        </apply>
      </math>
    </trigger>
    <listOfEventAssignments>
      <eventAssignment variable="P">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <cn type="integer"> 100 </cn>
        </math>
      </eventAssignment>
      <eventAssignment variable="P2">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <cn type="integer"> 0 </cn>
        </math>
      </eventAssignment>
    </listOfEventAssignments>
  </event>
  <event id="flush" useValuesFromTriggerTime="true">
    <trigger>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <gt/>
          <ci> P </ci>
          <cn type="integer"> 10 </cn>
        </apply>
      </math>
    </trigger>
    <delay>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <cn type="integer"> 20 </cn>
      </math>
    </delay>
    <listOfEventAssignments>
      <eventAssignment variable="P">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <cn type="integer"> 0 </cn>
        </math>
      </eventAssignment>
    </listOfEventAssignments>
  </event>
</listOfEvents>
```

So, the event assignments occur in a semi-colon-separated list after a colon, and the optional delay element can be specified after the trigger condition, separated from it with a semi-colon. If the variable `t` occurs in the trigger condition, it is interpreted as the SBML time `csymbol`.

3 Tool support

The author has written a reference implementation of a SBML-shorthand to SBML compiler, written in Python. This can be used either as a simple command-line translator, or by Python programmers as a Python module. Documentation for this module is available elsewhere. The Python module requires libSBML (and requires that libSBML is built with Python bindings). A reference implementation of a SBML to SBML-shorthand translator is also available.

See the SBML-shorthand web site for further details.

4 Acknowledgements

Several people have contributed to SBML-shorthand and the associated translation tools, including Jeremy Purvis, Carole Proctor, Mark Muldoon, and Lukas Endler.

5 Links

SBML-shorthand	`www.staff.ncl.ac.uk/d.j.wilkinson/software/sbml-sh/`
SBML	`sbml.org`
Python	`www.python.org`