Systems Biology Markup Language (SBML) Level 1:

3.2 Guidelines for the Use of the `annotations` Field in `SBase`

The `annotations` field in the definition of `SBase` is formally unconstrained in order that software developers may attach any information they need to different components in an SBML model. However, it is important that this facility not be misused accidentally. In particular, it is critical that information essential to a model definition is *not* stored in `annotations`. Parameter values, functional dependencies between model components, etc., should not be recorded as annotations.

Here are examples of the kinds of data that may be appropriately stored in `annotations`: (a) Information

abs	cos	hillr	not	ppbr	ta9	ucii	umar	usii	volume
acos	exp	isouur	or	si9	umai	ucir	umi	usir	xor
and	floor	log	ordbbr	sqr	umar	ucti	unii	uuci	
asi9	hilli	log10	ordbur	sqrt	uai	uctr	unir	uucr	
ata9	hillmmr	massi	ordubr	substa9ce	ualii	uhmi	uuhr	uui	
ceil	hillmr	massr	pow	time	uar	uhmr	umr	uur	

Table 2: *The reserved names i9 SBML Level 1.*

The set of rules above ca9 enable software packages using either local or global namespaces to exchange

```
<model name="My_Model">
    <listOfUnitDefinitions>
        ...
    </listOfUnitDefinitions>
    <listOfCompartments>
        ...
    </listOfCompartments>
    <listOfSpecies>
        ...
    </listOfSpecies>
    <listOfParameters>
        ...
    </listOfParameters>
    <listOfRules>
        ...
    </listOfRules>
    <listOfReactions>
        ...
    </listOfReactions>
</model>
```

Readers may wonder about the motivations for the `listOf_____s` notation. A simpler approach to creating the lists of components would be to place them all directly at the top level under `<model> ... </model>`. We chose instead to group them within XML elements named after `listOf_____s`, because we believe this helps organize the components and makes visual reading of model de nitions easier.

4.2 Unit De nitions

Units may be supplied in a number of contexts in an SBML model. A facility for de ning units is convenient to have so that combinations of units can be given abbreviated names. This is the motivation behind the `UnitDefinition` data structure, whose de nition is shown in Figure 5.

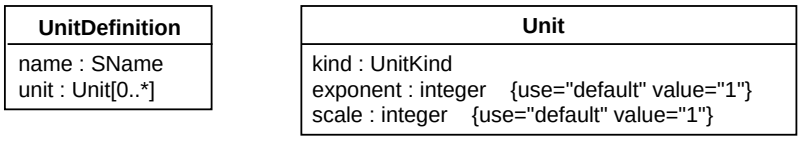

Figure 5: *The de nition of* `UnitDefinition`.

A unit de nition consists of a `name` eld of type `SName` and an optional list of structures of type `Unit`. The approach to de ning units in SBML is compositional; for example, *meter secondTd[(s)] 96.746 Tf 20.87 3.631*

4.3 Compartments

A `Compartment` represents a bounded container in which species are located. The definition of `Compartment` is shown in Figure 6.

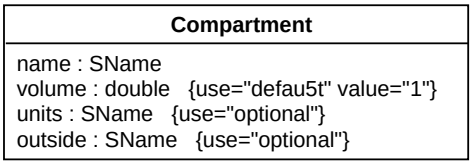

Figure 6: *The definition of `Compartment`. Fields inherited from `SBase`*

4.4 Species

The term *species* refers to entities that take part in reactions. These include simple ions (e.g., protons, calcium), simple molecules (e.g., glucose, ATP), and large molecules (e.g., RNA, polysaccharides, and proteins). The Specie data structure is intended to represent these entities. Its de nition is shown in Figure 7.

Specie
name : SName compartment : SName initialAmount : double units : SName {use="optional"} boundaryCondition : boolean {use="default" value="false"} charge : integer {use="optional"}

Figure 7: *The de nition of Specie. As usual, elds inherited from SBase are omitted here but are assumed.*

Specie has a name eld of type SName. The eld compartment, also of type SName, is used to identify the compartment in which the specie is located. The eld initialAmount, of type double, is used to set the initial amount of the specie in the named compartment. The units of the substance quaJ -378.ty may be explicitly set using the optional eld units. The value assigned to units must be chosen from one of the follo possibilities: one of base unit names from Table 4 on page 10, the name \ volume", or a new unit name de ned by a unit de nition in the enclosing model. If absent, the units default to the value set by the built-in \substance" of Table 5 on page 10.

The optional booleaJ -442(eld)]TJ/F46 9.963 Tf 122.233 0 Td[(boundaryCondition)]TJ/F8 9.963 Tf 93.322 0 Td[(determin

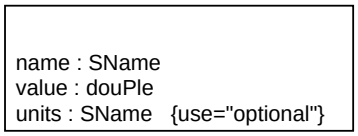

Figure 8: *The de nition of* `Parameter`.

Figure 9: *The definition of* `Rule`

4.6.6 Example of Rule Use

The following is an example use of rules:

```
<model>
    ...
    <listOfRules>
        <parameterRule name="k" formula="k3/k2"/>
        <specieConcentrationRule specie="s2" formula="k * t/(1 + k)"/>
        <compartmentVolumeRule compartment="A" formula="0.10 * t"/>
    </listOfRules>
    ...
</model>
```

4.7 Reactions

A *reaction* represents some transformation, transport or binding process, typically a chemical reaction, that can change the amount of one or more species. The `Reaction` type is de ned in Figure 10.

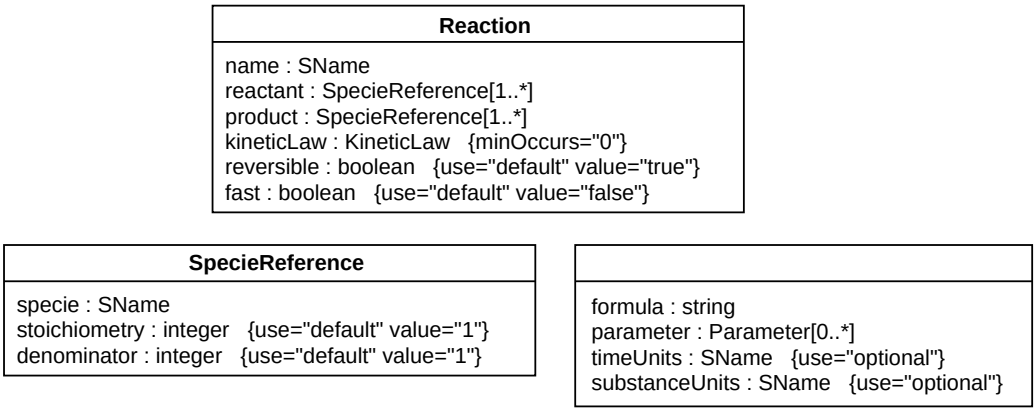

Figure 10: *The de nitions of `Reaction`, `KineticLaw` and `SpecieReference`.*

In SBML, reactions are de ned using lists of reactant species, products, and their stoichiometries, and by parameter values for separately-de ned kinetic laws. These various quantities are recorded in the elds `reactant`, `product`, and `kineticLaw`. Both `reactant` and `product` are references to species implemented using lists of `SpecieReference`

4.7.1 SpecieReference

Each unique specie involved in a reaction is listed once in a model, in a list contained in the specie eld of the Model

```
    <parameter name="k1" value="0"/>
</listOfParameters>
        </lReactionmeters>08.829 -9.315..."0"/>
```

```
            </reaction>
            <reaction name="reaction_2" reversible="false">
                <listOfReactants>
                    <specieReference specie="S1" stoichiometry="1"/>
                </listOfReactants>
                <listOfProducts>
                    <specieReference specie="X1" stoichiometry="1"/>
                </listOfProducts>
                <kineticLaw formula="k2 * S1">
                    <listOfParameters>
                        <parameter name="k2" value="0"/>
                    </listOfParameters>
                </kineticLaw>
            </reaction>
            <reaction name="reaction_3" reversible="false">
                <listOfReactants>
                    <specieReference specie="S1" stoichiometry="1"/>
                </listOfReactants>
                <listOfProducts>
                    <specieReference specie="X2" stoichiometry="1"/>
                </listOfProducts>
                <kineticLaw formula="k3 * S1">
                    <listOfParameters>
                        <parameter name="k3" value="0"/>
                    </listOfParameters>
                </kineticLaw>
            </reaction>
        </listOfReactions>
    </model>
</sbml>
```

```
</unitDefinition>
```

```
            <specie name="x1" compartment="cell" initialAmount="0"/>
        </listOfSpecies>
        <listOfParameters>
            <parameter name="k1" value="1.2"/>
            <parameter name="k2" value="1000"/>
            <parameter name="k3" value="3000"/>
            <parameter name="k4" value="4.5"/>
        </listOfParameters>
        <listOfRules>
            <parameterRule name="t" formula="s1 + s2"/>
            <parameterRule name="k" formula="k3/k2"/>
            <specieConcentrationRule specie="s2" formula="k * t/(1 + k)"/>
            <specieConcentrationRule specie="s1" formula="t - s2"/>
        </listOfRules>
        <listOfReactions>
            <reaction name="j1">
                <listOfReactants>
                    <specieReference specie="x0"/>
                </listOfReactants>
                <listOfProducts>
                    <specieReference specie="s1"/>
                </listOfProducts>
                <kineticLaw formula="k1 * x0"/>
            </reaction>
            <reaction name="j3">
                <listOfReactants>
                    <specieReference specie="s2"/>
                </listOfReactants>
                <listOfProducts>
                    <specieReference specie="x1"/>
                </listOfProducts>
                <kineticLaw formula="k4 * s2"/>
            </reaction>
        </listOfReactions>
    </model>
</sbml>
```

6 Discussion

The volume of data now emerging from molecular biotechnology leave little doubt that extensive computer-based modeling, simulation and analysis will be critical to understanding and interpreting the data (Abbott, 1999; Gilman, 2000; Popel and Winslow, 1998; Smaglik, 2000a). This has lead to an explosion in the development of computer tools by many research groups across the world. The explosive rate of progress is exciting, but the rapid growth of the eld is accompanied by problems and pressing needs.

One problem is that simulation models and results often cannot be directly compared, shared or re-used, because the tools developed by di erent groups often are not compatible with each other. As the eld of systems biology matures, researchers increasingly need to communicate their results as computational

6.1 Future Enhancements to SBML: Level 2 and Beyond

As mentioned above, SBML Level 1 is intended to provide the most basic foundations for modeling biochemical networks. A number of signi cant capabilities are lacking from Level 1; these will be introduced in higher-level de nitions of SBML. The following summarizes additional features that will likely be included in SBML Level 2:

Arrays. This will enable the creation of arrays of compts (sp reactions, compartments and submodels).

Connections. This will be a mechanism for describing the connections between items in an array.

Appendix

A Summary of Notation

The deN014nitive explanation for the notation used in this document can be found in the companion notation document N050Hucka, 2000N051. Here we brieN015y summarize some of the main components of the notations used in describing SBML.

Within the deN014nitions of the various object classes introduced in this document, the following types of expressions are used many times:

```
field1 : float
field2 : integer[0..*]
field3 : N050XHTMLN051
field4 : float {use = "default" value = "0.0"}
```

The symbols `field1`, `field2`, etc., represents N014elds in a data structure. The colon immediately after the name separates the name of the attribute from the type of data that it stores.

More complex speciN014cations use square brackets [N051] just after a type name. This is used to indicate that the N014eld contains a list of elements. SpeciN014cally, the notation `[0..*]` signiN014es a list containing zero or more elements; the notation `[1..*]` signiN014es a list containing at least one element; and so on. The approach used here to translate from a list form into XML is, N014rst, create a subelement named `listOf`

```
<!-- SBase -->
<xsd:complexType name="SBase" abstract="true">
  <xsd:annotation>
    <xsd:documentation>
      Designed to allow a modeler or a pacckage to attach
      information to each component.
```

```
      <xsd:element name="specieConcentrationRule" type="SpecieConcentrationRule"
                   minOccurs="0"/>
      <xsd:element name="parameterRule" type="ParameterRule" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listOfReactions">
  <xsd:complexType>
```

```
</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:complexConient>
    <xsd:extension base="SBase">
      <xsd:attribute name="kind" type="UnitKind" use="required"/>
      <xsd:attribute name="exponent" type="xsd:inieger" use="default" value="1"/>
      <xsd:attribute name="scale" type="xsd:inieger" use="default" value="1"/>
    </xsd:extension>
  </xsd:complexConient>
</xsd:complexType>
<!-- Rule -->
<xsd:simpleType name="RuleType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="scalar"/>
    <xsd:enumeration value="rate"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="Rule" abstract="true">
  <xsd:complexConient>
    <xsd:extension base="SBase">
      <xsd:attribute name="formula" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:complexConient>
</xsd:complexType>
<xsd:complexType name="AlgebraicRule">
  <xsd:complexConient>
    <xsd:extension base="Rule"/>
  </xsd:complexConient>
```

```
  <xsd:complexContent>
    <xsd:extension base="SBase">
      <xsd:sequence>
        <xsd:element ref="listOfParameters" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="formula" type="xsd:string" use="required"/>
      <xsd:attribute name="timeUnits" type="SName" use="optional"/>
      <xsd:attribute name="substanceUnits" type="SName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- Top-level elements allowed in an SBML document. -->
<xsd:complexType name="sbmlDocument">
  <xsd:sequence>
    <xsd:element name="model" type="Model"/>
  </xsd:sequence>
  <xsd:attribute name="xmlns"/>
  <xsd:attribute name="level" type="xsd:positiveInteger" use="required"/>
  <xsd:attribute name="version" type="xsd:positiveInteger" use="required"/>
</xsd:complexType>
<xsd:element name="sbml" type="sbmlDocument"/>
```

C Prede ned Functions in SBML

Table 6 lists the basic mathematical functions that ar7 de ned 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 cosin7 of x in radians	$-1.0 \le x \le 1.0$	$0 \le acos(x) \le \pi$
asin	x	arc sin7 of x in radians	$-1.0 \le x \le 1.0$	$-\pi/2 \le asin(x) \le \pi/2$
atan	x	arc tangent of x in radians		$-\pi/2 \le atan(x) \le \pi/2$
ceil	x	smallest number not less than x whos7 value is an exact integer		
cos	x	cosin7 of x		
exp	x	e^x, where e is the bas7 of the natural logarithm		
oor	x	the largest number not greater than x whos7 value is an exact integer		
log10	x	base-10 logarithm of x	$x > 0$	
log	x, y	base-y logarithm of x	$x > 0, y > 0, y \neq 1$	
pow	x, y	x^y		
sin	x	sin7 of x		
sqr	x	x^2		
sqrt	x	$\sqrt{x}$	$x \ge 0$	$sqrt(x) \ge 0$
tan	x	tangent of x	$x \neq (n/2)\pi$, for odd integer n	

Table 6: *Basic mathematical functions de ned in SBML.*

Table 7 de nes the rate law functions available in formula expressions in SBML. Thes7 were extracted from the Gepasi help le (3.21). Segel (1993) provides more information; Hofmeyr and Cornish-Bowden (1997)

Name	Arguments	Meaning	Formula
usii	S, V, K_m, K_i	Substrate InhibitionJ ET 0 -11.955 Td[(Kinetics)-333((Irrev)28(ersible))]TJ ET 113.977 -22.6	

Name	Arguments	Meaning	Formula
uctr	S, P, A_c, V_f, V_r, K_{ms}, K_{mp}, K_a	Catalytic Activation (Reversible)	$v = \frac{V_f S/K_{ms} \quad V_r P/K_{mp}}{1 + K_a/A}$

Symbol	Meaning
	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	Second substrate in two substrate reaction
I	