Systems Biology Markup Language (SBML) Level 2 Proposal: Multistate Features

Nicolas Le Novère, Thomas Simon Shimizu, Andrew Finney
lenov@pasteur.fr,tss26@cus.cam.ac.uk,afinney@cds.caltech.edu

July 1, 2002

Contents

1 Introduction

This document describes a proposed extension for inclusion in Systems Biology Markup Language (SBML) Level 2. It describes features enabling the inclusion of complexes with several alternative states in models.

This document is not a definition of SBML Level 2 or part of it. This document simply presents various features which could be incorporated into SBML Level 2 as the Systems Biology community wishes. This document is intended for detailed review by that community and to provoke alternative proposals. Throughout this document issues that the authors believe will require further discussion have been highlighted.

For brevity the text of this document is with reference to SBML Level 1 (Hucka et al., 2001), i.e. features are described in terms of changes to SBML Level 1. This document uses UML diagrams in the same way except that new features are shown in red.

All types proposed in this document will be derived from the `SBase` type.

2 Why this extension?

An alternative introduction to the problem of multistate reactants can be found in Andrew Finney's initial proposal "Complex Species:species with multiple states" (Finney, 2001).

Many biological macromolecules possess multiple internal states which can affect reaction rates. Typical examples are:

- Different relative atomic coordinates ⇒ Conformational changes or folding
- Covalent modification ⇒ glycosylation, phosphorylation, methylation
- Non-covalent modification ⇒ metal or ligand binding

In modelling reaction systems that involve such molecules, it is possible to treat all the different states as separate species. However the number of possible reactions increases exponentially with the number of reacting species (of course in most cases, not all states will affect every reaction, so the number of reactions which need to be computed separately will be somewhat less than this). Writing out all of these reactions separately is tedious at best, and devastating at worst. It is desirable to have an efficient notation which compresses the redundant information. In addition, some simulators (e.g. StochSim (Morton-Firth and Bray, 1998) or MCell (Stiles et al., 1996)) explicitly consider individual molecules, not populations of molecular species. This calls for a mechanism in SBML which can distinguish between specific instances of the same specie.

3 Specie

Andrew's initial proposal (Finney, 2001) allowed a subset of the state-dependent reaction instances that involve different reactant states, but have identical rates, to be grouped together and expressed as a single reaction. This was achieved by defining several new SBML elements such as `complexSpecie` (for defining species with multiple states) and `complexSpecieInstance` (for distinguishing between specific instances of the same specie that take part in a reaction). However, in the case that different states have a different effect on the reaction rate, these had to be defined as separate reactions.

We take a similar, but slightly different approach that introduces some new elements, but attempts to incorporate much of the multistate-specific information by extending the existing SBML 1 elements with optional attributes.

The proposed structure of the `Specie` type is shown in figure 1.

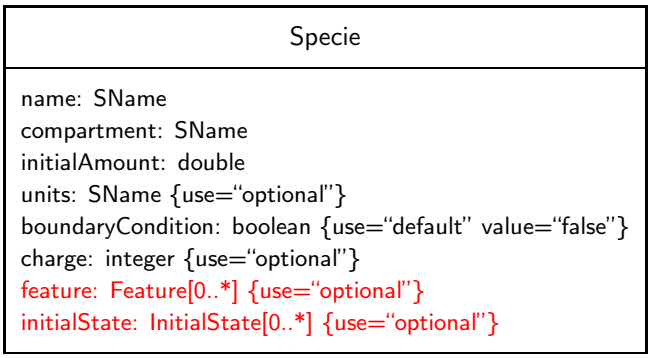

Figure 1: *The definition of the proposed extended* `Specie` *type. Addenda are shown in red.*

The "state" of a multi-state molecule is defined collectively by the states of all "features" that it possesses. A "feature" here is a characteristic of the specie which can be in one of at least two states that affect certain reaction rates. Therefore the extended `specie` element posesses now two new attributes, a `listOfFeatures` and a `listOfInitialStates`.

3.1 Feature

The `listOfFeatures` lists all the features of the specie which can possess several alternative states.

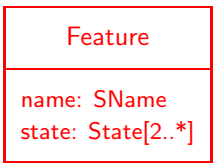

Figure 2: *The definition of a specific Feature attached to a Specie type.*

3.1.0.1 State

Each `feature` element contains a `listOfStates` child, containing at least two `state` element (otherwise one doesn't need this feature, do we?).

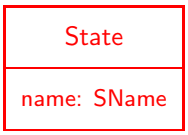

Figure 3: *The definition of one of the states possibly taken by a specific `feature` of a `specie` type.*

3.1.1 Example of a Feature

The following example describes a protein which can exist under various conformations, according to its degree of folding. Therefore we define a feature named "Folding", which here can take three alternative values: "unfolded", "folded" and "inactivated" (the latter can correspond to a degradation, a misfolding, or even to an interaction with some kind of other molecule such as a chaperone).

```
<feature name="Folding">
    <listOfStates>
        <state name="unfolded">
        <state name="folded">
        <state name="inactivated">
    </listOfStates>
</feature>
```

3.2 InitialState

The `listOfInitialStates` expresses the initial amount of each state of the specie (that is specific sets of values taken by each of the features) which is present at the beginning of the simulation. All features with an `initialAmount` different of zero must be listed in the `listOfInitialStates`.

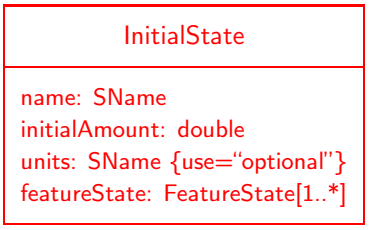

Figure 4: *The definition of a specific `InitialState` of a given `Specie` type*

The sum of the `initialAmount` of all the `initialStates` (Figure 4) in a `listOfInitialStates` must equal the `initialAmount` of the corresponding specie (i.e. all instances of the specie are under one state or another).

3.2.1 FeatureState

The `listOfFeatureState` element describes one state of the specie, i.e. a unique list of values taken by all the features.

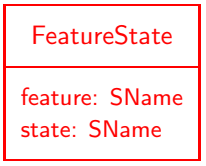

Figure 5: *The definition of a particular state taken by a specific `feature` of a `specie` type*

3.2.2 Example of an InitialState

```
<initialState name="nonactivated" initialAmount="500">
    <listOfFeatureStates>
        <featureState name="Folding" state="unfolded">
        <featureState name="Phosphorylation" state="noPhosphate">
   </listOfFeatureStates>
</initialState>
```

3.3 Complete example of a specie element

```
<specie name="Specie1" initialAmount="1000">
    <listOfFeatures>
        <feature name="Folding">
            <listOfStates>
                <state name="unfolded">
                <state name="folded">
                <state name="inactivated">
            </listOfStates>
        </feature>
        <feature name="Phosphorylation">
            <listOfStates>
                <state name="noPhosphate">
                <state name="Phosphate">
            </listOfStates>
        </feature>
    </listOfFeatures>
    <listOfInitialStates>
        <initialState name="nonactivated" initialAmount="500">
            <listOfFeatureStates>
                <featureState name="Folding" state="unfolded">
                <featureState name="Phosphorylation" state="noPhosphate">
            </listOfFeatureStates>
        </initialState>
        <initialState name="activated" initialAmount="500">
            <listOfFeatureStates>
                <featureState name="Folding" state="folded">
                <featureState name="Phosphorylation" state="Phosphate">
            </listOfFeatureStates>
        </initialState>
    </listOfInitialStates>
</specie>
```

3.4 Issues

In this proposal, we have attempted to express multistate molecules by extending the `complex` element present in SBML 1. This contrasts from Andrew's initial approach of introducing a novel `complexSpecie` element. We think our approach works quite well, but have we left anything out? Are there any objections to extending the `complex` element using optional attributes?

Confusion could arise from the dual meaning we use for the word "state". A *feature* is one of the many characteristics of a specie, which can take a discrete number of **states**. At the same time, a **state** of the *specie* itself is defined by a specific set of feature **states**.

The `compartment` element could be conserved for SBML 1 compatibility. However its removal would be coherent with the extension proposed by the ECell group. If a specie is defined at the root of the model, it is defined for every compartment (The initialAmount has to be expressed as a concentration then). If a specie is defined in a subset of the various compartments, we don't need to specify the compartments since all the definitions are located within the compartments themselves.

4 Reactions

We present two alternatives of how to extend the `reaction` element to enable the distinction between specific instances of species. One does not require any change of the `reaction` element, but affects the `specieReference` instead, while the other affects the `reaction` element itself. We would appreciate feedback on which you think is better.

Before that, we will present parts of the extension which are common to both options, dealing with the modification of reaction rates, and with the states of nascent species, created or modified by reactions. We utilise the concept of a "reaction modifier", which defines the effect of the reactant state on a reaction. This allows all state-dependent instances of a reaction to be expressed as a single reaction.

4.1 kineticLaw

Because the value of a reaction rate modifier can depend on the state of more than one species, we need to tie it to kinetic parameters rather than individual specie concentrations.

In this proposal a `listOfStateEffects` is an optional element, child of the `kineticLaw` element (Figure 6).

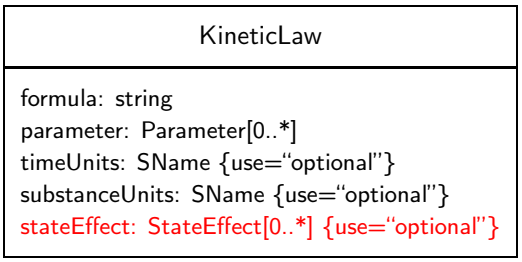

Figure 6: *The definition of the proposed extended `kineticLaw` element. Addenda are shown in red.*

4.1.1 stateEffect

The `stateEffect` element specifies which parameter it will modify (by name), the actual value of the modifier (as a real-valued number) and a `listOfSpecieStates`. `specieState` (there might be a better name for this) is a new element used to specify the set of conditions under which this `stateEffect` applies. The elements of the `listOfSpecieStates` are interpreted using the AND operator, so the conditions in all elements of the list must be satisfied for the `stateEffect` to apply (figure 7).

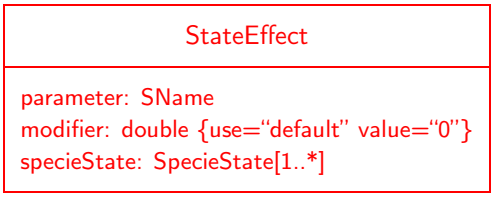

Figure 7: *The definition of the stateEffect element.*

The "modifier" attribute is the coefficient which modulates the reaction velocity. It is multiplied with the term of the kinetic law which involves the specie to which it is associated. Within the framework of the mass action law, this is effectively equivalent to modifying the quantity of the reacting specie.

4.1.1.1 specieState

Each `specieState` element specifies a `specieReference` (or a `specieInstance`, see section 4.4) by its name and a `listOfFeatureConditions`.

Figure 8: *The definition of the specieState element.*

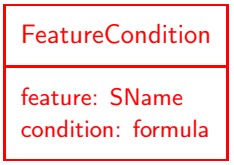

Figure 9: *The definition of the `featureCondition` element.*

4.1.1.1.1 featureCondition Each `featureCondition` element uses logic expressions (consisting of parentheses and the operators AND, OR and NOT) to define the states of each feature that the `stateEffect` applies to. For instance, if a feature has five states A, B, C, D and E, and a `stateEffect` applies to states A or B, one could either write "A OR B", or "NOT (C OR D OR E)".

(Comment: The data type "LogicExpression" could be called LogExpression, LExpression or even LExp, or again BooleanExpression or BoolExpression.)

States which do not match the `listOfFeatureConditions` of any of the `stateEffects` of a reaction are assumed to have a modifier of 0 (i.e. they cannot take part in the reaction).

4.1.2 example

Reactants A and B bind to produce reactant C. Both A and B can be in one of two states, active, or inactive (denoted by a subscript of 1 or 0, respectively). The rates are:

- $A_0 + B_0 \rightarrow C$ — modifier $= 0$
- $A_0 + B_1 \rightarrow C$ — modifier $= 0.6$
- $A_1 + B_0 \rightarrow C$ — modifier $= 0.2$
- $A_1 + B_1 \rightarrow C$ — modifier $= 1$

```
<kineticLaw formula="kf * A * B - kr * C">
    <listOfParameters>
        <parameter name ="kf" value="1000">
        <parameter name ="kr" value="1">
    </listOfParameters>
    <listOfStateEffect>
        <stateEffect parameter="kf" modifier="0.6">
            <listOfSpecieStates>
                <specieState specie="A">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="inactive">
                    </listOfFeatureConditions>
                </specieState>
                <specieState specie="B">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="active">
                    </listOfFeatureConditions>
```

```
                </specieState>
            </listOfSpecieStates>
        </stateEffect>
        <stateEffect parameter="kf" modifier="0.2">
            <listOfSpecieStates>
                <specieState specie="A">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="active">
                    </listOfFeatureConditions>
                </specieState>
                <specieState specie="B">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="inactive">
                    </listOfFeatureConditions>
                </specieState>
            </listOfSpecieStates>
        </stateEffect>
        <stateEffect parameter="kf" modifier="1.0">
            <listOfSpecieStates>
                <specieState specie="A">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="active">
                    </listOfFeatureConditions>
                </specieState>
                <specieState specie="B">
                    <listOfFeatureConditions>
                        <featureCondition feature="Activity" state="active">
                    </listOfFeatureConditions>
                </specieState>
            </listOfSpecieStates>
        </stateEffect>
    </listOfStateEffects>
</kineticLaw>
```

4.2 States of nascent molecules

If the newly created molecule is also a multistate complex, it is necessary to specify all the state of all of its features. This is easily implemented by creating a new element called `nascentState` (figure 10) which has a `listOfFeatureStates` element as child (see section 3.2.1). `nascentState` in turn becomes an optional child of the `specieReference` element (option 1 section 4.3) or the `specieInstance` element (option 2 section 4.4).

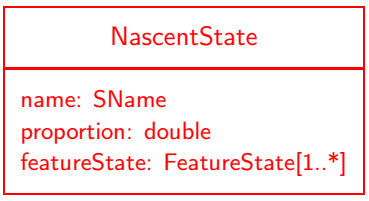

Figure 10: *The definition of a specific* `nascentState` *of a given* `specie` *type*

If the created molecules come to existence under a specific set of nascent states, which well defined probabilities, we fill the `nascentState` elements. Each element contains an attribute which specify the probability of that state (the default is 1. It is up to the parser to rescale everything if the sum of the `proportion` elements over all the `nascentState` elements is more than 1).

4.2.1 Example

Here is the result of a ligand binding affecting the activity of a specie. The specie has two features: the presence of ligand, and the activity.

```
<listOfNascentState>
    <nascentState name="A1" proportion="0.9">
        <listOfFeatureState>
            <featureState name="ligand" state="bound"/>
            <featureState name="activity" state="active"/>
```

```
            </listOfFeatureState>
        </nascentState>
        <nascentState name="I1" proportion="0.1">
            <listOfFeatureState>
                <featureState name="ligand" state="bound"/>
                <featureState name="activity" state="inactive"/>
            </listOfFeatureState>
        </nascentState>
    </listOfNascentState>
```

In case the nascent state depends on the state(s) of the reactants, we have to enumerate the various reactions, with each `listOfNascentState` limited to one element.

If no nascent states are defined for a multistate product, the feature values are allocated randomly with a stochastic algorithm, or evenly with a deterministic algorithm.

And now, let's turn to the alternative proposals for the `reaction` element and its children.

4.3 *First possibility:* `reaction` *unchanged*

4.3.1 *specieReference*

In a first option, the `reaction` element is not modified, but the `specieReference` element, as shown in figure 11.

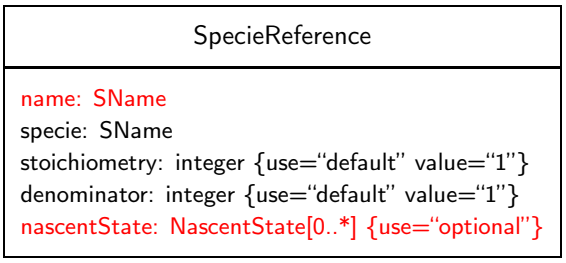

Figure 11: *The definition of the extended specieReference element under the first option. Addenda are shown in red.*

The addition of the `name` attribute to the `specieReference` element allows different instances of the same specie to be distinguished within a reaction. This name would be used instead of the specie name in the `kineticLaw`. It takes the same value as the `specie` by default, so it need not be explicitly defined for reactants that don't need specific instances identified (i.e. if the reactant and product species do not have multiple states, as in SBML 1).

4.3.2 *example*

An example of a reaction which requires specific instances to be identified is the reversible transfer of a phosphate group between two molecules of specie A:

$$A1P + A2 \rightleftharpoons A1 + A2P$$

For this reaction, `listOfReactants` and `listOfProducts` would look like this (see figure 7 for the definition of `stateEffect` and figure 9 for the definition of `featureCondition`):

```
<listOfReactants>
    <specieReference name="A1" specie="A" />
    <specieReference name="A2" specie="A" />
</listOfReactants>
<listOfProducts>
    <specieReference name="A1" specie="A" />
    <specieReference name="A2" specie="A"/>
</listOfReactants>
<kineticLaw formula="(kf * Specie1A * Specie1B - kr * Specie1A * Specie1B) * compOne * N_A">
    <listOfParameters>
```

```
            <parameter name="kf" value="100" />
            <parameter name="kr" value="0" />
        </listOfParameters>
        <listOfStateEffects>
            <stateEffect parameter="kf" modifier="1">
                <listOfSpecieStates>
                    <specieState specie="A1">
                        <listOfFeatureConditions>
                            <featureCondition feature="Phosphorylation" state="Phosphate"/>
                        </listOfFeatureConditions>
                    </specieState>
                    <specieState specie="A2">
                        <listOfFeatureConditions>
                            <featureCondition feature="Phosphorylation" state="noPhosphate">
                        </listOfFeatureConditions>
                    </specieState>
                </listOfSpecieState>
            </stateEffect>
             <stateEffect parameter="kr" modifier="1">
                <listOfSpecieStates>
                    <specieState specie="A1">
                        <listOfFeatureConditions>
                            <featureCondition feature="Phosphorylation" state="noPhosphate"/>
                        </listOfFeatureConditions>
                    </specieState>
                    <specieState specie="A2">
                        <listOfFeatureConditions>
                            <featureCondition feature="Phosphorylation" state="Phosphate">
                        </listOfFeatureConditions>
                    </specieState>
                </listOfSpecieState>
            </stateEffect>
        </listOfStateEffects>
    </kineticLaw>
```

Note that both `specieReference` elements A1 and A2 point to the same specie A. To be compared with the second possibility described in the section 4.4

4.4 *Second possibility: extension of the* `reaction` *element*

The second option requires modifications of the `reaction` element as shown in figure 12.

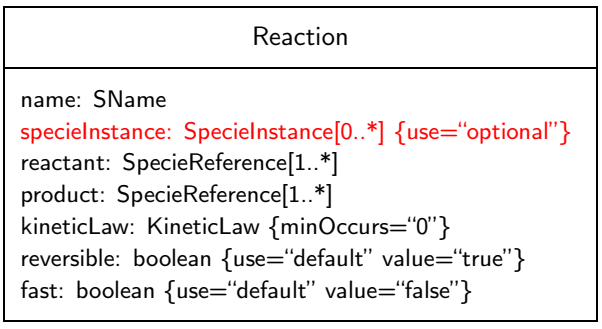

Figure 12: *The definition of the proposed extended* `reaction` *element. Addenda are shown in red.*

4.4.1 *specieInstance*

This solution follows Andrew's idea of explicitly representing individual instances of species with a new element, which we call `specieInstance` (depicted in figure 13). This is used to distinguish between individual instances of the species involved in a reaction. The `listOfSpecieInstance` attribute of the reaction element is optional, and would only need to be used in reactions where specific instances of reacting species need to be distinguished.

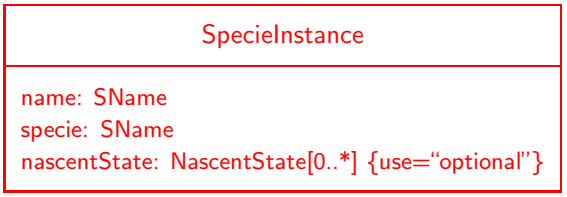

Figure 13: *The definition of specieInstance*

4.4.2 example

Under this scheme, to express the reactants and products of the reversible phosphorylation example presented above, one would also need a `listOfSpecieInstances`. Note that the `specieReference` elements point to different `specie` elements, in fact `specieInstance` elements (to be compared with the first possibility presented in section 4.3). The definition would then looks like (see figure 7 for the definition of `stateEffect` and figure 9 for the definition of `featureCondition`):

```
<listOfSpecieInstances>
    <SpecieInstance name="A1" specie="A" />
    <SpecieInstance name="A2" specie="A" />
</listOfSpecieInstances>
<listOfReactants>
    <specieReference specie="A1" />
    <specieReference specie="A2" />
</listOfReactants>
<listOfProducts>
    <specieReference specie="A1" />
    <specieReference specie="A2" />
</listOfReactants>
<kineticLaw formula="(kf * Specie1A * Specie1B - kr * Specie1A * Specie1B) * compOne * N_A">
    <listOfParameters>
        <parameter name="kf" value="100" />
        <parameter name="kr" value="0" />
    </listOfParameters>
    <listOfStateEffects>
        <stateEffect parameter="kf" modifier="1">
            <listOfSpecieStates>
                <specieState specie="A1">
                    <listOfFeatureConditions>
                        <featureCondition feature="Phosphorylation" state="Phosphate"/>
                    </listOfFeatureConditions>
                </specieState>
                <specieState specie="A2">
                    <listOfFeatureConditions>
                        <featureCondition feature="Phosphorylation" state="noPhosphate">
                    </listOfFeatureConditions>
                </specieState>
            </listOfSpecieState>
        </stateEffect>
         <stateEffect parameter="kr" modifier="1">
            <listOfSpecieStates>
                <specieState specie="A1">
                    <listOfFeatureConditions>
                        <featureCondition feature="Phosphorylation" state="noPhosphate"/>
                    </listOfFeatureConditions>
                </specieState>
                <specieState specie="A2">
                    <listOfFeatureConditions>
                        <featureCondition feature="Phosphorylation" state="Phosphate">
                    </listOfFeatureConditions>
                </specieState>
            </listOfSpecieState>
        </stateEffect>
    </listOfStateEffects>
</kineticLaw>
```

5 Complete example

Note that this example has intentionally been made independent of StochSim. In StochSim, the features of multistate complexes are represented by binary flags, so each can only have two states. In this example the feature "Folding" possess three states (this could be encoded in StochSim using two binary flags).

```
<sbml version="1" level="1">
    <model name="ExampleMultipleState" />
    <notes>
        <body xmlns="http://www.w3.org/1999/xhtml">
            <p>This model exemplifies the use of the extension proposed by the StochSim team.</p>
            <p>The main reaction is Specie1 + Specie2 <=> Specie3</p>
            <p>Specie1 possesses 2 features affecting the reaction rate.
               The "Folding" exists under three states "unfolded", "folded" and "inactivated",
               the "Phosphorylation" under two "noPhosphate" and "Phosphate".</p>
            <p>The features are regulated by two interconversions and one cross-phosphorylation.</p>
            <p>Specie1-unfolded <=> Specie1-folded</p>
            <p>Specie1-folded <=> Specie1-inactivated</p>
            <p>Specie1-folded + Specie1-inactivated-P <=> Specie1-folded-P + Specie1-inactivated </p>
        </body>
    </notes>

    <listOfParameters>
        <parameter name="N_A" value="6.022e23" />
    </listOfParameters>

    <listOfCompartments>
        <compartment name="compOne">
            <listOfSpecies>
                <specie name="Specie1" initialAmount="1000">
                    <listOfFeatures>
                        <feature name="Folding">
                            <listOfStates>
                                <state name="unfolded"/>
                                <state name="folded"/>
                                <state name="inactivated"/>
                            </listOfStates>
                        </feature>
                        <feature name="Phosphorylation">
                            <listOfStates>
                                <state name="noPhosphate"/>
                                <state name="Phosphate"/>
                            </listOfStates>
                        </feature>
                    </listOfFeatures>
                    <listOfInitialStates>
                        <initialState name="nonactivated" initialAmount="500">
                            <listOfFeatureStates>
                                <featureState feature="Folding" state="unfolded"/>
                                <featureState feature="Phosphorylation" state="noPhosphate"/>
                            </listOfFeatureStates>
                        </initialState>
                        <initialState name="activated" initialAmount="500">
                            <listOfFeatureStates>
                                <featureState feature="Folding" state="folded"/>
                                <featureState feature="Phosphorylation" state="Phosphate"/>
                            </listOfFeatureStates>
                        </initialState>
                    </listOfInitialStates>
                </specie>
                <specie name="Specie2" initialAmount="1000" />
                <specie name="Specie3" initialAmount="0" />
            </listOfSpecies>

            <listOfReactions>
                <reaction  name="main">
                    <listOfReactants>
                        <specieReference specie="Specie1" stoichiometry="1"/>
                        <specieReference specie="Specie2" stoichiometry="1"/>
                    </listOfReactants>
                    <listOfProducts>
                        <specieReference name="Specie3" stoichiometry="1"/>
```

```
                    </listOfProducts>
                    <kineticLaw formula="(kf * Specie1 * Specie2 - kr * Specie3) * compOne * N_A">
                        <listOfParameters>
                            <parameter name="kf" value="2500" />
                            <parameter name="kr" value="100" />
                        </listOfParameters>
                        <listOfStateEffects>
                            <stateEffect parameter="kf" modifier="0.5">
                                <listOfSpecieState>
                                    <specieState specie="specie1">
                                        <listOfFeatureConditions>
                                            <featureCondition feature="Folding" condition="folded"/>
                                            <featureCondition feature="Phosphorylation" condition="noPhosphate"/>
                                        </listOfFeatureConditions>
                                    </specieState>
                                </listOfSpecieState>
                            </stateEffect>
                            <stateEffect parameter="kf" modifier="1">
                                <listOfSpecieState>
                                    <specieState specie="specie1">
                                        <listOfFeatureConditions>
                                            <featureCondition feature="Folding" condition="folded"/>
                                            <featureCondition feature="Phosphorylation" condition="Phosphate"/>
                                        </listOfFeatureConditions>
                                    </specieState>
                                </listOfSpecieState>
                            </stateEffect>

<!-- This is optional (since the default modifier is 0), and will be omitted in later listOfStateEffects -->

                            <stateEffect parameter="kf" modifier="0">
                                 <listOfSpecieState>
                                    <specieState specie="specie1">
                                        <listOfFeatureConditions>
                                            <featureCondition feature="Folding" condition="unfolded or inactivated"/>
                                        </listOfFeatureConditions>
                                    </specieState>
                                </listOfSpecieState>
                            </stateEffect>

<!-- ================================================================================================== -->

                        </listOfStateEffects>
                    </kineticLaw>
                </reaction>
                <reaction name="folding">
                    <listOfReactants>
                        <specieReference specie="Specie1" stoichiometry="1" />
                    </listOfReactants>
                    <listOfProducts>
                        <specieReference specie="Specie1" stoichiometry="1">
                    </listOfProducts>
                    <kineticLaw formula="(kf * Specie1 - kr * Specie1) * compOne * N_A">
                        <listOfParameters>
                            <parameter name="kf" value="1000" />
                            <parameter name="kr" value="10" />
                        </listOfParameters>
                        <listOfStateEffects>
                            <stateEffect parameter="kf" modifier="1">
                                <listOfSpecieState>
                                    <specieState specie="specie1">
                                        <listOfFeatureConditions>
                                            <featureCondition feature="Folding" condition="unfolded"/>
                                        </listOfFeatureConditions>
                                    </specieState>
                                </listOfSpecieState>
                            </stateEffect>
                            <stateEffect parameter="kr" modifier="1">
                                <listOfSpecieState>
                                    <specieState specie="specie1">
                                        <listOfFeatureConditions>
                                            <featureCondition feature="Folding" condition="folded"/>
```

```
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                    </listOfStateEffects>
                </kineticLaw>
            </reaction>

            <reaction name="inactivation">
                <listOfReactants>
                    <specieReference specie="Specie1" stoichiometry="1" />
                </listOfReactants>
                <listOfProducts>
                    <specieReference specie="Specie1" stoichiometry="1">
                </listOfProducts>
                <kineticLaw formula="(kf * Specie1 - kr * Specie1) * compOne * N_A">
                    <listOfParameters>
                        <parameter name="kf" value="100" />
                        <parameter name="kr" value="0" />
                    </listOfParameters>
                    <listOfStateEffects>
                        <stateEffect parameter="kf" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1">
                                    <listOfFeatureConditions>
                                        <featureCondition feature="Folding" condition="folded"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                        <stateEffect parameter="kr" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1">
                                    <listOfFeatureConditions>
                                        <featureCondition feature="Folding" condition="inactivated"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                    </listOfStateEffects>
                </kineticLaw>
            </reaction>

            <reaction name="trans-phosphorylation">

<!-- ========================================== Second solution ====================================== -->

                <listOfSpecieInstances>
                    <specieInstance name="Specie1A" specie="Specie1" />
                    <specieInstance name="Specie1B" specie="Specie1" />
                </listOfSpecieInstances>

                <listOfReactants>

<!-- ========================================== First solution ======================================= -->

                    <specieReference name="Specie1A" specie="Specie1" stoichiometry="1" />
                    <specieReference name="Specie1B" specie="Specie1" stoichiometry="1" />

<!-- ========================================== Second solution ====================================== -->

                    <specieReference specie="Specie1A" stoichiometry="1" />
                    <specieReference specie="Specie1B" stoichiometry="1" />

                </listOfReactants>
                <listOfProducts>
<!-- ========================================== First solution ======================================= -->

                    <specieReference name="Specie1A" specie="Specie1" stoichiometry="1" />
                    <specieReference name="Specie1B" specie="Specie1" stoichiometry="1" />

<!-- ========================================== Second solution ====================================== -->
```

```
                    <specieReference specie="Specie1A" stoichiometry="1">
                    <specieReference specie="Specie1B" stoichiometry="1">

                </listOfProducts>
                <kineticLaw formula="(kf * Specie1A * Specie1B - kr * Specie1A * Specie1B) * compOne * N_A">
                    <listOfParameters>
                        <parameter name="kf" value="100" />
                        <parameter name="kr" value="0" />
                    </listOfParameters>
                    <listOfStateEffects>
                        <stateEffect parameter="kf" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1A">
                                    <listOfFeatureConditions>
                                        <featureCondition feature="Folding" condition="folded"/>
                                        <featureCondition feature="Phosphorylation" condition="noPhosphate"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                        <stateEffect parameter="kf" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1B">
                                    <listOfFeatureConditions>
                                        <featureCondition feature="Folding" condition="inactivated"/>
                                        <featureCondition feature="Phosphorylation" condition="Phosphate"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                        <stateEffect parameter="kr" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1A">
                                    <listOfFeatureConditions>
                                        <featureCondition name="Folding"  condition="folded"/>
                                        <featureCondition name="Phosphorylation" condition="Phosphate"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                        <stateEffect parameter="kr" modifier="1">
                            <listOfSpecieState>
                                <specieState specie="specie1B">
                                    <listOfFeatureConditions>
                                        <featureCondition feature="Folding" condition="inactivated"/>
                                        <featureCondition feature="Phosphorylation" condition="noPhosphate"/>
                                    </listOfFeatureConditions>
                                </specieState>
                            </listOfSpecieState>
                        </stateEffect>
                    </listOfStateEffects>
                </kineticLaw>
            </reaction>
        </listOfReactions>
    </compartment>
</listOfCompartments>
</model>
```

References

Finney, A. (2001). Possible extension to the systems biology markup language. complex species: species with multiple states. Technical report, ERATO Kitano Systems Biology Workbench Development Group. Internal Discussion Document.

Hucka, M., Finney, A., Sauro, H., and Bolouri, H. (2001). Systems biology markup language (sbml) level 1: Structures and facilities for basic model definitions. Technical report, Systems Biology Workbench Development Group. available via the World Wide Web `http://www.cds.caltech.edu`.

Morton-Firth, C. and Bray, D. (1998). Predicting temporal fluctuations in an intracellular signalling pathway.

J. Theor. Biol., 192:117–128. available via the World Wide Web `http://www.zoo.cam.ac.uk/comp-cell/StochSim.html`.

Stiles, J., Van Helden, D., Bartol, TM, J., Salpeter, E., and Salpeter, M. (1996). iniature endplate current rise times <100 ms from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. *Proc. Natl. Acad. Sci. USA*, 93. available via the World Wide Web `http://www.mcell.cnl.salk.edu/`.