1 Introduction

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

tionally omitted from the language definition. Future software tools will undoubtedly require the evolution of SBML; we expect that subsequent releases of SBML (termed *levels*) will add additional structures and facilities currently missing from Level 1, once the simulation community gains experience with the current language definition. In Section 6.1, we discuss extensions that will likely be included in SBML Level 2.

The definition of the model description language presented here does not specify *how* programs should communicate or read/write SBML. We assume that for a simulation program to communicate a model encoded in SBML, the program will have to translate its internal data structures to and from SBML, use a suitable transmission medium and protocol, etc., but these issues are outside of the scope of this document.

1.3 Notational Conventions

SBML is intended to be a common XML-based format for encoding systems biology models in a simple form that software tools can use as an exchange format. However, for easier communication to human readers, we define SBML using a graphical notation based upon UML, the Unified Modeling Language (Eriksson and

The meaning of each component is as follows:

Unit definition: A name for a unit used in the expression of quantities in a model. Units may be supplied in a number of contexts in an SBML model, and it is convenient to have a facility for both setting default units and for allowing combinations of units to be given abbreviated names.

Compartment: A container of finite volume for substances. In SBML Level 1, a compartment is primarily

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>
...
```

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

Figure 3: *The definition of the type SName, expressed in the variant of Extended Backus-Naur Form (EBNF) used by the XML 1.0 specification (Bray et al., 2000). The characters (and) are used for grouping, and the character * signifies "zero or more times" the immediately-preceding term.*

3.4 Component Names and Namespaces in SBML

4 SBML Components

4.2 Unit Definitions

The optional boolean field `boundaryCondition` determines whether the amount of the species is fixed or variable over the course of a simulation. The value of `boundaryCondition` defaults to "`false`", indicating that by default, the amount is not fixed. If the amount of a species is defined as being fixed, it implies that some external mechanism maintains a constant quantity in the compartment throughout the course of a reaction. (The term *boundary condition* alludes to the role of this constraint in a simulation.)

The optional field `charge` is an integer indicating the charge on the species (in terms of electrons, not the SI unit Coulombs). This may be useful when the species involved is a charged ion such as calcium (Ca^{2+}).

The fol1unda87-17.53ns221xampl(e)-42516(shda87-s53ns2228(y)w)(o)-2JET1001-2330562417.532cmBT0.890.940.28k00.890.9

that are defined within a reaction are local to the particular reaction and (within that reaction) *override* any global parameters having the same names. (See Section 3.4 for further details.)

The following is an example of parameters defined at the `Model` level:

```
<model>
    ...
    <listOfSpecies>
        ...
    </listOfSpecies>
    <listOfParameters>
        <parameter name="Km1" value="2.3" units="second"/>
        <parameter name="Km2" value="10.7" units="second"/>
    </listOfParameters>
    <listOfReactions>
        ...
    </listOfReactions>
    ...
</model>
```

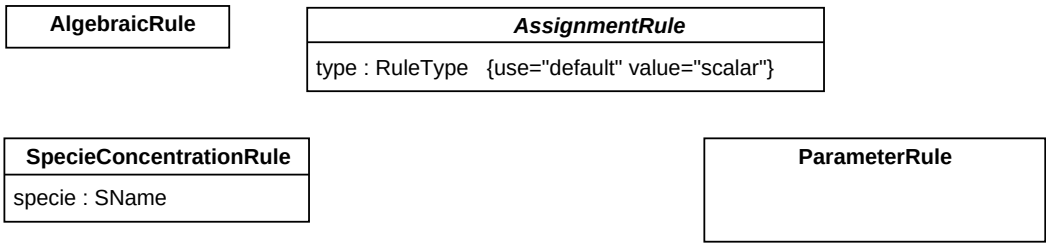
AlgebraicRule
AssignmentRule
type : RuleType {use="default" value="scalar"}
SpecieConcentrationRule
specie : SName
ParameterRule

Figure 9:

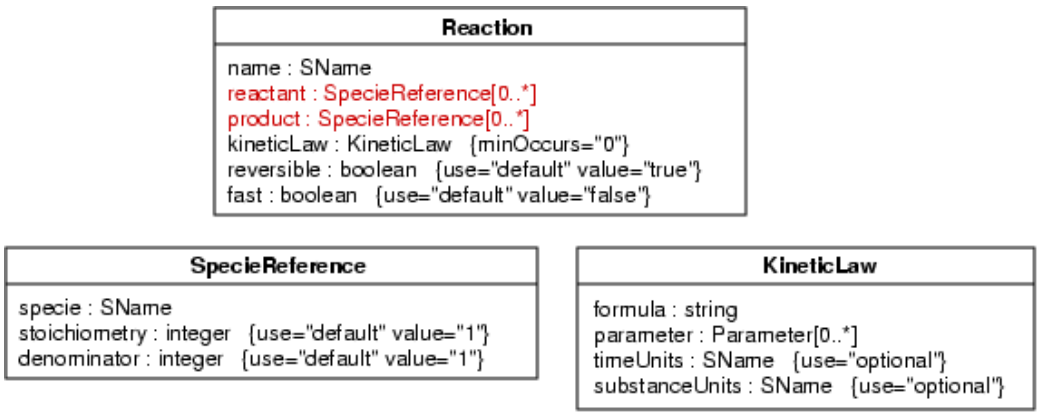

Figure 10: *The definitions of* `Reaction`*,* `KineticLaw` *and* `SpeciesReference`*.*

The field `fast` is another boolean attribute in the `Reaction` data structure; a value of "`true`" signifies

4.7.2 KineticLaw

A kineticLaw

Hugh Spence, Joerg Stelling, Kouichi Takahashi, Masaru Tomita, and John Wagner.

We are indebted to Daniel Lucio of the Virtual Cell group for generating the XML Schema of SBML Level 1 presented in Appendix B.

```
            <xsd:documentation>
               SName type used for expressing
               names of components in a model
            </xsd:documentation>
          </xsd:annotation>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="(_|[a-z]|[A-Z])(_|[a-z]|[A-Z]|[0-9])*"/>
          </xsd:restriction>
        </xsd:simpleType>
        <!-- SBase -->
        <xsd:complexType name="SBase" abstract="true">
          <xsd:annotation>
            <xsd:documentation>
              Deaigned to allow a modeler or a pacckage to attach
              information to each component.
            </xsd:documentation>
          </xsd:annotation>
          <xsd:sequence>
            <xsd:element name="notes" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:any namespace="http://www.w3.org/1999/xhtml"
                           processContents="skip" maxOccurs="unbounded"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="annotation" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:any processContents="skip" maxOccurs="unbounded"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
        <!-- LisOfParameter  -->
        <xsd:element name="listOfParameters">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="parameter" type="Parameter"
                           maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <!-- Model -->
        <xsd:complexType name="Model">
          <xsd:complexContent>
            <xsd:extension base="SBase">
              <xsd:sequence>
                <xsd:element name="listOfUnitDefinitions" minOccurs="0">
                  <xsd:complexType>
                    <xsd:sequence>
                      <xsd:element name="unitDefinition" type="UnitDefinition"
                                   maxOccurs="unbounded"/>
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
                <xsd:element name="listOfCompartments414-9.315Td[(</xsd:complexType>)]TJ-9.415-9.315Td[(</xsd:element>)]T
  </xsd:sequence>
</xsd:complexType>
```

```
            </xsd:complexType>
          </xsd:element>
          <xsd:element ref="listOfParameters" minOccurs="0"/>
          <xsd:element name="listOfRules" minOccurs="0">
            <">
            <"hoiceles"nOccuunboundedters
              <xsd:element naalgebraicstOfules" minOccurs="0"/>

  <xsd:element naspecieConcelerattersionstOfulesMinOccurs="0"/><xsd:element naptOfParamstOfules" minOccurs="0"/></"h
</xsd:element>
<xsd:element name="liseactiontersss="0">
  <">
  <sequence="0">
    <xsd:element nareactionules"nOccuunboundeds="0"/></sequence="0"></"sd:complexType>
  </xsd:element></sequence="0">
  <attributement na525(ules")-t.4terse="SN25(ules")-use="optionals="0"/></extension"0"/></"sd:comContelement></"sd:c
  <"sd:complexment naUnitDefinitio5(mi)"5-2.36cm6.415-6.315Td[(<"sd:comContelement>)]TJ9.415-9.315Td[(<xxtensionment
    <">
    <sequence="0">
      <xsd:element naunitules""nOccuunboundeds="0"/></sequence="0"></xsd:complexType>
    </xsd:element></sequence="0">
    <attributement na525(ules")-t.4terse="SN25(ules")-use="requireds="0"/></extension"0"/></"sd:comContelement></"sd

    <"sd:comContelement>
      <xxtensionment<attributement na525(ules")-t.4terse="SN25(ules")-use="requireds="0"/>
      <attributement navolumeules"<doubl(ules")-use="defaultules"
      <attributement naunitRules"ules"
      <attributement naoutsid(ules"6-t.41="SN25(ules")-use="optionals="0"/>)-TJ-4.415-6.315Td[(</extension"0"/>)-TJ-
      <"sd:complexment naSpecie"5-2.36cm6.415-9.315Td[(<"sd:comContelement>)]TJ9.415-9.315Td[(<xxtensionment)-base="
      <attributement nainitialA5tersountules"<doubl(ules")-use="requireds="0"/>
      <attributement naunitRules"ules"
```

```
<!-- Parameter -->
<xsd:complexType name="Parameter">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="name" type="SName" use="required"/>
      <xsd:attribute name="value" type="xsd:double" use="required"/>
      <xsd:attribute name="units" type="SName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- UnitKind -->
<xsd:simpleType name="UnitKind">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ampere"/>
    <xsd:enumeration value="becquerel"/>
    <xsd:enumeration value="candela"/>
    <xsd:enumeration value="celsius"/>
    <xsd:enumeration value="coulomb"/>
    <xsd:enumeration value="dimensionless"/>
    <xsd:enumeration value="farad"/>
    <xsd:enumeration value="gram"/>
    <xsd:enumeration value="gray"/>
    <xsd:enumeration value="henry"/>
    <xsd:enumeration value="hertz"/>
    <xsd:enumeration value="item"/>
    <xsd:enumeration value="joule"/>
    <xsd:enumeration value="katal"/>
    <xsd:enumeration value="kelvin"/>
    <xsd:enumeration value="kilogram"/>
    <xsd:enumeration value="liter"/>
    <xsd:enumeration value="litre"/>
    <xsd:enumeration value="lumen"/>
    <xsd:enumeration value="lux"/>
    <xsd:enumeration value="meter"/>
    <xsd:enumeration value="metre"/>
    <xsd:enumeration value="mole"/>
    <xsd:enumeration value="newton"/>
    <xsd:enumeration value="ohm"/>
    <xsd:enumeration value="pascal"/>
    <xsd:enumeration value="radian"/>
    <xsd:enumeration value="second"/>
    <xsd:enumeration value="siemens"/>
    <xsd:enumeration value="sievert"/>
    <xsd:enumeration value="steradian"/>
    <xsd:enumeration value="tesla"/>
    <xsd:enumeration value="volt"/>
    <xsd:enumeration value="watt"/>
    <xsd:enumeration value="weber"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- Unit -->
<xsd:complexType name="Unit">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:attribute name="kind" type="UnitKind"on base="SBase">
```

```
    <xsd:extension base="SBase">
      <xsd:attribute name="formula" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AlgebraicRule">
  <xsd:complexContent>
    <xsd:extension base="Rule"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AssignmentRule" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="Rule">
      <xsd:attribute name="type" type="RuleType" use="default" value="scalar"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CompartmentVolumeRule">
  <xsd:complexContent>
    <xsd:extension base="AssignmentRule">
      <xsd:attribute name="compartment" type="SName" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SpecieConcentrationRule">
  <xsd:complexContent>
    <xsd:extension base="AssignmentRule">
      <xsd:attribute name="specie" type="SName" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ParameterRule">
  <xsd:complexContent>
    <xsd:extension base="AssignmentRule">
      <xsd:attribute name="name" type="SName" use="required"/>
      <xsd:attribute name="units" type="SName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- Reaction --="SName" use="optio6/ement name="specieReference">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="SBase">
        <xsd:attribute name="specie" type="xsd:string" use="required"/>
        <xsd:attribute name="stoichiometry" type="xsd:integer" use="default" value="1"/>
        <xsd:attribute name="denominator" type="xsd:integer" use="default" value="1"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:e/ement>
<xsd:complexType name="Reaction">
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:sequence>
        <xsd:e/ement name="listOfReactants">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:e/ement ref="specieReference" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:e/ement>
```

C Predefined Functions in SBML

Table 6 lists the basic mathematical functions that are defined in SBML Level 1 at this time.

Name	Args.	Formula or Meaning	Argument Constraints	Result Constraints
abs	x	absolute value of x		
acos	x	arc cosine of x in radians	$-1.0 \leq x \leq 1.0$	$0 \leq acos(x) \leq \pi$
asin	x	arc sine of x in radians	$-1.0 \leq x \leq 1.0$	$-\pi/2 \leq asin(x) \leq \pi/2$
atan	x	arc tangent of x in radians		$-\pi/2 \leq atan(x) \leq \pi/2$
ceil	x			

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
uctr	S, P, A_c, V_f, V_r, K_{ms}, K_{mp}, K_a	Catalytic Activation (Reversible)	$v = \frac{V_f S/K_{ms} - V_r P/K_{mp}}{1 + K_a/A_c + (S/K_{ms} + P/K_{mp})(1 + K_a/A_c)}$
umai	S, A_c, V, K		

Symbol	Meaning
α	E□ect of S and P on binding of M (if $M < 1$, M is inhibitor; if $M > 1$, M is activator)
A	First substrate in two substrate reaction
A_c	Activator
B	Secbnd substrate in two substrate reaction
I	Inhibitor
K	

K

References