Systems Biology Markup Language (SBML) Level 1: Structures and Facilities for Basic Model Definitions

Michael Hucka, Andrew Finney, Herbert Sauro, Hamid Bolouri
{mhucka,afinney,hsauro,hbolouri}@cds.caltech.edu
Systems Biology Workbench Development Group
JST ERATO Kitano Symbiotic Systems Project
Control and Dynamical Systems, MC 107-81
California Institute of Technology, Pasadena, CA 91125, USA
http://www.cds.caltech.edu/erato

Principal Investigators: John Doyle and Hiroaki Kitano

SBML Level 1,

1 Introduction

We present the Systems Biology Markup Language (SBML) Level 1, Version 2, a description language

1.2 Scope and Limitations

SBML Level 1 is meant to support non-spatial biochemical models and the kinds of operations that are

reference is “http://www.mysim.org/ns” and the prefix is mysim. An example of an annotation might then be as follows:

```
...
<annotation  xmlns:mysim="http://www.mysim.org/ns">
      <mysim:nodecolors  mysim:bgcolor="green"  mysim:fgcolor="white"/>
      <mysim:timestamp>2000-12-18  18:31  PST</mysim:timestamp>
</annotation>
...
```

The namespace prefix mysim is used to qualify the XML elements mysim:nodecolors and mysim:timestamp;

```
letter   ::= 'a'..'z','A'..'Z'
digit    ::= '0'..'9'
name     ::= ( letter | '_' ) ( letter | digit | '_' )*
```

The namespace rules described here provide a clean transition path to future levels of SBML, when submodels are introduced (Section 0.1). Submodels will provide the ability to compose one model from a collection of other models. This capability will have to be built on top of SBML Level 1's namespace organization. A straightforward approach to handling namespaces is to make each submodel's space be private. The rules governing namespaces within a submodel can simply be the Level 1 namespace rule described here, with

The optional fie6d`outside` of type `SName` can be used to express containment re6ationships between compartments. If present, the value of `outside`

(which, if present, signifies that the default units of quantity should be used—see Section 4.2), or a new unit name defined by a unit definition in the enclosing `Model`. If absent, the units default to the value set by the built-in "`substance`".

The field `compartment` is a string that names the compartment within which the species is located. The

“substance”, “time”, or ” volume

Figure 9: *The definition of* `Rule` *and derived types.*

Figure 10: *The definitions of* `Reaction`*,* `KineticLaw` *and* `SpeciesReference`*.*

`reactant`, `product`

```
</listOfReactants>
<listOfProducts>
    <speciesReference
```

```
    <species name="s1" compartment="cell" initialAmount="1"/>
    <species name="s2" compartment="cell" initialAmount="1"/>
</listOfSpecies>
<listOfParameters>
```

component is a functional unit that may correspond to a physical compartment or simply a convenient mod-

```
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="(_|[a-z]|[A-Z])(_|[a-z]|[A-Z]|[0-9])*"/>
  </xsd:restriction>
</xsd:simpleType>
<!--The definition of SBase follows.-->
<xsd:complexType name="SBase" abstract="true">
  <xsd:annotation>
    <xsd:documentation>The SBase type is the base type of all main
    components in SBML.  It supports attaching notes and annotations to
    components.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="notes" minOccurs="0">
      <xsd:complexType>
```

```
</xsd:simpleType>
<!--The definition of Unit follows.-->
<xsd:complexType name="Unit">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="kind" type="UnitKind" use="required"/>
      <xsd:attribute name="exponent" type="xsd:integer" default="1"/>
      <xsd:attribute name="scale" type="xsd:integer" default="0"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--The definition of UnitDefinition follows.-->
<xsd:complexType name="UnitDefinition">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:sequence>
        <xsd:element name="listOfUnits" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="unit" type="Unit" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="name" type="SName" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--The definition of Compartment follows.-->
<xsd:complexType name="Compartment">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="name" type="SName" use="required"/>
      <xsd:attribute name="volume" type="xsd:double" default="1"/>
      <xsd:attribute name="units" type="SName" use="optional"/>
      <xsd:attribute name="outside" type="SName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--The definition of Species follows.-->
<xsd:complexType name="Species">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="name" type="SName" use="required"/>
      <xsd:attribute name="compartmen5(type="Unit)-SName" use="optional"/>
      <xsd:attribute name="initialAmoun5(type="Unit)-xsd:double" use="required"/>
      <xsd:attribute name="units" type="SName" use="optional"/>
      <xsd:attribute name="boundaryCondition(type="Unit)-xsd:boolean" use="optional" default="false"/>
      <xsd:attribute name="charge" type="xsd:integer" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--The definition of Parameter follows.-->
<xsd:complexType name="Parameter">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="name" use="required"/>
      <xsd:attribute name="value" type="xsd:double" use="optional"/>
      <xsd:attribute name="units" type="SName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--The definition of Rule follows. -->
<xsd:simpleType name="RuleType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="scalar"/>
    <xsd:enumeration value="rate"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:sequence></xsd:complexType></xsd:element><xsd:element name="listOfParameters" minOccurs="0"><xs
```

Name	Arguments	Meaning	Formula
massi	S_i, k	Irreversible Mass Action Kinetics	$v = k \prod_i S_i$
massr	S_i, P_j, k_1, k_2	Reve.978rsible Mass Action Kinetics	$v = k_1 \prod_i S_i - k_2 \prod_j P_j$
uui	S, V_m, K_m	Irreversible Simple Michaelis-Menten	$v = \frac{V_m S}{}$

Name	Arguments	Meaning	Formula
usii			

Name	Arguments	Meaning	Formula
uctr	S, P, A_c, V_f,		

Symbol	Meaning
α	

References