

## Multistate, Multicomponent and Multicompartment Species Package for SBML Level 3 (*Multi*)

Fengkai Zhang  
[zhangfen@niaid.nih.gov](mailto:zhangfen@niaid.nih.gov)  
Laboratory of Systems Biology  
NIAID/NIH  
Bethesda, MD, USA

Martin Meier-Schellersheim  
[mms@niaid.nih.gov](mailto:mms@niaid.nih.gov)  
Laboratory of Systems Biology  
NIAID/NIH  
Bethesda, MD, USA

Version 1, Release 0.1 (Draft, Rev 369)

September 2013

This is a draft specification for the SBML Level 3 package called “*Multi*”. It is not a normative document. Please send feedback to the package mailing list at [sbml-multi@lists.sourceforge.net](mailto:sbml-multi@lists.sourceforge.net).

The latest release, past releases, and other materials related to this specification are available at

[TBD](#)

*This* release of the specification is available at

[TBD](#)



## Contributors

Fengkai Zhang  
Laboratory of Systems Biology  
NIAID/NIH  
Bethesda, MD, USA

Anika Oellrich  
European Bioinformatics Institute  
Wellcome Trust Genome Campus  
Hinxton, Cambridge, UK

Lucian P. Smith  
Computing and Mathematical Sciences  
California Institute of Technology  
Seattle, Washington, USA

Bastian Angermann  
Laboratory of Systems Biology  
NIAID/NIH  
Bethesda, MD, USA

James Faeder  
Department of Computational Biology  
University of Pittsburgh School of Medicine  
Pittsburgh, PA, USA

Leonard A. Harris  
Department of Computational Biology  
University of Pittsburgh School of Medicine  
Pittsburgh, PA, USA

Stefan Hoops  
Virginia Bioinformatics Institute  
Virginia Tech  
Blacksburg, VA, USA

Michael Hucka  
Computing and Mathematical Sciences  
California Institute of Technology  
Pasadena, CA, USA

Nicolas Rodriguez  
European Bioinformatics Institute  
Wellcome Trust Genome Campus  
Hinxton, Cambridge, UK

Martin Meier-Schellersheim  
Laboratory of Systems Biology  
NIAID/NIH  
Bethesda, MD, USA

Nicolas Le Novère  
European Bioinformatics Institute  
Wellcome Trust Genome Campus  
Hinxton, Cambridge, UK

Michael Blinov  
Dept. of Genetics & Developmental Biology  
University of Connecticut Health Center  
Farmington, CT, USA

Andrew Finney  
University Hertfordshire  
Hatfield, Herts, UK

William S. Hlavacek  
Theoretical Division  
Los Alamos National Laboratory  
Los Alamos, NM, USA

Bin Hu  
Theoretical Division  
Los Alamos National Laboratory  
Los Alamos, NM, USA

Alida Palmisano  
Department of Biological Sciences  
Virginia Tech  
Blacksburg, VA, USA

And all the people who contributed to the discussions on the sbml-multi mailing list.

# Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Proposal corresponding to this package specification	5
1.2 Package dependencies	5
1.3 Document conventions	5
<b>2 Background and context</b>	<b>6</b>
2.1 Past work on this problem or similar topics	6
<b>3 Package syntax and semantics</b>	<b>8</b>
3.1 Namespace URI and other declarations necessary for using this package	8
3.2 Primitive data types	8
3.2.1 Type <code>BindingStatus</code>	8
3.2.2 Type <code>Relation</code>	8
3.2.3 Type <code>RepresentationType</code>	8
3.3 The new and extended classes in the <i>Multi</i> Package	8
3.4 <b>Model</b>	9
3.4.1 <b>ListOfSpeciesTypes</b>	9
3.5 <b>Compartment</b>	9
3.5.1 The <code>isType</code> attribute	9
3.5.2 The <code>compartmentType</code> attribute	10
3.5.3 <b>ListOfCompartmentReferences</b>	10
3.6 <b>CompartmentReference</b>	10
3.6.1 The <code>id</code> and <code>name</code> attributes	10
3.6.2 The <code>compartment</code> attribute	10
3.7 The relationship of <b>Compartment</b> , <b>CompartmentReference</b> and <b>ListOfCompartmentReferences</b>	10
3.8 <b>SpeciesType</b> class	11
3.8.1 The <code>id</code> and <code>name</code> attributes	11
3.8.2 The <code>isBindingSite</code> attribute	12
3.8.3 The <code>compartment</code> attribute	12
3.8.4 <b>ListOfSpeciesFeatureTypes</b>	12
3.8.5 <b>ListOfSpeciesTypeInstances</b>	12
3.8.6 <b>ListOfInSpeciesTypeBonds</b>	12
3.8.7 <b>ListOfSpeciesTypeComponentIndexes</b>	12
3.9 <b>SpeciesFeatureType</b>	12
3.9.1 The <code>id</code> and <code>name</code> attributes	12
3.9.2 The <code>occur</code> attribute	13
3.10 <b>PossibleSpeciesFeatureValue</b>	13
3.10.1 The <code>id</code> and <code>name</code> attributes	13
3.10.2 The <code>numericValue</code> attribute	13
3.11 <b>SpeciesTypeInstance</b>	13
3.11.1 The <code>id</code> and <code>name</code> attributes	14
3.11.2 The <code>speciesType</code> attribute	14
3.11.3 The <code>compartmentReference</code> attribute	14
3.11.4 The <code>occur</code> attribute	15
3.12 <b>SpeciesTypeComponentIndex</b>	15
3.12.1 The <code>id</code> attribute	16
3.12.2 The <code>component</code> attribute	16
3.12.3 The <code>identifyingParent</code> attribute	16
3.12.4 The <code>occur</code> attribute	17
3.12.5 <b>ListOfDenotedSpeciesTypeComponentIndexes</b>	17
3.13 <b>DenotedSpeciesTypeComponentIndex</b>	17
3.13.1 The <code>speciesTypeComponentIndex</code> attribute	17
3.13.2 Reference a <i>component</i> in a <i>speciesType</i> or a <i>species</i>	17
3.14 <b>InSpeciesTypeBond</b>	17
3.14.1 The <code>id</code> and <code>name</code> attributes	17
3.14.2 The <code>bindingSite1</code> and <code>bindingSite2</code> attributes	18
3.15 Uniqueness of <b>SpeciesType</b> definitions	18
3.16 <b>Species</b>	20
3.16.1 The <code>speciesType</code> attribute	20
3.16.2 <b>ListOfOutwardBindingSites</b>	20
3.16.3 The <b>ListOfSpeciesFeatures</b> container component	21
3.17 <b>OutwardBindingSite</b>	21
3.17.1 The <code>bindingStatus</code> attribute	21

3.17.2 The <code>component</code> attribute	21
3.17.3 Example	21
3.18 <b>SpeciesFeature</b>	22
3.18.1 The <code>id</code> attribute	23
3.18.2 The <code>speciesFeatureType</code> attribute	23
3.18.3 The <code>occur</code> attribute	23
3.18.4 The <code>component</code> attribute	23
3.18.5 <b>SpeciesFeatureValue</b>	24
3.18.6 Example	24
3.19 "Fully defined" <i>species</i> and the mapping to "pattern" <i>species</i>	26
3.20 <b>Reaction</b>	27
3.20.1 The <code>isIntraSpeciesReaction</code> attribute	27
3.21 <b>SimpleSpeciesReference</b>	28
3.22 <b>SpeciesReference</b>	29
3.22.1 <b>ListOfSpeciesTypeComponentMapsInProduct</b>	29
3.23 <b>SpeciesTypeComponentMapInProduct</b>	29
3.23.1 The <code>reactant</code> attribute	29
3.23.2 The <code>reactantComponent</code> attribute	29
3.23.3 The <code>productComponent</code> attribute	30
3.23.4 <b>ListOfSpeciesFeatureChanges</b>	30
3.23.5 Example	30
3.24 <b>SpeciesFeatureChange</b>	33
3.24.1 The <code>id</code> attribute	33
3.24.2 The <code>reactantSpeciesFeature</code> attribute	33
3.24.3 The <code>productSpeciesFeature</code> attribute	33
3.24.4 Example	33
3.25 The <code>outwardBindingSites</code> and <i>speciesFeatures</i> in "don't care" state in a reaction <i>product</i>	34
3.26 Extended <i>ci</i> element in <b>Math</b>	35
3.26.1 The <code>speciesReference</code> attribute	35
3.26.2 The <code>representationType</code> attribute	37
4 <b>Examples</b>	38
4.1 Example: <b>Compartment</b> , <b>SpeciesType</b> and <b>Species</b>	38
4.2 <i>Simmune</i> example: the <i>Ecad</i> model	39
4.3 The "microtubule" example in the previous <i>Multi</i> proposal (2010)	47
4.4 A <i>BioNetGen</i> example from its user manual	50
4.5 Mappings between <i>reactants</i> and <i>products</i> in a published model	62
4.6 Example from <i>Kappa</i> 's documentation	69
A <b>Validation of SBML documents using <i>Multi</i> constructs</b>	72
<b>Acknowledgments</b>	73
<b>References</b>	74

# 1 Introduction

This *Multi* package provides an extension of SBML Level 3 [Hucka et al. (2010)] that supports encoding models with molecular complexes that have multiple components and can exist in multiple states and in multiple compartments. One of its goals also is to provide a platform for sharing models based on the specifications of bi-molecular interactions and the rules governing such interactions [Angermann et al. (2012); Faeder et al. (2006); Feret et al. (2009); Zhang et al. (2013)]. This specification covers the goals and features described in the previous *Multi* proposal [Novère and Oellrich (2010)] for extending SBML to carry the information for **multistate multicomponent species** with revised data structure. In addition, this specification includes the feature for **multicompartment species** as described in the most recent releases of the *Multi* proposal [Zhang and Meier-Schellersheim (2013a), Zhang et al. (2012)].

## 1.1 Proposal corresponding to this package specification

This specification for *Multi* in SBML Level 3 Version 1 is mainly based on the new *Multi* proposal (May 2013, Rev 280) located at the following URL:

<http://goo.gl/2375K>

## 1.2 Package dependencies

The *Multi* package has no dependencies on other SBML Level 3 packages.

## 1.3 Document conventions

UML 1.0 notation is used in this document to define the constructs provided by this package. Colors in the diagrams carry the following additional information for the benefit of those viewing the document on media that can display color:

- **Black** Items colored black are components taken unchanged from their definitions in the SBML Level 3 Core specification document.
- **Green** Items colored green are components that exist in SBML Level 3 Core, but are extended by this package. Class boxes are also drawn with dashed lines to further distinguish them.
- **Blue** Items colored blue are new components introduced in this package specification. They have no equivalent in the SBML Level 3 Corespecification.

For other matters involving the use of UML, XML and typographical conventions, this document follows the conventions used in the SBML Level 3 Core specification document [Hucka et al. (2010)].

For simplicity, “...” in all example code refers to some unspecified code content, that is not important for the purpose of illustrating the issue at hand.

## 2 Background and context

Rule-based modeling approaches (*BioNetGen*[Faeder et al. (2009)], *Kappa*[Danos and Laneve (2004)], and *Sim-mune*[Angermann et al. (2012); Meier-Schellersheim et al. (2006)]) define rules for interactions between pairs of molecule components, specifying how the interactions depend on particular states of the molecules (pattern) and their locations in specific compartments. In order to generate networks of biochemical reactions these rules are applied to the molecular components of the systems to be modeled, either at the beginning of the modeling (simulation) process or “on the fly” (as molecule complexes emerge from the interaction rules). Expressing such rule-based reaction networks using the concepts of **Species** and **Compartment** in SBML (L3 core and L2) can be difficult for rules and molecule sets that lead to large numbers of resulting molecular complexes. It would therefore be desirable to have an SBML standard for encoding rule-based models using their “native” concepts for describing reactions instead of having to apply the rules and unfold the networks prior to encoding in an SBML format.

We proposed a revised proposal of the *Multi*: “Multistate, Multicomponent and Multicompartment Species Package for SBML Level 3” (abbreviated as *Multi*)[Zhang et al. (2012) and Zhang and Meier-Schellersheim (2013a)] which takes the previous *Multi* proposal [Novère and Oellrich (2010)] as base and addresses some issues arising from a rule-based modeling point of view. This specification documentation has been developed from the most recent release of the new *Multi* proposal [Rev 280, Zhang and Meier-Schellersheim (2013a)] with several new or updated features, including:

- **SpeciesTypeComponentIndex** under the **SpeciesType** class to replace **SpeciesTypeInstanceReference** in the proposal.
- Configuration for multiple occurrences of **SpeciesFeatureType**.
- New **numericValue** attribute in **PossibleSpeciesFeatureValue**.
- Concentration summation of pattern *species* accomplished with the **representationType** attribute of the *ci* element in **Math**.

### 2.1 Past work on this problem or similar topics

- Nicolas Le Novère and Anika Oellrich proposed the previous *Multi* proposal (2010). That proposal provides a history of the development of the *Multi* package before 2010:
  - Andrew Finney was probably the first to formulate, in March 2001, proposed SBML extensions to support complex species, to be able to cover multistates species and species made up of graphs of components, as part of a collection of proposals for new SBML development. At the 3rd Workshop on Software Platforms for Systems Biology in June 2001, Nicolas Le Novère gave a presentation entitled Multistate molecules and complex objects proposing to extend Andrew’s multistate proposal.
  - Nicolas Le Novère and Tom Shimizu came up in July 2001 with an alternative proposal for encoding and using states in SBML. A slightly extended and corrected version of this proposal presented by Nicolas at the 5th Workshop on Software Platforms for Systems Biology in July 2002. Nicolas Le Novère, Tom Shimizu and Andrew Finney published a complete description of this extension in December 2002.
  - In March 2004, before the 2nd SBML hackathon, Andrew Finney published an updated proposal to encode complex species made up of several components. Planed as an extension for SBML Level 3, the document also described **SpeciesTypes** that would later be incorporated to SBML Level 2, from version 2 onward.
  - In October 2004, Michael Blinov published, together with Jim Fader, Byron Goldstein, Andrew Finney and Bill Hlavacek, an alternative proposal for encoding multi-component species, that also contained some possibilities of encoding multistate features.

- Anika Oellrich started to implement a new SBML L2 support for StochSim in spring 2007, storing multistate information in proprietary annotations. This led in June 2007 to a proposal for Level 3 by Le Novegravere and Oellrich, meant to work in conjunction with 2004 Finney’s multicomponents proposal. The proposal was presented at the 12th SBML forum meeting. A light correction was published in December 2007.
  - Also at the 12th SBML forum meeting, Michael Blinov presented an updated version of his proposal. He later published two proposals for SBML L3, one with a hierarchical speciesTypes structure and one with a non-hierarchical speciesTypes structure.
  - On December 6 and 7, 2007, an SBML Focused Videoconference was held, which launched the effort to develop the Level 3 package *Multi*.
- In August 2012, Fengkai Zhang from the *Simmune* group presented “Draft for discussion SBML Proposals for Revised Multi, Simple Spatial and Multi-Spatial Extensions” at COMBINE 2012[Zhang et al. (2012)]. The three proposals cover the goals and scope of the previous *Multi* proposal (2010), revise it and add some new features that improve usage of the proposal for rule-based approaches.
- Based on the discussions and suggestions received during COMBINE 2012 as well as on feedback from the SBML discussion forum, the new *Multi* proposal (Dec 2012, Rev 221) was released to the SBML-Multi community, which integrates and covers most of the features in the three previous proposals of Aug 2012.
- In May 2013, a new reversion of the *Multi* proposal was released before the meeting of HARMONY 2013. The extended **Compartment** class and its related classes have been reorganized. All optional boolean attributes have been removed/replaced. A new optional *Multi* attribute, “**whichValue**”, was added to the **ci** elements in **KineticLaw** to identify the sources of *species*. (Lucian Smith gave many comments/suggestions about this proposal and Willam Hlavacek gave thoughtful feedback about the *BioNetGen* example in this proposal). This revision 280 was presented at HARMONY 2013 with plus new features to configure multiple occurrences of **SpeciesFeatureType** [Zhang and Meier-Schellersheim (2013b)]. Several new or revised features were discussed during and after HARMONY 2013, including multiple occurrences of **SpeciesFeatureType**, multiple copies of **SpeciesTypeInstance**, the **numericValue** attribute for **PossibleSpeciesFeatureValue** and concentration summation of pattern *species*. These features are covered or updated in this specification.

## 3 Package syntax and semantics

This section contains a definition of the syntax and semantics of the *Multi* package for SBML Level 3 Core.

### 3.1 Namespace URI and other declarations necessary for using this package

The following is the namespace URI for this version of the *Multi* package for SBML Level 3 Core:

```
"http://www.sbml.org/sbml/level3/version1/multi/version1"
```

In addition, SBML documents using a given package must indicate whether the package can be used to change the mathematical interpretation of a model. This is done using the attribute **required** on the `<sbml>` element in the SBML document. For the *Multi* package, the value of this attribute must be **"true"**.

The following fragment illustrates the beginning of a typical SBML model using SBML Level 3 Core and this version of the *Multi* package:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">
```

### 3.2 Primitive data types

The *Multi* package uses a number of the primitive data types described in Section 3.1 of the SBML Level 3 Core [Hucka et al. (2010)] specification such as `SIId`, `SIIdRef`, `string`, `boolean`, `int` and `postiveInteger`, and adds three additional primitive types described below.

#### 3.2.1 Type `BindingStatus`

The `BindingStatus` primitive data type is used in the definition of the `OutwardBindingSite` class. `BindingStatus` is derived from type `string` and its values are restricted to be one of the following possibilities: **"bound"**, **"unbound"**, and **"either"**. Attributes of type `BindingStatus` cannot take on any other values. The meaning of these three values is discussed in the context of the `OutwardBindingSite` class in Section 3.17 on page 21.

#### 3.2.2 Type `Relation`

The `Relation` primitive data type is used in the definition of the `ListOfSpeciesFeatures` class. `Relation` is derived from type `string` and its values are restricted to be one of the following possibilities: **"and"**, **"or"**, and **"not"**. Attributes of type `Relation` cannot take on any other values. The meaning of these three values is discussed in the context of the `ListOfSpeciesFeatures` class in Section 3.16.3 on page 21.

#### 3.2.3 Type `RepresentationType`

The `RepresentationType` primitive data type is used in the extension of the `ci` element. `RepresentationType` is derived from type `string` and its values are restricted to be one of the following possibilities: **"sum"** or **"numericValue"**. If present, attributes of type `RepresentationType` cannot take on any other values. The meaning of these three values is discussed in the context of the `ci` element in Section 3.26 on page 35.

### 3.3 The new and extended classes in the *Multi* Package

The *Multi* package defines or extends the following object classes, `Model`, `ListOfSpeciesTypes`, `Compartment`, `ListOfCompartmentReferences`, `CompartmentReference`, `SpeciesType`, `ListOfSpeciesTypeInstances`, `ListOfSpeciesFeatureTypes`, `ListOfInSpeciesTypeBonds`, `ListOfSpeciesTypeComponentIndexes`, `SpeciesFeatureType`, `PossibleSpeciesFeatureValue`, `SpeciesTypeInstance`, `InSpeciesTypeBond`, `SpeciesTypeComponentIndex`, `ListOfDenotedSpeciesType`



[ComponentIndexes](#), [DenotedSpeciesTypeComponentIndex](#), [Species](#), [ListOfOutwardBindingSites](#), [ListOfSpeciesFeatures](#), [OutwardBindingSite](#), [SpeciesFeature](#), [SpeciesFeatureValue](#), [Reaction](#), [SimpleSpeciesReference](#), [SpeciesReference](#), [ListOfSpeciesTypeComponentMapsInProduct](#), [SpeciesTypeComponentMapInProduct](#), [ListOfSpeciesFeatureChanges](#), and [SpeciesFeatureChange](#).

All the classes in the *Multi* package are directly or indirectly derived from **SBase**, and **SBase** provides the ability to attach SBO terms as well as MIRIAM annotations, the semantics of a given class in the *Multi* package can be made more precise by referencing to external controlled vocabularies and ontologies.

Like the classes in SBML Level 3 Core, most new *Multi* classes have the attribute **id** (typically mandatory but not all, and of type **SId**), which serves as an identifier to provide a way to identify the class object. The identifier of a class object reference may or may not carry mathematical interpretation or be used in mathematical formulas, depending on its class and the class object referencing it. Unless specified, the **id** of a *Multi* class object should be unique within the *model*.

## 3.4 Model

The *Multi* package extends the **Model** class of SBML Level 3 Core and adds an optional [ListOfSpeciesTypes](#) child to **Model**. Figure 1 provides the UML diagram for the extended **Model** class.

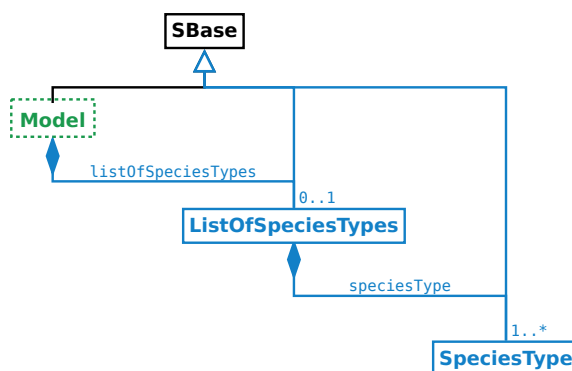


Figure 1: The extension of the **Model** class.

### 3.4.1 ListOfSpeciesTypes

The [ListOfSpeciesTypes](#) class is defined in Figure 1. If present, a [ListOfSpeciesTypes](#) object must contain at least one [SpeciesType](#) object.

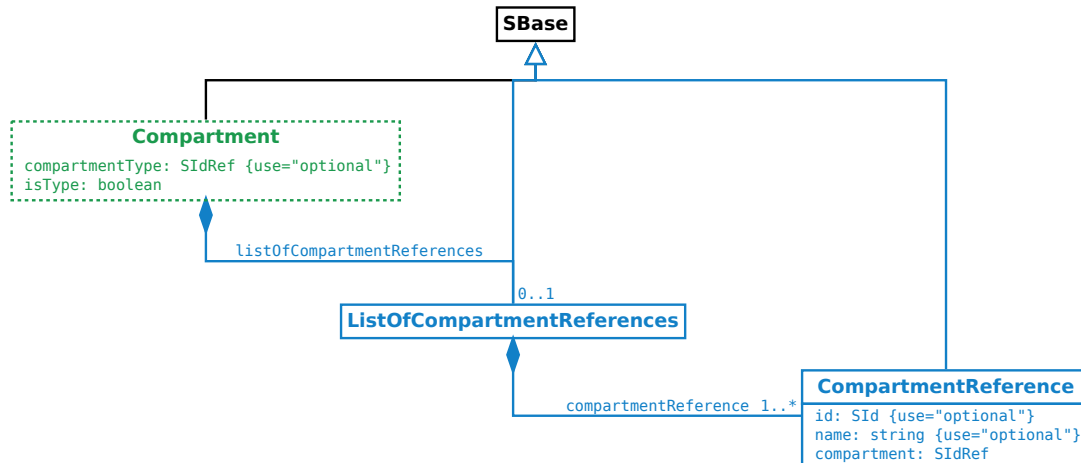
## 3.5 Compartment

A *compartment* in SBML Level 3 Core represents a bounded space in which species are located. In the *Multi* package, **Compartment** is extended. A *Multi compartment* can be a type that multiple *compartments* can map to. A *Multi compartment* can also be a composite *compartment* or a container that includes other *compartments*.

The extension of the **Compartment** class is defined in Figure 2 on the next page. The **Compartment** class is extended with a new required attribute **isType**, a new optional attribute **compartmentType** and an extra [ListOfCompartmentReferences](#) child. The example at Section 4.1 on page 38 illustrates the use of the **Compartment** class.

### 3.5.1 The isType attribute

The required attribute **isType**, of type boolean, on the **Compartment** class serves to provide a way to indicate whether the *compartment* is a *compartment type*.



**Figure 2:** The definitions of **Compartment**, **ListOfCompartmentReferences** and **CompartmentReference**

A *compartment* is a *compartment type* if the value of its **isType** attribute is “true”. A *compartment type* is a template (in the sense of prototype) for all *compartments* referencing it (via **compartmentType** attributes). A *species* directly referencing a *compartment type* is not a “fully defined” *species* (see [Section 3.19 on page 26](#)).

If the value of the **isType** attribute is “false”, the *compartment* is a “not-a-type” *compartment*, and it is similar to a SBML core *compartment* except it can reference a *compartment type* and can have a **ListOfCompartmentReferences** child.

### 3.5.2 The **compartmentType** attribute

The optional attribute **compartmentType**, of type **SIDRef**, is used for a “not-a-type” *compartment* to reference a *compartment type*. A *compartment type* can not have the **compartmentType** attribute defined.

### 3.5.3 **ListOfCompartmentReferences**

The **ListOfCompartmentReferences** class is defined in [Figure 2](#), and must have one or more **CompartmentReference** children.

## 3.6 **CompartmentReference**

The **CompartmentReference** class is defined in [Figure 2](#). It has two optional attributes **id** and **name**, and a required attribute **compartment**.

### 3.6.1 The **id** and **name** attributes

The optional **id** attribute, of type **SID**, serves to provide a way to identify the *compartmentReference*. **CompartmentReference** also has an optional **name** attribute, of type **string**.

### 3.6.2 The **compartment** attribute

The required **compartment** attribute, of type **SIDRef**, serves to provide a way to reference a *compartment*.

## 3.7 The relationship of **Compartment**, **CompartmentReference** and **ListOfCompartmentReferences**

In a *listOfCompartmentReferences*, different *compartmentReferences* must reference, directly or indirectly, different “not-a-type” *compartments* which can be of the same *compartment type*. See the *compartments* in the example in

Section 4.1 on page 38.

All *compartments* referenced by a *listOfCompartmentReferences* must have the values of their **isType** attributes the same as that in the parent *compartment* of the *listOfCompartmentReferences*. For example, a *compartment* “A” with **isType**=“true” has a *listOfCompartmentReferences* with two children referencing *compartment* “A1” and *compartment* “A2”. Then, “A1” and “A2” must have **isType**=“true”.

### 3.8 SpeciesType class

The **SpeciesType** class is defined in Figure 3 and serves to provide backbone structures for *species*. **SpeciesType** has two required attributes, **id** and **isBindingSite**, two optional attributes, **name** and **compartment** and four optional children, *listOfSpeciesFeatureTypes*, *listOfSpeciesTypeInstances*, *listOfInSpeciesTypeBonds* and *listOfSpeciesTypeComponentIndexes*.

The *listOfSpeciesTypeInstances* child provides a way to define multicomponents which are instances of other *speciesTypes*. A *speciesType* can be defined as a **bindingSite**, and therefore its instance can further define the **bindingStatus**. The *listOfSpeciesFeatureTypes* child and its *speciesFeatureTypes* set up a framework for the referencing *species* or the instances of the *speciesType* to be able to have multistates. The *listOfSpeciesTypeComponentIndexes* provides a flexible way to reference any **component** in a *speciesType*.

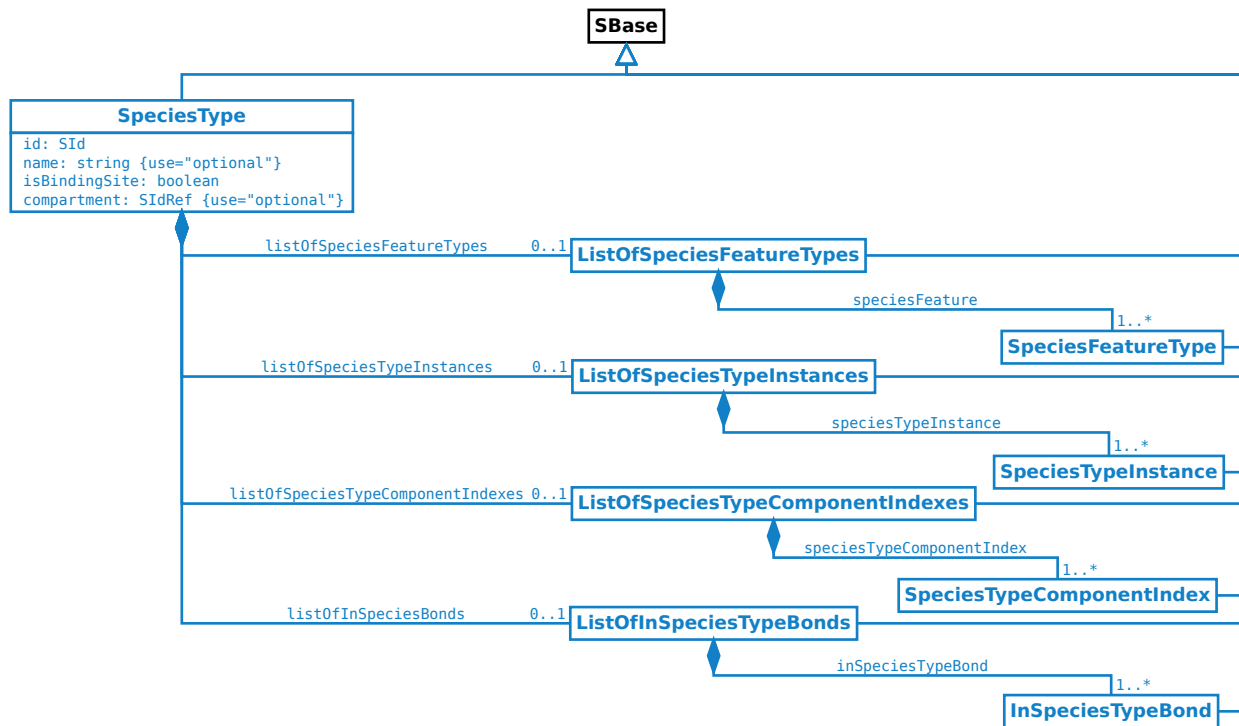


Figure 3: The definition of the **SpeciesType** class.

#### 3.8.1 The id and name attributes

The required **id** attribute, of type **SId**, on the **SpeciesType** class serves to provide a way to identify the *speciesType*. **SpeciesType** also has an optional **name** attribute, of type **string**.

### 3.8.2 The isBindingSite attribute

**SpeciesType** has a required attribute **isBindingSite** of type `boolean`. This attribute is used to indicate whether a *speciesType* is a **bindingSite**. If the value of the **isBindingSite** attribute is “`true`”, the instance of or the *species* referencing to the *speciesType* is a **bindingSite** and can participate a binding internally and explicitly in an **InSpeciesTypeBond** object, or externally and implicitly defined by an **OutwardBindingSite** object.

 *Note:*

*In the Multi package, a **bindingSite** can participate one binding at a time. That means a **bindingSite** can not bind two partners at the same time. A binding is one-to-one relationship.*

### 3.8.3 The compartment attribute

**SpeciesType** has an optional attribute **compartment**, of type `SIIdRef`, to be used to identify the *compartment* where the *speciesType* is located. The attribute value must be the identifier of an existing **Compartment** object in the *model*. If present, it must have no conflict with the **compartment** attributes of the referencing *species* (see [Section 3.16 on page 20](#)) and the **compartmentReference** attributes of its instances (see [Section 3.11.3 on page 14](#)). The example in [Section 4.1 on page 38](#) demonstrates the usage of this attribute.

### 3.8.4 ListOfSpeciesFeatureTypes

The **ListOfSpeciesFeatureTypes** class is defined in [Figure 3 on the previous page](#), and must have one or more **SpeciesFeatureType** children.

### 3.8.5 ListOfSpeciesTypeInstances

The **ListOfSpeciesTypeInstances** class is defined in [Figure 3 on the preceding page](#), and must have one or more **SpeciesTypeInstance** children.

### 3.8.6 ListOfInSpeciesTypeBonds

The **ListOfInSpeciesTypeBonds** class is defined in [Figure 3 on the previous page](#), and must have one or more **InSpeciesTypeBond** children.

### 3.8.7 ListOfSpeciesTypeComponentIndexes

The **ListOfSpeciesTypeComponentIndexes** class is defined in [Figure 3 on the preceding page](#), and must have one or more **SpeciesTypeComponentIndex** children.

## 3.9 SpeciesFeatureType

The **SpeciesFeatureType** class is defined in [Figure 4 on the next page](#), and serves to provide frameworks or templates to define the **SpeciesFeature** objects. **SpeciesFeatureType** has two required attributes **id** and **occur**, an optional attribute **name**, one or more *possibleSpeciesFeatureValues*. Multiple **PossibleSpeciesFeatureValue** children permit constructing multistate *species* via the **SpeciesFeature** class objects under the *species*’ **ListOfSpeciesFeatures** object.

### 3.9.1 The id and name attributes

The required **id** attribute, of type `SIId`, serves to provide a way to identify the *speciesFeatureType*. Its value must be unique within its direct parent *speciesType*. When a *speciesFeatureType* is referenced by a **SpeciesFeature** object, the **SpeciesTypeComponentIndex** object indexing its containing *component* can be used to avoid ambiguity. See the example in [Section 3.23.5 on page 30](#).

**SpeciesFeatureType** also has an optional **name** attribute, of type `string`.

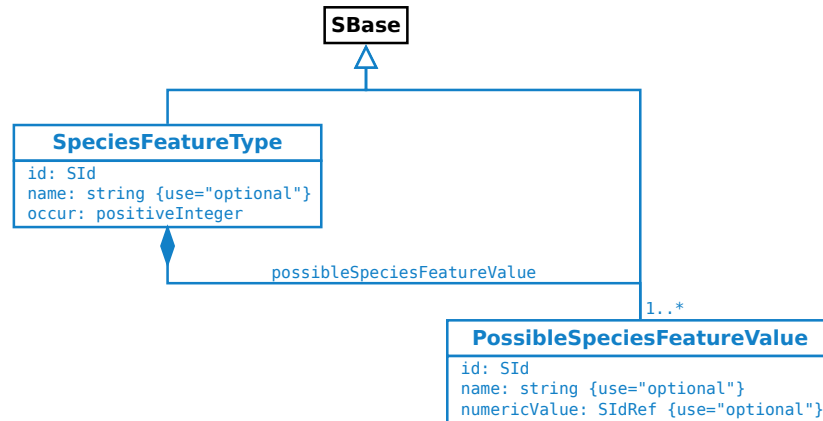


Figure 4: The definitions of **SpeciesFeatureType** and **PossibleSpeciesFeatureValue** classes.

### 3.9.2 The occur attribute

**SpeciesFeatureType** has a required attribute `occur`, of type `positiveInteger`, used to indicate the number of instances of the *speciesFeatureType*. This attribute can be used to infer the number of the instances in “don’t care” state with the use of the `occur` attribute in the referencing *speciesFeature* (also see [Section 3.18.3 on page 23](#)).

## 3.10 PossibleSpeciesFeatureValue

The **PossibleSpeciesFeatureValue** is defined in [Figure 4](#), and is used to define the possible values a *speciesFeature* can take. It has a required attribute `id` and two optional attributes `name` and `numericValue`.

### 3.10.1 The id and name attributes

The required `id` attribute, of type `SId`, on the **PossibleSpeciesFeatureValue** class serves to provide a way to identify the *possibleSpeciesFeatureValue*. Its value must be unique within the *speciesType*.

If the `id` of a *speciesFeature* is the content of a *ci* element in a MathML expression, it can either represent the `numericValue` (when the *ci* has `representationType` = “`numericValue`”) or the `count` of the feature instances (default) which have this value. Also see the example at [Section 3.26.2 on page 37](#).

**PossibleSpeciesFeatureValue** also has an optional `name` attribute, of type `string`.

### 3.10.2 The numericValue attribute

**PossibleSpeciesFeatureValue** has an optional attribute `numericValue` to be used to provide a reference to a numeric value that the **PossibleSpeciesFeatureValue** object can have. This attribute has type of `SIdRef`, and the value must be the identifier of a **Parameter** object in the **Model**. The numeric value along with the unit can be defined in the **Parameter** object.

The modeler can either use the identifier of the **Parameter** object, or the identifier of the **PossibleSpeciesFeatureValue** object (with *ci*’s `representationType` and `speciesReference` attribute) as the content of a *ci* element to represent its `value` in MathML expressions in SBML.

## 3.11 SpeciesTypeInstance

The **SpeciesTypeInstance** class serves to provide a way to construct *speciesTypes* and *species* with multiple components. A *speciesType* can contain a list of instances of other *speciesTypes* which can also have their own *speciesType* instances, so the complete structure of a *speciesType* can be like a tree. A *speciesType* can not contain an instance of any other *speciesType* that already contains the instance of it. In other words, circular references are not allowed

when constructing *speciesTypes*. For example, if a *speciesType* “A” contains the instance of another *speciesType* “B”, “B” must not contain the instance of “A” anywhere in the complete structure of “B”.

The **SpeciesTypeInstance** class is defined in Figure 5. It has three required attributes, **id**, **speciesType** and **occur**, and two optional attributes **name** and **compartmentReference**.

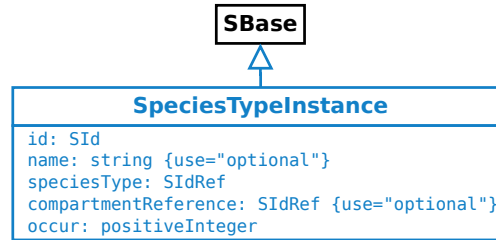


Figure 5: The definition of the **SpeciesTypeInstance** class

### 3.11.1 The id and name attributes

The required **id** attribute, of type **SId**, on the **SpeciesTypeInstance** class serves to provide a way to identify the *speciesTypeInstance*. The identifier of a *speciesTypeInstance* must be unique within its direct parent **SpeciesType** object.

**SpeciesTypeInstance** also has an optional **name** attribute of type **string**.

### 3.11.2 The speciesType attribute

**SpeciesTypeInstance** has a required attribute **speciesType**, of type **SIdRef**, is used to reference a *speciesType*.

### 3.11.3 The compartmentReference attribute

**SpeciesTypeInstance** has an optional attribute **compartmentReference**, of type **SIdRef**, can be used to indicate which sub-compartment in a composite *compartment* the *speciesTypeInstance* is located in.

For example, a *compartment* “cA” has two sub-compartments “cB1” (referenced by *compartmentReference* “crB1”) and “cB2” (referenced by *compartmentReference* “crB2”) of the same *compartment* type “cB”. A *speciesType* “stA” has two *speciesTypeInstances* “stiB1” and “stiB2” of the same *speciesType* “stB”. The *speciesType* “stA” references the *compartment* “cA” and the *speciesType* “stB” references the *compartment* “cB”. The *speciesTypeInstance* “stiB1” is located in “cB1” via the *compartmentReference* “crB1” and the *speciesTypeInstance* “stiB2” is located in “cB2” via the *compartmentReference* “crB2”. The SBML code can be as follows:

```

<listOfCompartments>
  <compartment id="cB" multi:isType="true" ... />
  <compartment id="cB1" multi:isType="false" multi:compartmentType="cB" ... />
  <compartment id="cB2" multi:isType="false" multi:compartmentType="cB" ... />
  <compartment id="cA" multi:isType="false" ... >
    <multi:listOfCompartmentReferences>
      <multi:compartmentReference multi:id="crB1" multi:compartment="cB1" />
      <multi:compartmentReference multi:id="crB2" multi:compartment="cB2" />
    </multi:listOfCompartmentReferences>
  </compartment>
</listOfCompartments>
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stB" multi:compartment="cB" ... />
  <multi:speciesType multi:id="stA" multi:compartment="cA" ... >
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="stiB1" multi:speciesType="stB"
        multi:compartmentReference="crB1" ... />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
  
```

```

<multi:speciesTypeInstance multi:id="stiB2" multi:speciesType="stB"
  multi:compartmentReference="crB2" ... />
</multi:listOfSpeciesTypeInstances>
</multi:speciesType>
</multi:listOfSpeciesTypes>

```

### 3.11.4 The occur attribute

**SpeciesTypeInstance** has a required attribute **occur**, of type **positiveInteger**, is used to indicate the total number of occurrences of the *speciesTypeInstance*. In the most cases, the value of this attribute will be “1”. If the value is more than “1”, different occurrences may have different **bindingStatus** and/or *speciesFeatures* in a referencing *species* with the use of the **occur** attribute in **SpeciesTypeComponentIndex**.

## 3.12 SpeciesTypeComponentIndex

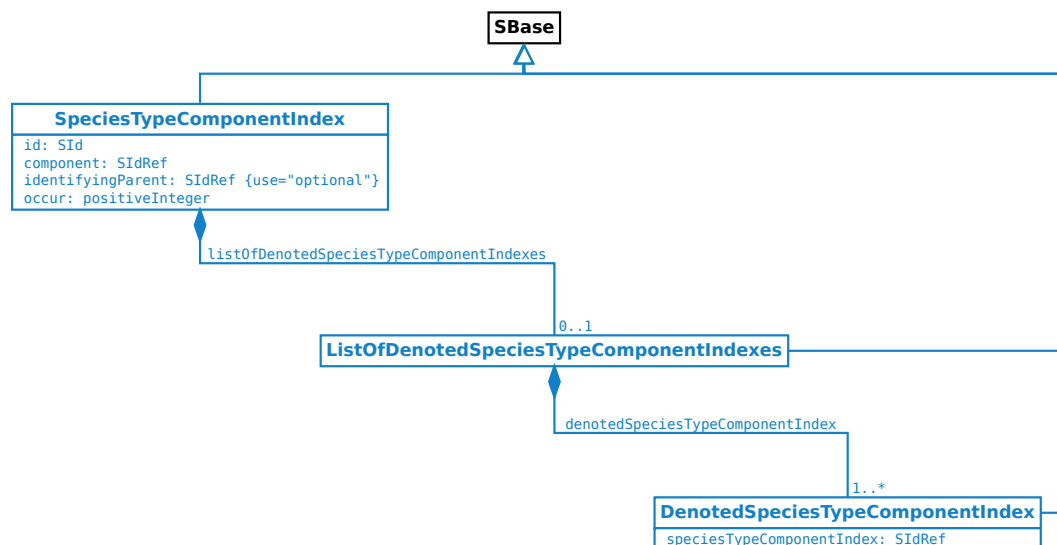
The **SpeciesTypeComponentIndex** class provides a way to identify or index a *component* within a *speciesType*. A *speciesTypeComponentIndex* can be referenced by other class objects, such as **InSpeciesTypeBond**, **OutwardBindingSite**, **SpeciesFeature** or **SpeciesTypeComponentMapInProduct** objects, which needs to identify a *component* in a particular *speciesType*.

The **SpeciesTypeComponentIndex** class is defined in Figure 6. It has three required attributes, **id**, **component** and **occur**, and an optional attribute **identifyingParent**.

The **SpeciesTypeComponentIndex** has an optional **ListOfDenotedSpeciesTypeComponentIndexes** child. The *listOfDenotedSpeciesTypeComponentIndexes*, if present, specifies a group of *speciesTypeComponentIndexes* that permit this *speciesTypeComponentIndex* to select from.

A *speciesTypeComponentIndex* can have the **ListOfDenotedSpeciesTypeComponentIndexes** child only when the **occur** of the indexed *speciesTypeInstance* *component* is larger than “1”.

When a *speciesTypeInstance* has multiple occurrences (**occur** > “1”), all referencing *speciesTypeComponentIndexes* with the same **identifyingParent** and no *listOfDenotedSpeciesTypeComponentIndexes* within a *speciesType* are mutually exclusive (ensuring unique indexing). See the examples in Section 4.4 on page 50 and Section 4.5 on page 62.



**Figure 6:** The definitions of the **SpeciesTypeComponentIndex** class and the **DenotedSpeciesTypeComponentIndex** class



### 3.12.1 The id attribute

The **id** attribute, of type **SIId**, provides a way to identify the *speciesTypeComponentIndex*. The value must be unique within the parent *speciesType*.

### 3.12.2 The component attribute

The **component** attribute, of type of **SIIdRef**, references a *speciesTypeInstance* in the *speciesType* or the *speciesType* itself. The value of this attribute can be the **id** of a *speciesTypeInstance* or a *speciesTypeComponentIndex* that is defined in the *speciesType* of a *speciesTypeInstance*. If the **component** attribute takes the **id** of another *speciesTypeComponentIndex*, their **occur** attributes must have the same value and the referencing *speciesTypeComponentIndex* can not define a *listOfDenotedSpeciesTypeComponentIndexes* (but the referenced *speciesTypeComponentIndex* can). The example in [Section 3.23.5 on page 30](#) illustrates the use of the **component** attribute.

### 3.12.3 The identifyingParent attribute

The **component** attribute itself may not be sufficient to uniquely reference a *component* in a *speciesType*. The **identifyingParent** attribute provides assistance for the identification of a *component*. It references a parent of the *component* and the value can be the **id** of a *speciesTypeInstance*, a *speciesType*, or a *speciesTypeComponentIndex*.

This example illustrates the use of the **identifyingParent** attribute. There are three *speciesTypes* “**stA**”, “**stB**” and “**stC**”. The *speciesType* “**stB**” contains two *speciesTypeInstances* “**C1**” and “**C2**” of the same *speciesType* “**stC**”. The *speciesType* “**stA**” contains two *speciesTypeInstances* “**B1**” and “**B2**” of the same *speciesType* “**stB**”. The *speciesType* “**A**” may be required to index every “**C1**” and “**C2**” by its [ListOfInSpeciesTypeBonds](#) child or referencing *species*. The following SBML code demonstrates how to do the indexing with assistance from the **identifyingParent** attribute.

```
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stC" ... />
  <multi:speciesType multi:id="stB" ... >
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="C1" multi:speciesType="stC" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="C2" multi:speciesType="stC" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="stA" ... >
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="B1" multi:speciesType="stB" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="B2" multi:speciesType="stB" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
      <multi:speciesTypeComponentIndex multi:id="B1C1"
        multi:component="C1" multi:identifyingParent="B1" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="B1C2"
        multi:component="C2" multi:identifyingParent="B1" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="B2C1"
        multi:component="C1" multi:identifyingParent="B2" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="B2C2"
        multi:component="C2" multi:identifyingParent="B2" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
  </multi:speciesType>
  ...
</multi:listOfSpeciesTypes>
```

In the *speciesType* “**stA**”, “**B1C1**” identifies the “**C1**” in “**B1**” and “**B2C1**” identifies the “**C1**” in “**B2**”. Similarly, “**B1C2**” identifies the “**C2**” in “**B1**” and “**B2C2**” identifies “**C2**” in “**B2**”.



### 3.12.4 The occur attribute

**SpeciesTypeComponentIndex** has a required attribute **occur**, of type `positiveInteger`, used to indicate the number of occurrences of the indexed **component**. When the indexed **component** is a *speciesTypeInstance* with **occur**="1", the value of this **occur** attribute must be "1". If the indexed component is a *speciesTypeInstance* with multiple occurrences, the value of this **occur** attribute must be less than or equal to the value of the **occur** attribute of the indexed *speciesTypeInstance*.

### 3.12.5 ListOfDenotedSpeciesTypeComponentIndexes

The **ListOfDenotedSpeciesTypeComponentIndexes** class is defined in Figure 6 on page 15, and must have one or more **DenotedSpeciesTypeComponentIndex** children.

## 3.13 DenotedSpeciesTypeComponentIndex

The **DenotedSpeciesTypeComponentIndex** class can be used to define a *listOfDenotedSpeciesTypeComponentIndexes* for a **SpeciesTypeComponentIndex** object. **DenotedSpeciesTypeComponentIndex** has only one attribute **speciesTypeComponentIndex**.

### 3.13.1 The speciesTypeComponentIndex attribute

The required **speciesTypeComponentIndex** attribute, of type `SIRef`, is used to reference a *speciesTypeComponentIndex* that must partially (the **occur** of the *speciesTypeComponentIndex* is less than the **occur** of the indexed *speciesTypeInstance* **component**) reference the same *speciesTypeInstance*.

### 3.13.2 Reference a component in a speciesType or a species

In the *Multi* package, a **component** of a *speciesType* may be a *speciesTypeInstance* in the *speciesType* or the *speciesType* itself. This permits, for example, to define the **bindingStatus** of a **bindingSite** which may be a *speciesTypeInstance* in a *species* or a *speciesType* directly referenced by a *species*. The second case will be to reference a *speciesFeatureType* of a *speciesTypeInstance* in a *speciesType* or a *speciesType* itself.

In many cases, to reference a **component**, the **id** of the **component** will be sufficient and it is not necessary to create an index (*speciesTypeComponentIndex*). The example in Section 3.12.3 on the preceding page illustrates two equivalent ways to reference a **component**, for example, the "B1" **component** in the "stA" *speciesType*. There are two situations in which the creation of a *speciesTypeComponentIndex* cannot be avoided:

- A *speciesType* (indirectly) has two *speciesTypeInstances* that have the same **id**.
- The **occur** of a *speciesTypeInstance* is larger than "1".

## 3.14 InSpeciesTypeBond

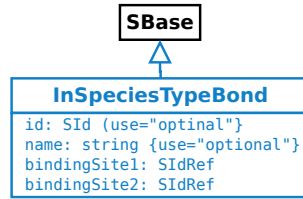
An *inSpeciesTypeBond* defines a bond existing within a *speciesType*. The bond therefore exists in every *species* that references the *speciesType*.

The **InSpeciesTypeBond** class is defined in Figure 7 on the next page. It has two optional attributes, **id** and **name**, and two required attributes, **bindingSite1** and **bindingSite2**.

### 3.14.1 The id and name attributes

The optional **id** attribute, of type `SIId`, provides a way to identify the *inSpeciesTypeBond*. If present, the value of the **id** attribute must be unique within its parent *speciesType*.

**InSpeciesTypeBond** also has an optional **name** attribute, of type `string`.

Figure 7: The definition of the **InSpeciesTypeBond** class

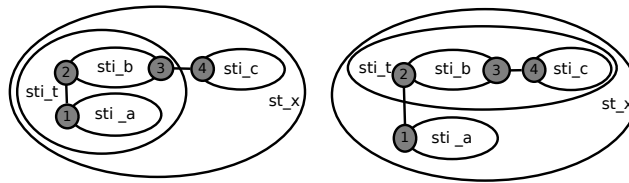
### 3.14.2 The bindingSite1 and bindingSite2 attributes

**InSpeciesTypeBond** has two required attributes, **bindingSite1** and **bindingSite2**, both of type **SIdRef**, used to reference two **bindingSites** in the *speciesType*. The referenced identifiers of the two **bindingSites** can be the ids of the *speciesTypeInstances* (**bindingSites**), or the ids of the **SpeciesTypeComponentIndex** objects indexing the **bindingSites**.

## 3.15 Uniqueness of **SpeciesType** definitions

In some special cases, it may be possible to define a **SpeciesType** object in multiple equivalent ways.

Figure 8 shows an example of a *speciesType* constructed in two different formats. The two “**st\_x**” *speciesTypes* in the diagram can be the results of different *reaction* paths, but they are equivalent and define the same *speciesType*.

Figure 8: Different formats of the same *speciesType*

Construct 1: The definition of *speciesType* “**st\_x**” on the left in Figure 8.

```

<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="st1" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st2" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st3" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st4" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_a" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_1" multi:speciesType="st1" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_b" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_2" multi:speciesType="st2" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="_3" multi:speciesType="st3" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_c" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_4" multi:speciesType="st4" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
  
```

```

<multi:speciesType multi:id="st_t" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="sti_a" multi:speciesType="st_a" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="sti_b" multi:speciesType="st_b" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="_1" multi:bindingSite2="_2" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
<multi:speciesType multi:id="st_x" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="sti_t" multi:speciesType="st_t" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="sti_c" multi:speciesType="st_c" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="_3" multi:bindingSite2="_4" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
</multi:listOfSpeciesTypes>

```

Construct 2: The definition of *speciesType* “st\_x” on the right in [Figure 8 on the preceding page](#).

```

<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="st1" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st2" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st3" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st4" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_a" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_1" multi:speciesType="st1" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_b" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_2" multi:speciesType="st2" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="_3" multi:speciesType="st3" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_c" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_4" multi:speciesType="st4" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_t" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="sti_b" multi:speciesType="st_b" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="sti_c" multi:speciesType="st_c" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="_3" multi:bindingSite2="_4" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
  <multi:speciesType multi:id="st_x" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="sti_a" multi:speciesType="st_a" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="sti_t" multi:speciesType="st_t" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="_1" multi:bindingSite2="_2" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
</multi:listOfSpeciesTypes>

```

This kind of ambiguity cannot be avoided for *speciesTypes* involving more than two subcomponents connected by

*inSpeciesTypeBonds*, for example, the *speciesType* referenced by the product *species* in an association *reaction*. It is up to the modeler (parser) to identify whether the two *speciesTypes* such as those in the example above are identical.

## 3.16 Species

A *species* in SBML Level 3 Core refers a pool of entities. A *species* in the *Multi* package is extended from a pool to a template or pattern which multiple pools may map to. An extended *species* can reference a *speciesType* that provides the backbone for the *species* such as *components* (including *bindingSites*) and *speciesFeatureTypes*. When referencing a *speciesType*, a *species* can be further defined with regard to the binding statuses of its *outwardBindingSites* and the *speciesFeatures*. With the options to have variable values selected, such as “either” for the *bindingStatus* attribute and multiple *possibleSpeciesFeatureValues* for a *speciesFeature*, an extended *species* can work as a template or pattern how *species* participate in *reactions*.

The extension of the **Species** class is illustrated in Figure 9. The extended **Species** class has a new optional attribute **speciesType**, and two extra optional **ListOfOutwardBindingSites** and **ListOfSpeciesFeatures** children. A *species* may have a *listOfOutwardBindingSites* child and/or a *listOfSpeciesFeatures* child only when its **speciesType** attribute has been defined.

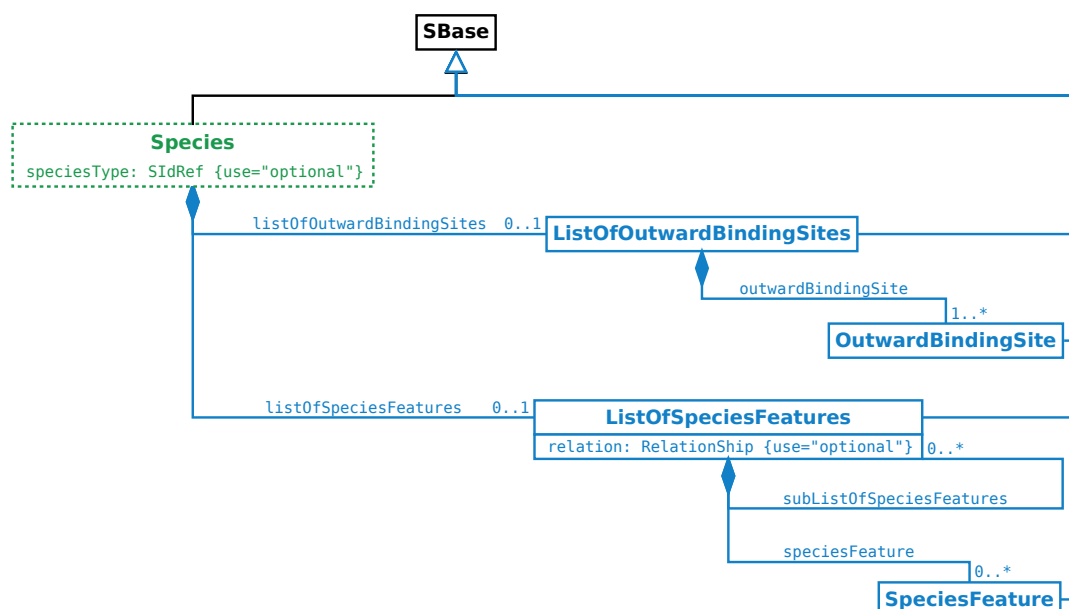


Figure 9: The extension of the **Species** class

### 3.16.1 The speciesType attribute

The optional attribute **speciesType**, of type **SIdRef**, references a **SpeciesType** object.

### 3.16.2 ListOfOutwardBindingSites

The **ListOfOutwardBindingSites** class is defined in Figure 9, and can only be defined when the **speciesType** attribute is defined. If present, it must have one or more **OutwardBindingSite** children.

**Note:**

The *listOfOutwardBindingSites* of a *species* is not necessary to list all the *outwardBindingSites* (the *bindingSites* not involved in any *inSpeciesTypeBond*) defined by the referenced *speciesType*. If an *outwardBindingSite* is not listed in the *listOfOutwardBindingSites*, the value of its *bindingStatus* is “either”, in other words, the *bindingSite* is in a

“don’t care” state.

### 3.16.3 The ListOfSpeciesFeatures container component

The **ListOfSpeciesFeatures** class is defined in Figure 9 on the previous page, and can only be defined when the **speciesType** attribute is defined. If present, it must have one or more children. A child can be a **SpeciesFeature** object, or a **subListOfSpeciesFeatures**, which is a **ListOfSpeciesFeatures** object.

**ListOfSpeciesFeatures** has an optional attribute **relation**, of type **Relation**, to define the logic relationship among its children. The **relation** attribute can not be defined if a **ListOfSpeciesFeatures** object has only one child, and it must be defined if the **ListOfSpeciesFeatures** object has more than one children.

 Note:

*The listOfSpeciesFeatures of a species does not have to cover all the speciesFeatures corresponding to all speciesFeatureTypes (see Section 3.9 on page 12) of every component defined by the referenced speciesType. If a speciesFeatureType is defined and there is no speciesFeature explicitly referencing it, the species has an implicit speciesFeature having all the possibleSpeciesFeatureValues and “or” relationships between them. In other words, the implicit speciesFeature has a “don’t care” state for the species.*

The example at Section 3.18.6 on page 24 illustrates the usage of the **ListOfSpeciesFeatures** class.

## 3.17 OutwardBindingSite

The **OutwardBindingSite** class is defined in Figure 10. It has two required attributes, **bindingStatus** and **component**. A **bindingSite** not involved in any **InSpeciesTypeBond** object in the **speciesType** referenced by a **species** is an **outwardBindingSite**.

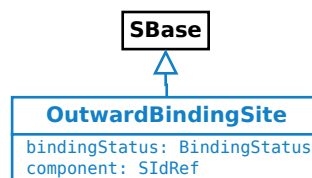


Figure 10: The definition of the **OutwardBindingSite** class

### 3.17.1 The bindingStatus attribute

The **bindingStatus** attribute takes a value of type **BindingStatus**.

### 3.17.2 The component attribute

The **component** attribute, of type **SIdRef**, references a **component** which is an **outwardBindingSite** in the **species**.

### 3.17.3 Example

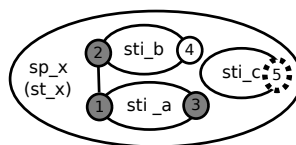


Figure 11: An example of **OutwardBindingSite**

Figure 11 illustrates the usage of the **OutwardBindingSite** class. The **species** “**sp\_x**” references the **speciesType** “**st\_x**”,

which has three *speciesTypeInstances* “sti\_a”, “sti\_b” and “sti\_c”. The *speciesTypeInstance* “sti\_a” has *bindingSites* “\_1” and “\_3”, the *speciesTypeInstance* “sti\_b” has *bindingSites* “\_2” and “\_4”, and the *speciesTypeInstance* “sti\_c” has *bindingSite* “\_5”. The *inSpeciesTypeBond* in “st\_x” involves two *bindingSites* “\_1” and “\_2”. The other three *bindingSites* “\_3”, “\_4” and “\_5”, in the *species* “sp\_x” are *outwardBindingSites*. The *outwardBindingSite* “\_3” is “bound” (filled circle with solid line in the diagram), the *outwardBindingSite* “\_4” is “unbound” (empty circle with solid line) and the *outwardBindingSite* “\_5” has binding status “either” (empty circle with dotted line). The corresponding SBML code would be as follows:

```
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="st_1" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_2" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_3" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_4" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_5" multi:isBindingSite="true" />
  <multi:speciesType multi:id="st_a" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_1" multi:speciesType="st_1" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="_3" multi:speciesType="st_3" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_b" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_2" multi:speciesType="st_2" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="_4" multi:speciesType="st_4" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_c" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="_5" multi:speciesType="st_5" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
  </multi:speciesType>
  <multi:speciesType multi:id="st_x" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="sti_a" multi:speciesType="st_a" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="sti_b" multi:speciesType="st_b" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="sti_c" multi:speciesType="st_c" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="_1" multi:bindingSite2="_2" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="sp_x" multi:speciesType="st_x">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="_3" multi:bindingStatus="bound" />
      <multi:outwardBindingSite multi:component="_4" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="_5" multi:bindingStatus="either" />
    </multi:listOfOutwardBindingSites>
  </species>
</listOfSpecies>
```

### 3.18 SpeciesFeature

The **SpeciesFeature** class is defined in Figure 12 on the following page. It has two optional attributes, *id* and *component*, and two required attributes, *speciesFeatureType* and *occur*. The **SpeciesFeature** class serves to define the state of a *component* in the *species* by selecting values from *possibleSpeciesFeatureValues*. A *speciesFeature* must have at least one **SpeciesFeatureValue** children. If a *speciesFeature* has multiple **SpeciesFeatureValue** child, the interpretation of the relationships between the *speciesFeatureValues* is “or”.

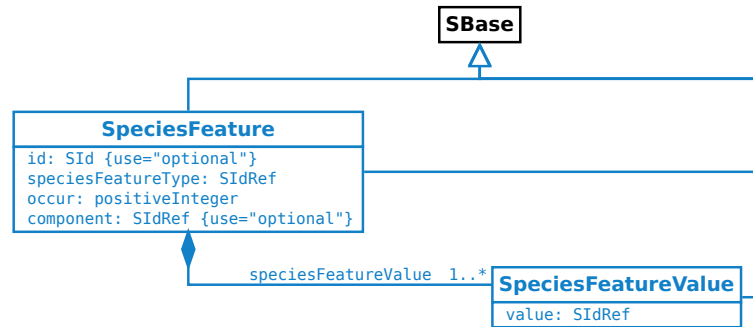


Figure 12: The definitions of the *SpeciesFeature* class and the *SpeciesFeatureValue* class

### 3.18.1 The id attribute

The optional **id** attribute, of type **SId**, can serve to provide a way to identify the *speciesFeature*. If present, the value must be unique within the *species*.

### 3.18.2 The speciesFeatureType attribute

**SpeciesFeature** has a required attribute **speciesFeatureType**, of type **SIdRef**, used to reference a *speciesFeatureType*.

### 3.18.3 The occur attribute

**SpeciesFeature** has a required attribute **occur**, of type of **positiveInteger**, used to define the number of instances of the referenced *speciesFeatureType*.

The value of the **occur** attribute can not be larger than the **occur** of the referenced *speciesFeatureType*. When a *speciesFeatureType* has multiple instances (*speciesFeatureType*'s **occur** > "1"), the *speciesFeature*'s **occur** attribute provides a way for a *species* to define the instances of the *speciesFeatureType* differently.

For example, a *speciesFeatureType* "ftA" has **occur**="2" and two *possibleSpeciesFeatureValues* "fva1" and "fva2". A *species* can be defined to have two *speciesFeatures* "sfA1" and "sfA2" both referencing "ftA". The *speciesFeature* "sfA1" has **occur**="1" and its value is "fva1". The *speciesFeature* "sfA2" has **occur**="1" and its value is "fva2".

If the **occur** of a *speciesFeature* is less than the **occur** of the referenced *speciesFeatureType*, the rest of the unspecified instances of the *speciesFeatureType* are in "don't care" state which means that the value of an unspecified instance can be any from the *possibleSpeciesFeatureValues*.

For example, a *speciesFeatureType* "phosphorylation" has two *possibleSpeciesFeatureValues* "phosphorylated" and "unphosphorylated" and the **occur** is "5". A *species* can be defined to have a *speciesFeature* of "phosphorylation" with the value of "phosphorylated" and the **occur** of "1". Then, the *species* is a pattern *species* with at least one "phosphorylated" site (the other four "phosphorylation" sites are in "don't care" state). This pattern *species* can be mapped by anyone of the "fully defined" *species* (see Section 3.19 on page 26) of the same type and with any of "1" to "5" phosphorylated sites.

### 3.18.4 The component attribute

The optional **component** attribute, of type **SIdRef**, can be used to indicate which *component* of the *species* the *speciesFeature* belongs to. It is required when the *component* can not be identified only based on the *speciesFeatureType* attribute.



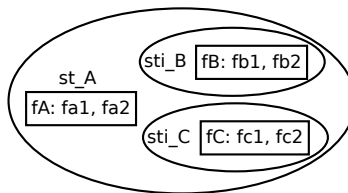
### 3.18.5 SpeciesFeatureValue

The **SpeciesFeatureValue** class is defined in Figure 12 on the preceding page. A *speciesFeatureValue* is one of the children of a **SpeciesFeature** object and serves to select a value for the *speciesFeature* from the *possibleSpeciesFeatureValues* defined in the referenced *speciesFeatureType*. The **SpeciesFeatureValue** class has only one attribute *value* of type *SIIdRef*, used to reference a **PossibleSpeciesFeatureValue** object.

### 3.18.6 Example

Figure 13 is an example *speciesType* to illustrate the usage of the **ListOfSpeciesFeatures** class and the **SpeciesFeature** class. The *speciesType* “**st\_A**” has a *speciesFeatureType* “**fa**” which has two *possibleSpeciesFeatureValues* “**fa1**” and “**fa2**”. The *speciesType* “**st\_A**” also has two children *speciesTypeInstances* “**sti\_B**” and “**sti\_C**”, which have the *speciesFeatureTypes* “**fb**” and “**fc**” respectively. The *speciesFeatureType* “**fb**” has *possibleSpeciesFeatureValues* “**fb1**” and “**fb2**”, and the *speciesFeatureType* “**fc**” has “**fc1**” and “**fc2**”. Here are several ways to construct the *listOfSpeciesFeatures* of a *species* referencing the *speciesType* “**st\_A**”:

- *listOfSpeciesFeatures*(relation=“and”, children=“fa1”, “fb1”, “fc1”) is a state:  
“[fa1] and [fb1] and [fc1]”
- *listOfSpeciesFeatures*(relation=“or”, children=  
    *subListOfSpeciesFeatures*(relation=“and”, children=“fa1”, “fb1”, “fc1”),  
    *subListOfSpeciesFeatures*(relation=“and”, children=“fa2”, “fb2”, “fc2”)  
) is a state:  
“[fa1] and [fb1] and [fc1]” or “[fa2] and [fb2] and [fc2]”
- *listOfSpeciesFeatures*(relation=“and”, children=  
    “fa1”,  
    *subListOfSpeciesFeatures*(relation=“not”, children=“fb1”, “fc1”)  
) is a state:  
“[fa1] and [fb1] and [fc2]” or “[fa1] and [fb2] and [fc2]” or “[fa1] and [fb2] and [fc1]”



**Figure 13:** An example *speciesFeatureType* to illustrate the usage of the **ListOfSpeciesFeatures** class and the **SpeciesFeature** class

The SBML code can be as follows and the *species* “**sp\_A1**”, “**sp\_A2**” and “**sp\_A3**” contain the tree *listOfSpeciesFeatures* above respectively.

```
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="st_B" multi:isBindingSite="true">
    <multi:listOfSpeciesFeatureTypes>
      <multi:speciesFeatureType multi:id="fb" multi:occur="1">
        <multi:possibleSpeciesFeatureValue multi:id="fb1" />
        <multi:possibleSpeciesFeatureValue multi:id="fb2" />
      </multi:speciesFeatureType>
    </multi:listOfSpeciesFeatureTypes>
  </multi:speciesType>
  <multi:speciesType multi:id="st_C" multi:isBindingSite="true">
    <multi:listOfSpeciesFeatureTypes>
```



```

    <multi:speciesFeatureType multi:id="fC" multi:occur="1">
      <multi:possibleSpeciesFeatureValue multi:id="fc1" />
      <multi:possibleSpeciesFeatureValue multi:id="fc2" />
    </multi:speciesFeatureType>
  </multi:listOfSpeciesFeatureTypes>
</multi:speciesType>
<multi:speciesType multi:id="st_A" multi:isBindingSite="false">
  <multi:listOfSpeciesFeatureTypes>
    <multi:speciesFeatureType multi:id="fA" multi:occur="1">
      <multi:possibleSpeciesFeatureValue multi:id="fa1" />
      <multi:possibleSpeciesFeatureValue multi:id="fa2" />
    </multi:speciesFeatureType>
  </multi:listOfSpeciesFeatureTypes>
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="sti_B" multi:speciesType="st_B" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="sti_C" multi:speciesType="st_C" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
</multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="sp_A1" multi:speciesType="st_A" ...>
    <!-- [fa1] and [fb1] and [fc1] -->
    <multi:listOfSpeciesFeatures multi:relation="and" >
      <multi:speciesFeature multi:speciesFeatureType="fA" multi:occur="1" multi:component="st_A">
        <multi:speciesFeatureValue multi:value="fa1" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="fB" multi:occur="1" multi:component="sti_B">
        <multi:speciesFeatureValue multi:value="fb1" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="fC" multi:occur="1" multi:component="sti_C">
        <multi:speciesFeatureValue multi:value="fc1" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
    <multi:listOfOutwardBindingSites>
      ...
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_A2" multi:speciesType="st_A" ...>
    <!-- {[fa1] and [fb1]} and [fc1]} or {[fa2] and [fb2] and [fc2]} -->
    <multi:listOfSpeciesFeatures multi:relation="or" >
      <multi:subListOfSpeciesFeatures multi:relation="and" >
        <multi:speciesFeature multi:speciesFeatureType="fA" multi:occur="1" multi:component="st_A">
          <multi:speciesFeatureValue multi:value="fa1" />
        </multi:speciesFeature>
        <multi:speciesFeature multi:speciesFeatureType="fB" multi:occur="1" multi:component="sti_B">
          <multi:speciesFeatureValue multi:value="fb1" />
        </multi:speciesFeature>
        <multi:speciesFeature multi:speciesFeatureType="fC" multi:occur="1" multi:component="sti_C">
          <multi:speciesFeatureValue multi:value="fc1" />
        </multi:speciesFeature>
      </multi:subListOfSpeciesFeatures>
      <multi:subListOfSpeciesFeatures multi:relation="and" >
        <multi:speciesFeature multi:speciesFeatureType="fA" multi:occur="1" multi:component="st_A">
          <multi:speciesFeatureValue multi:value="fa2" />
        </multi:speciesFeature>
        <multi:speciesFeature multi:speciesFeatureType="fB" multi:occur="1" multi:component="sti_B">
          <multi:speciesFeatureValue multi:value="fb2" />
        </multi:speciesFeature>
        <multi:speciesFeature multi:speciesFeatureType="fC" multi:occur="1" multi:component="sti_C">
          <multi:speciesFeatureValue multi:value="fc2" />
        </multi:speciesFeature>
      </multi:subListOfSpeciesFeatures>
    </multi:listOfSpeciesFeatures>
    <multi:listOfOutwardBindingSites>
      ...
    </multi:listOfOutwardBindingSites>
  </species>

```

```

</species>
<species id="sp_A3" multi:speciesType="st_A" ...>
  <!-- {[fa1] and [fb1] and [fc2]} or {[fa1] and [fb2] and [fc2]} or {[fa1] and [fb2] and [fc1]} -->
  <multi:listOfSpeciesFeatures multi:relation="and" >
    <multi:speciesFeature multi:speciesFeatureType="fA" multi:occur="1" multi:component="st_A">
      <multi:speciesFeatureValue multi:value="fa1" />
    </multi:speciesFeature>
    <multi:subListOfSpeciesFeatures multi:relation="not" >
      <multi:speciesFeature multi:speciesFeatureType="fB" multi:occur="1" multi:component="sti_B">
        <multi:speciesFeatureValue multi:value="fb1" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="fC" multi:occur="1" multi:component="sti_C">
        <multi:speciesFeatureValue multi:value="fc1" />
      </multi:speciesFeature>
    </multi:subListOfSpeciesFeatures>
  </multi:listOfSpeciesFeatures>
  <multi:listOfOutwardBindingSites>
    ...
  </multi:listOfOutwardBindingSites>
</species>
</listOfSpecies>

```

### 3.19 “Fully defined” species and the mapping to “pattern” species

An extended **Species** object functions as a “template” or a “pattern” which allows multiple pools of entities to map to it. A *species* is “fully defined” if there is only one pool mapping to it. A “fully defined” *species* can be considered the same as a SBML core *species*, and can be initialized with the `initialAmount` attribute, or the `initialConcentration` attribute, or via an **InitialAssignment** object. In the *Multi* package, a *species* is “fully defined” if the following conditions are fulfilled.

- All *outwardBindingSites* must be free (`bindingStatus`=“unbound”), since “bound” sites imply that there is a non-specified binding partner.
- Each *speciesFeature* occurrence can only have one *speciesFeatureValue*, and every occurrences of every *speciesFeatureTypes* of every **components** of the referenced *speciesType* must be referenced by exactly one *speciesFeature* occurrence.
- If applicable, only “and” values are allowed for the `relation` attributes of the **ListOfSpeciesFeatures** objects.
- Only one single **SpeciesFeatureValue** object is allowed for any *speciesFeature*.
- The referenced *compartment* can not be a *compartment type*, which means the value of the `isType` attribute of the referenced *compartment* can only be “false”.

The mapping from a “fully defined” *species* to a “pattern” *species* is implicit and can be inferred from the structure of the *species*. For example, a *speciesType* “stA” has one *speciesFeatureType* with two *possibleSpeciesFeatureValues* “v1” and “v2”. A *species* “spA1” references “stA” and has a *speciesFeature* with value of “v1”. Another *species* “spA” also references “stA” and has no *speciesFeature* explicitly defined. Thus, the *species* “spA1” is a “fully defined” *species* and can map to the “pattern” *species* “spA” because the *species* “spA” has an implicit *speciesFeature* which can take either value “v1” or value “v2” (see the note in [Section 3.16.3 on page 21](#)).

#### Note:

Theoretically, using “not” and “or” can also result in a “fully defined” *species*. For example, a *speciesType* has two feature types “A” (“a1” and “a2” as possible values) and “B” (“b1” and “b2” as possible values). A “fully defined” *species* referencing the *speciesType* can be defined to have a feature of “[a1 and b1]”. Equivalently, the *species* can also be defined to have “[not ([a1 and b2] or [a2 and b2] or [a2 and b1])]”. In the *Multi* package, the main reason to define “fully defined” *species* is to initialize *species* in a model. Therefore, the definition for “fully defined” *species* simply disallows “not” and “or” to make it easier for a modeler to define “fully defined” *species*.

## 3.20 Reaction

The **Reaction** class is extended from SBML Level 3 Core to use the *Multi Species* objects to construct *reactions*. The extended **Reaction** class can not only define the relations among pools (SBML core *species*), but also the relations among patterns (*Multi* extended *species*). Several related classes including **SimpleSpeciesReference** and **SpeciesReference** are also extended to handle some issues specific to the *Multi* package.

The extension of the **Reaction** class is defined in Figure 14. An extended *reaction* has a new optional attribute `isIntraSpeciesReaction`.

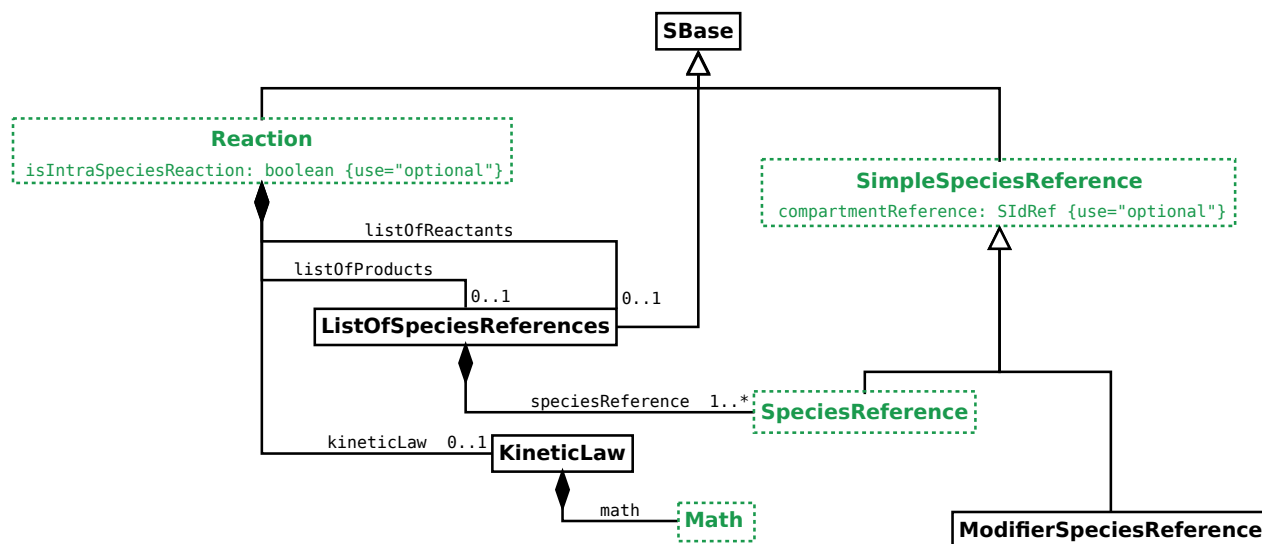


Figure 14: The extensions of the **Reaction** class and the related classes including **SimpleSpeciesReference**, **SpeciesReference** and **Math**

### 3.20.1 The `isIntraSpeciesReaction` attribute

The extended **Reaction** class has an optional `isIntraSpeciesReaction` attribute, of type `boolean`, to indicate whether the *reaction* happens within a *species* (a binding reaction that links two *bindingSites* within one complex would be an example).

A particular *reaction* may happen within a *species* and this attribute should be defined if the following conditions are fulfilled.

- The *reaction* is either an association *reaction* or a dissociation *reaction*.
- If it is an association *reaction*, each of the two reactant *species* has at least one *outwardBindingSite* free (“unbound”).
- If it is a dissociation *reaction*, each of the two product *species* has at least one *outwardBindingSite* free (“unbound”).

**Note:**

Technically, transformations are also reactions happening with one *species*, but they do not have the ambiguity in association and dissociation reactions. Therefore, the `isIntraSpeciesReaction` attribute is not required when defining transformation reactions.

## 3.21 SimpleSpeciesReference

The **SimpleSpeciesReference** class is extended with a new optional attribute **compartmentReference**, of type **SIdRef**, to reference a *compartmentReference*. The **compartmentReference** attribute can serve to indicate which sub-compartment where an object of a class (**SpeciesReference** or **ModifierSpeciesReference**) inheriting **SimpleSpeciesReference** is located.

This example illustrates the use of the **compartmentReference** attribute. A *model* has a *compartment* type “c” and a composite *compartment* type “cc” with two *compartmentReferences* “cr1” and “cr2” both referencing the *compartment* “c”. Both *species* “spA” and spM reference the *compartment* type “c”. A *reaction* happens between two “spA” *species* from two *compartments* respectively and results in a cross-compartment product. One condition for this *reaction* is that two “spM” *species* work as modifiers in the two “c” *compartments* respectively. The situation described here could correspond to interactions among *species* located on two adjacent membranes. Without the **compartmentReference** attribute in the **SimpleSpeciesReference** class, it is impossible to distinguish the two “spA” *species* as well as the two “spM” *species*. The SBML code can be as follows:

```
<listOfCompartments>
  <compartment id="c" constant="true" multi:isType="true" />
  <compartment id="cc" constant="true" multi:isType="true">
    <multi:listOfCompartmentReferences>
      <multi:compartmentReference multi:id="cr1" multi:compartment="c" />
      <multi:compartmentReference multi:id="cr2" multi:compartment="c" />
    </multi:listOfCompartmentReferences>
  </compartment>
</listOfCompartments>
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stA" multi:compartment="c" multi:isBindingSite="true" />
  <multi:speciesType multi:id="stM" multi:compartment="c" multi:isBindingSite="false" />
  <multi:speciesType multi:id="stAA" multi:compartment="cc" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="stiA1" multi:speciesType="stA"
        multi:compartmentReference="cr1" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="stiA2" multi:speciesType="stA"
        multi:compartmentReference="cr2" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="stiA1" multi:bindingSite2="stiA2" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="spA" multi:speciesType="stA" compartment="c" ... />
  <species id="spM" multi:speciesType="stM" compartment="c" ... />
  <species id="spAA" multi:speciesType="stAA" compartment="cc" ... />
</listOfSpecies>
<reaction id="reaction" ...>
  <listOfReactants>
    <speciesReference id="r1" species="spA" multi:compartmentReference="cr1" ... />
    <speciesReference id="r2" species="spA" multi:compartmentReference="cr2" ... />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="spAA" ... />
  </listOfProducts>
  <listOfModifiers>
    <modifierSpeciesReference id="m1" species="spM" multi:compartmentReference="cr1" />
    <modifierSpeciesReference id="m2" species="spM" multi:compartmentReference="cr2" />
  </listOfModifiers>
  ...
</reaction>
```

## 3.22 SpeciesReference

The **SpeciesReference** class is extended from SBML Level 3 Core and can establish **component** mappings between the reactant *species* and the product *species* when the mappings can not be inferred from the **ids** of the **SpeciesTypeInstance** objects. The **SpeciesReference** class has an optional **ListOfSpeciesTypeComponentMapsInProduct** child, as defined in Figure 15. Only a *reaction product* can contain the **ListOfSpeciesTypeComponentMapsInProduct** child and it is not necessary to store the mappings again in the *reactants*.

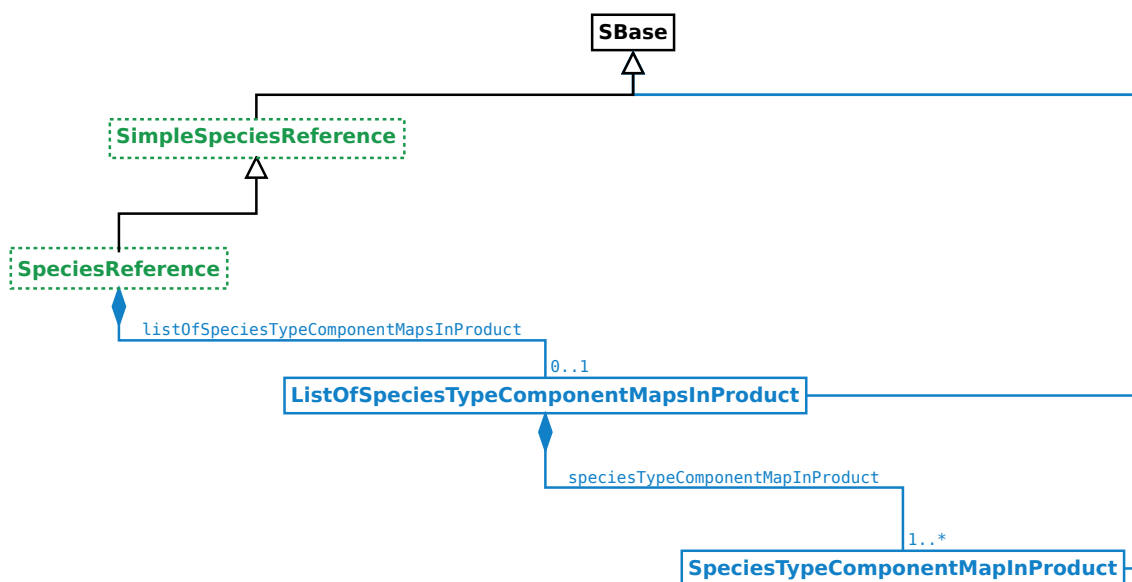


Figure 15: The extension of the **SpeciesReference** class

### 3.22.1 ListOfSpeciesTypeComponentMapsInProduct

The **ListOfSpeciesTypeComponentMapsInProduct** class is defined in Figure 15. If present, it must have one or more **SpeciesTypeComponentMapInProduct** children.

## 3.23 SpeciesTypeComponentMapInProduct

The **SpeciesTypeComponentMapInProduct** class is defined in Figure 16 on the following page. A *speciesTypeComponentMapInProduct* defines the mapping between a **component** in a reactant and a **component** in a product. The identifications of a **component** and the *speciesReference* should be sufficient to identify the **component** in the context of a *reaction*. The attributes **reactant** and **reactantComponent** can identify the **component** in a *reactant*, and the **productComponent** attribute and the product storing the mapping information can identify the component in a *product*.

### 3.23.1 The reactant attribute

The required **reactant** attribute, of type **SIIdRef**, on the **SpeciesTypeComponentMapInProduct** class is used to reference a *reactant* in the *reaction*. It can either reference the **id** of a *reactant speciesReference*, or the **id** of a *reactant species* if the *reactant* can be identified by the *species id* without any ambiguity.

### 3.23.2 The reactantComponent attribute

The required **reactantComponent** attribute, of type **SIIdRef**, on the **SpeciesTypeComponentMapInProduct** class is used to reference a **component** of a reactant *species*.

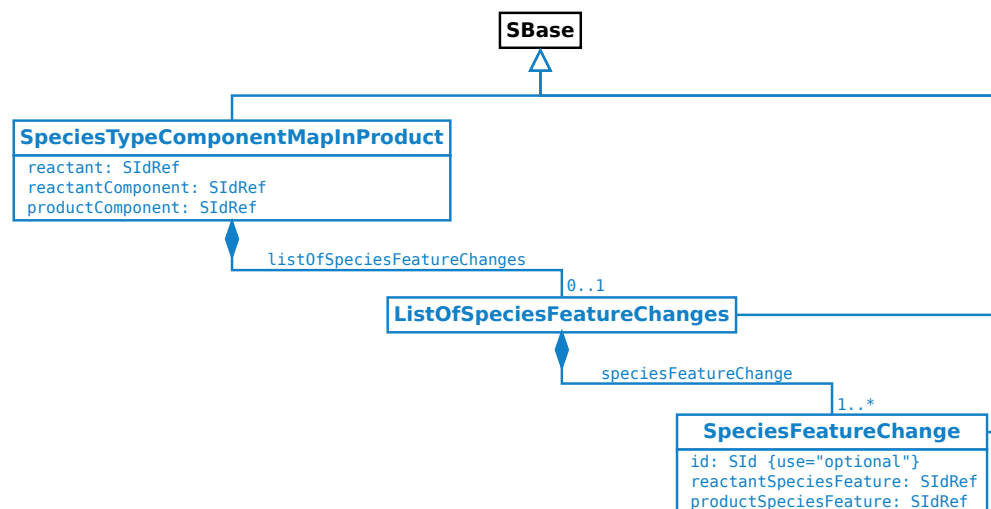


Figure 16: The definitions of the **SpeciesTypeComponentMapInProduct** and **SpeciesFeatureChange** classes

### 3.23.3 The productComponent attribute

The required **productComponent** attribute, of type **SIdRef**, on the **SpeciesTypeComponentMapInProduct** class is used to reference a **component** of a product **species**.

### 3.23.4 ListOfSpeciesFeatureChanges

The **SpeciesTypeComponentMapInProduct** class also has an optional **ListOfSpeciesFeatureChanges** child to explicitly define changes of **speciesFeatures** in a **reaction**. If present, it must have one or more **SpeciesFeatureChange** children.

### 3.23.5 Example

Figure 17 on the next page illustrates the use of the **SpeciesTypeComponentMapInProduct** class. The **speciesType** “**stX**” has two “**b**” **bindingSites** and one “**A**” **speciesFeatureType** that has two **possibleSpeciesFeatureValues** “**a1**” and “**a2**”. The **species** “**spY**” has two **speciesTypeInstances** “**stiX1**” and “**stiX2**” both of the **speciesType** “**stX**”. The **species** “**spX**” references the **speciesType** “**stX**”. The **reaction** is an association between one **bindingSite** “**b**” of “**stiX2**” in “**spY**” and one **bindingSite** “**b**” in “**spX**”.

The mappings are as follows and indicated by the colors in the diagram:

- “**stiX1**” of “**spY**” <==> “**stiX1**” of “**spZ**”.
- “**stiX2**” of “**spY**” <==> “**stiX2**” of “**spZ**”.
- “**spX**” <==> “**stiX3**” of “**spZ**”.

The SBML code can be as follows:

```

<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stB" multi:isBindingSite="true" />
  <multi:speciesType multi:id="stX" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="b" multi:speciesType="stB" multi:occur="2" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesFeatureTypes>
      <multi:speciesFeatureType multi:id="A">
        <multi:possibleSpeciesFeatureValue multi:id="a1" />
      </multi:speciesFeatureType>
    </multi:listOfSpeciesFeatureTypes>
  </multi:speciesType>
</multi:listOfSpeciesTypes>

```

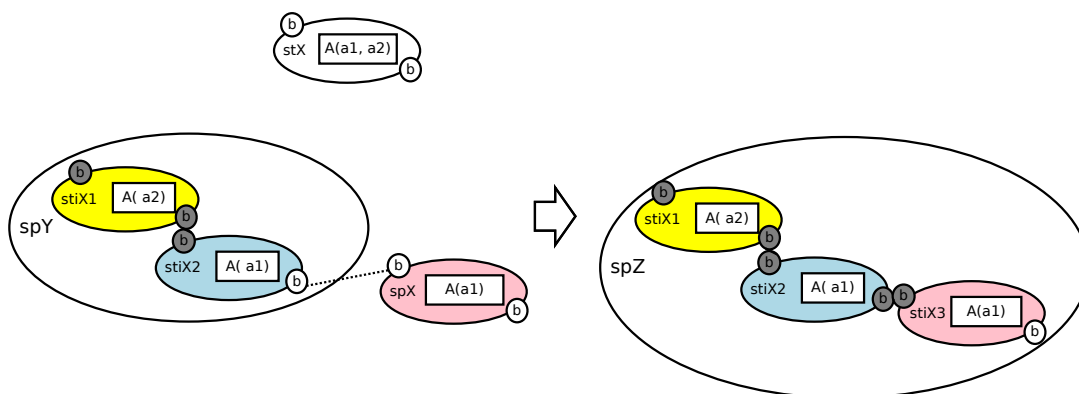


Figure 17: An example illustrating the usage of the [SpeciesTypeComponentMapInProduct](#) class (colors for mappings)

```

<multi:possibleSpeciesFeatureValue multi:id="a2" />
</multi:speciesFeatureType>
</multi:listOfSpeciesFeatureTypes>
<multi:listOfSpeciesTypeComponentIndexes>
  <multi:speciesTypeComponentIndex multi:id="b1" multi:component="b"
    multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="b2" multi:component="b"
    multi:occur="1"/>
</multi:listOfSpeciesTypeComponentIndexes>
</multi:speciesType>
<multi:speciesType multi:id="stY" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="stiX1" multi:speciesType="stX" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="stiX2" multi:speciesType="stX" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfSpeciesTypeComponentIndexes>
    <multi:speciesTypeComponentIndex multi:id="x1b1" multi:component="b"
      multi:identifyingParent="stiX1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x1b2" multi:component="b"
      multi:identifyingParent="stiX1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x2b1" multi:component="b"
      multi:identifyingParent="stiX2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x2b2" multi:component="b"
      multi:identifyingParent="stiX2" multi:occur="1" />
  </multi:listOfSpeciesTypeComponentIndexes>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="x1b2" multi:bindingSite2="x2b1" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
<multi:speciesType multi:id="stZ">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="stiX1" multi:speciesType="stX" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="stiX2" multi:speciesType="stX" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="stiX3" multi:speciesType="stX" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfSpeciesTypeComponentIndexes>
    <multi:speciesTypeComponentIndex multi:id="x1b1" multi:component="b"
      multi:identifyingParent="stiX1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x1b2" multi:component="b"
      multi:identifyingParent="stiX1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x2b1" multi:component="b"
      multi:identifyingParent="stiX2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x2b2" multi:component="b"
      multi:identifyingParent="stiX2" multi:occur="1" />
  </multi:listOfSpeciesTypeComponentIndexes>

```

```

    <multi:speciesTypeComponentIndex multi:id="x3b1" multi:component="b"
      multi:identifyingParent="stiX3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="x3b2" multi:component="b"
      multi:identifyingParent="stiX3" multi:occur="1" />
  </multi:listOfSpeciesTypeComponentIndexes>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="x1b2" multi:bindingSite2="x2b1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="x2b2" multi:bindingSite2="x3b1" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="spX" multi:speciesType="stX" ...>
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="b1" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="b2" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:speciesFeatureType="A">
        <multi:speciesFeatureValue multi:value="a1" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
  </species>
  <species id="spY" multi:speciesType="stY" ... >
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="x1b1" multi:bindingStatus="bound" />
      <multi:outwardBindingSite multi:component="x2b2" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:speciesFeatureType="A" multi:component="stiX1">
        <multi:speciesFeatureValue value="a2" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="A" multi:component="stiX2">
        <multi:speciesFeatureValue value="a1" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
  </species>
  <species id="spZ" multi:speciesType="st_z" ... >
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="x1b1" multi:bindingStatus="bound" />
      <multi:outwardBindingSite multi:component="x3b2" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:speciesFeatureType="A" multi:component="stiX1">
        <multi:speciesFeatureValue value="a2" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="A" multi:component="stiX2">
        <multi:speciesFeatureValue value="a1" />
      </multi:speciesFeature>
      <multi:speciesFeature multi:speciesFeatureType="A" multi:component="stiX3">
        <multi:speciesFeatureValue value="a1" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
  </species>
</listOfSpecies>
<listOfReactions>
  <reaction id="association" ...>
    <listOfReactants>
      <speciesReference id="reactantX" species="spX" />
      <speciesReference id="reactantY" species="spY" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference id="productZ" species="spZ">
        <multi:listOfSpeciesTypeComponentMapsInProduct>
          <multi:speciesTypeComponentMapInProduct multi:reactant="reactantY"
            multi:reactantComponent="stiX1" multi:productComponent="stiX1" />
        </multi:listOfSpeciesTypeComponentMapsInProduct>
      </speciesReference>
    </listOfProducts>
  </reaction>
</listOfReactions>

```



```

    <multi:speciesTypeComponentMapInProduct multi:reactant="reactantY"
      multi:reactantComponent="stiX2" multi:productComponent="stiX2" />
    <multi:speciesTypeComponentMapInProduct multi:reactant="reactantX"
      multi:reactantComponent="stX" multi:productComponent="stiX3" />
  </multi:listOfSpeciesTypeComponentMapsInProduct>
</speciesReference>
</listOfProducts>
...
</reaction>
</listOfReactions>

```

## 3.24 SpeciesFeatureChange

The **SpeciesFeatureChange** class is defined in Figure 16 on page 30 and provides a way to specify that some of or all instances of a *speciesFeatureType* change. This class should only be used when the **occur** of the referenced *speciesFeatureType* is larger than “1”. The parent **components** of the changed *speciesFeatures* are identified in the **SpeciesTypeComponentMapInProduct** object. **SpeciesFeatureChange** has one optional attribute **id** and two required attributes, **reactantSpeciesFeature** and **productSpeciesFeature**. The **occur** attributes of the changed *speciesFeatures* in *reactant* and *product* respectively must have the same value.

### 3.24.1 The id attribute

The optional **id** attribute, of type **SIId**, provides a way to identify the *speciesFeatureChange*.

### 3.24.2 The reactantSpeciesFeature attribute

The **reactantSpeciesFeature** attribute, of type **SIIdRef**, references a *speciesFeature* in the reactant **component** in a reaction mapping.

### 3.24.3 The productSpeciesFeature attribute

The **productSpeciesFeature** attribute, of type **SIIdRef**, references a *speciesFeature* in the product **component** in a reaction mapping.

### 3.24.4 Example

Here is an example to illustrate the use of the **SpeciesFeatureChange** class in a phosphorylation *reaction*. One among the five sites in a *species* is transformed from “unphosphorylated” to “phosphorylated” and the phosphorylation sites are defined as the referenced *speciesFeatureType* with **occur**=“5”. The SBML code can be as follows:

```

<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stX" ... >
    <multi:listOfSpeciesFeatureTypes>
      <multi:speciesFeatureType multi:id="phosphorylation" multi:occur="5">
        <multi:possibleSpeciesFeatureValue multi:id="phosphorylated" />
        <multi:possibleSpeciesFeatureValue multi:id="unphosphorylated" />
      </multi:speciesFeatureType>
    </multi:listOfSpeciesFeatureTypes>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="spX1" multi:speciesType="stX">
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:id="U"
        multi:speciesFeatureType="phosphorylation" multi:occur="1" >
        <multi:speciesFeatureValue multi:value="unphosphorylated" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
    ...
  </species>

```

```

<species id="spX2" multi:speciesType="stX">
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:id="P"
      multi:speciesFeatureType="phosphorylation" multi:occur="1" >
      <multi:speciesFeatureValue multi:value="phosphorylated" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
  ...
</species>
</listOfSpecies>
<listOfReactions>
  <reaction id="transformation" ... >
    <listOfReactants>
      <speciesReference id="reactant" species="spX1" ... />
    </listOfReactants>
    <listOfProducts>
      <speciesReference id="product" species="spX2" ...>
        <multi:listOfSpeciesTypeComponentMapsInProdcut>
          <multi:speciesTypeComponentMapInProduct multi:reactant="reactant"
            multi:reactantComponent="stX" multi:productComponent="stX">
            <multi:listOfSpeciesFeatureChanges>
              <multi:speciesFeatureChange multi:reactantFeature="P"
                multi:productFeature="U" />
            </multi:listOfSpeciesFeatureChanges>
          </multi:speciesTypeComponentMapInProduct>
        </multi:listOfSpeciesTypeComponentMapsInProdcut>
      </speciesReference>
    </listOfProducts>
  </reaction>
</listOfReactions>

```

### 3.25 The *outwardBindingSites* and *speciesFeatures* in “don’t care” state in a reaction *product*

An *outwardBindingSite* is in “don’t care” state if its *bindingStatus* is “either” or is not specified (also see [Section 3.16.2 on page 20](#)). A *speciesFeature* or an instance of a *speciesFeature* (the *occur* of its *speciesFeatureType* is larger than “1”) is in “don’t care” state if it has all the *possibleSpeciesFeatureValues* under its *speciesFeatureType*, or it is not specified in the *species* (also see [Section 3.16.3 on page 21](#)).

For a *species* as a *product* in a reaction, if it has “don’t care” *outwardBindingSites* or “don’t care” *speciesFeatures*, the interpretation of the “don’t care” is “don’t change”. In a *product*, a “don’t care” *outwardBindingSite* has the same *bindingStatus* as the mapped *outwardBindingSite* in the *reactant*, and a “don’t care” *speciesFeature* or instance of a *speciesFeature* has the same value as the mapped *speciesFeature* or the mapped *speciesFeature* instance in the *reactant*.

For the phosphorylation example in [Section 3.24.4 on the previous page](#), the reactant *species* has one “unphosphorylated” site and four “don’t care” sites, and the product *species* has one “phosphorylated” site and four don’t care sites. The “phosphorylation” reaction can apply to the following *reactions* of “fully defined” *species*.

- Reactant: a *species* with “0” phosphorylated site and “5” unphosphorylated sites.  
Product: a *species* with “1” phosphorylated site and “4” unphosphorylated sites.
- Reactant: a *species* with “1” phosphorylated site and “4” unphosphorylated sites.  
Product: a *species* with “2” phosphorylated sites and “3” unphosphorylated sites.
- Reactant: a *species* with “2” phosphorylated sites and “3” unphosphorylated sites.  
Product: a *species* with “3” phosphorylated sites and “2” unphosphorylated sites.
- Reactant: a *species* with “3” phosphorylated sites and “2” unphosphorylated sites.

Product: a *species* with “4” phosphorylated sites and “1” unphosphorylated site.

- Reactant: a *species* with “4” phosphorylated sites and “1” unphosphorylated site.

Product: a *species* with “5” phosphorylated sites and “0” unphosphorylated site.

## 3.26 Extended *ci* element in Math

The *Multi* package extends the *ci* element in **Math** in **Reaction** with optional attributes `speciesReference` and `representationType`.

### 3.26.1 The `speciesReference` attribute

The optional `speciesReference` attribute, of type `SIdRef`, can only be used when the content of the *ci* element is a *species* id, or when the content of the *ci* element is a *speciesFeature* id. The `speciesReference` attribute can identify which *species* is referenced in a *reaction*.

If the *ci* content references a *species*’ id, the id represent the concentration of the *species*.

If the *ci* content references a *speciesFeature*’s id, the id represent the count of the *speciesFeature* instances with the *speciesFeatureValue* (also see [Section 3.18.1 on page 23](#)).

The example in [Section 3.21 on page 28](#) can be further extended with a block of *kineticLaw* in the *reaction* to illustrate the use of the `speciesReference` attribute with a *species*’ id.

```
<reaction id="reaction" ... >
  ...
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> k </ci>
        <ci multi:speciesReference="r1"> spA </ci>
        <ci multi:speciesReference="m1"> spM </ci>
        <ci multi:speciesReference="r2"> spA </ci>
        <ci multi:speciesReference="m2"> spM </ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="k" value="0.1" ... />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>
```

Two “spA” *species* and two “spM” *species* are distinguished by the “r1” and “r2” *speciesReferences* respectively.

Here is another example to show the use of the `speciesReference` attribute for a *possibleSpeciesFeatureValue*. This example is a simplified adaptation of published models [[Malleshaiah et al. \(2010\)](#), [Barik et al. \(2010\)](#)]. A *species* “Y” has 10 phosphorylation sites. It can be phosphorylated by another *species* “M” one site one time and the phosphorylation rate depends on the number of sites already phosphorylated in *species* “Y”. The SBML code can be as follows:

```
<mult:listOfSpeciesTypes>
  <multi:speciesType multi:id="Y" multi:occur="1" >
    <multi:listOfSpeciesFeatureType multi:id="phosphorylation" multi:occur="10" >
      <multi:possibleSpeciesFeatureValue multi:id="P" multi:name="phosphorylated" />
      <multi:possibleSpeciesFeatureValue multi:id="U" multi:name="unphosphorylated" />
    </multi:listOfSpeciesFeatureType>
  </multi:speciesType>
</mult:listOfSpeciesTypes>
<listOfSpecies>
  <species id="Yu" multi:speciesType="Y" ...>
```

```

<multi:listOfSpeciesFeatures>
  <multi:speciesFeature mult:id="fU" multi:occur="1">
    <multi:speciesFeatureValue value="U" />
  </multi:speciesFeature>
</multi:listOfSpeciesFeatures>
</species>
<species id="Yp" multi:speciesType="Y" ...>
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature mult:id="fP" multi:occur="1">
      <multi:speciesFeatureValue value="P" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="M" ... />
</listOfSpecies>
<listOfReactions>
  <reaction id="reaction" .>
    <listOfReactants>
      <speciesReference id="r" species="Yu" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference id="p" species="Yp" >
        <multi:listOfSpeciesTypeComponentMapsInProduct>
          <multi:speciesTypeComponentMapInProduct reactant="r" reactantComponent="stY"
            productReactant="stY" >
            <multi:listOfSpeciesFeatureChanges>
              <multi:speciesFeatureChange reactantSpeciesFeature="fU"
                productSpeciesFeature="fP" />
            </multi:listOfSpeciesFeatureChanges>
          </multi:speciesTypeComponentMapInProduct>
        </multi:listOfSpeciesTypeComponentMapsInProduct>
      </speciesReference>
    </listOfProducts>
    <listOfModifierSpeciesReferences>
      <modifierSpeciesReference species="M" />
    </listOfModifierSpeciesReferences>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci> k </ci>
          <ci> Yu </ci>
          <ci> M </ci>
          <ci multi:speciesReference="r"> P </ci>
        </apply>
      </math>
      <listOfLocalParameters>
        <localParameter id="k" ... />
      </listOfLocalParameters>
    </kineticLaw>
  </reaction>
</listOfReactions>

```

Any “fully defined” *species* referencing “Y” with at least one unphosphorylated site maps to the *species* “Yu”. Any “fully defined” *species* referencing “Y” with at least one phosphorylated site maps to the *species* “Yp”. The *species-FeatureChange* references *speciesFeatures* “fu” and “fP” and the value of “1” for both *occur* attributes of “fu” and “fP” indicates that one site is phosphorylated in the *reaction*. The `<ci multi:speciesReference="r"> P </ci>` depends on the “fully defined” *species* mapping to the *species* “Yu” which is referenced by the *speciesReference* “r”. If the “fully defined” *species* has 1 site phosphorylated, the *ci* is “1” in the math, similarly, *ci* is 2 for 2 phosphorylated sites, ..., *ci* is 9 for 9 phosphorylated sites.

### 3.26.2 The representationType attribute

The optional `representationType` attribute, of type `RepresentationType`, can only be used when the content of the *ci* element is a *species* id or a *possibleSpeciesFeatureValue*'s id. The `representationType` and `speciesReference` attributes can both be used for the same *ci* element at the same time.

The `representationType` attribute can only have the value of “sum” when the content of the *ci* is *species*. The interpretation of such a *ci* element is the total concentration or amount of all “fully defined” *species* (see [Section 3.19 on page 26](#)) mapping to the referenced pattern *species*.

The `representationType` attribute can have the value of “numericValue” when the content of the *ci* is *possibleSpeciesFeatureValue* and the `speciesReference` attribute must be defined. The interpretation of such a *ci* is the same as a *ci* element having a *parameter* which the *possibleSpeciesFeatureValue* links via its `numericValue` attribute.

The following example demonstrates the use of this attribute for “sum” of *species* concentrations.

$$k1 \cdot Si / (k2 + \text{SUM}(Si))$$

In this example, the *reactant* “Si” is a pattern *species* which may have multiple “fully defined” *species* mapping to it, for example *species* “S1”, “S2”, ..., “Sn”. “SUM(Si)” is a function to calculate the total concentration of all “fully defined” *species* mapping to “Si”. The *product* can be another pattern *species* “Pi”. The SBML code for the math expression can be as follows:

```
<reaction id="r">
  <listOfReactants>
    <speciesReference species="Si" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="Pi" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <divide>
          <apply>
            <times />
            <ci>Si</ci>
            <ci>k1</ci>
          </apply>
          <apply>
            <plus />
            <ci>k2</ci>
            <ci multi:patternSpeciesConcentration="sum">Si</ci>
          </apply>
        </divide>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="k1" ... />
      <localParameter id="k2" .... />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>
```

The math expressions for the individual *species* in the example can be:

```
For species S1:  k1*S1/(k2 + (S1 + S2 + ... + Sn))
For species S2:  k1*S2/(k2 + (S1 + S2 + ... + Sn))
...
For species Sn:  k1*Sn/(k2 + (S1 + S2 + ... + Sn))
```

## 4 Examples

This section contains examples employing the *Multi* package for SBML Level 3.

### 4.1 Example: **Compartment**, **SpeciesType** and **Species**

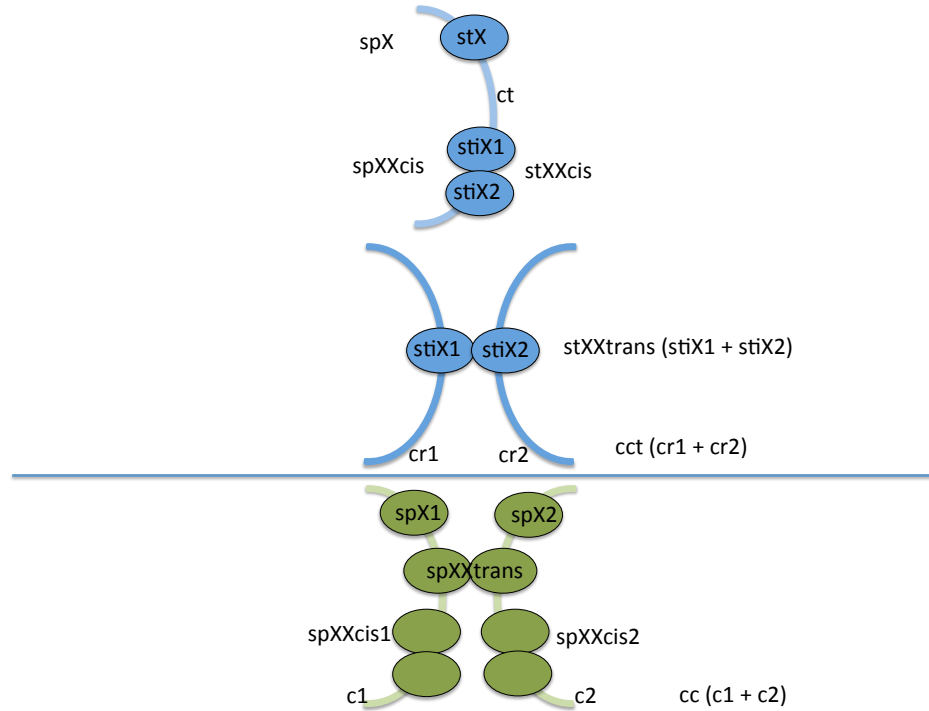


Figure 18: Diagram for an example of **Compartment**, **SpeciesType** and **Species**

Figure 18 shows an example illustrating the usages of and relations among the **Compartment**, **SpeciesType** and **Species** classes.

“ct” is a *compartment* type. “cct” is a composite *compartment* type with two *compartmentReferences* “cr1” and “cr2” both referencing “ct”. “c1” is a “not-a-type” *compartment* and references “ct”. Similarly, “c2” is also a “not-a-type” *compartment* and references “ct”. “cc” is a composite “not-a-type” *compartment* composed of “c1” and “c2”.

“stX” is a *speciesType* on the “ct” *compartment*. “stXXcis” is a *speciesType* on the “ct” *compartment*, and has two *speciesTypeInstances* “stiX1” and “stiX2” both of that reference the “stX” *speciesType*. “stXXtrans” is a *speciesType* on the “cct” *compartment* with two *speciesTypeInstances* “stiX1” and “stiX2” sitting in different sub-compartments.

“spX” is a *species* referencing *speciesType* “stX”. “spXXcis” is a *species* referencing “stXXcis”. “spX1” is a *species* referencing “stX” and sitting in the “c1” *compartment*. “spX2” is a *species* also referencing “stX”, but sitting in “c2”. “spXXtrans” is a *species* referencing “stXXtrans”. “spXXcis1” is a *species* referencing “stXXtrans” and sitting in “c1”. “spXXcis2” is a *species* referencing “stXXtrans” and sitting in “c2”.

“spX1”, “spX2”, “spXXtrans”, “spXXcis1” and “spXXcis2” are “fully defined” *species* (see Section 3.19 on page 26).

The SBML code can be as follows:

```
<listOfCompartments>
  <compartment id="ct" multi:isType="true" />
  <compartment id="cct" multi:isType="true">
    <multi:listOfCompartmentReferences>
      <multi:compartmentReference multi:id="cr1" multi:compartment="ct" />
      <multi:compartmentReference multi:id="cr2" multi:compartment="ct" />
    </multi:listOfCompartmentReferences>
  </compartment>
  <compartment id="c1" multi:isType="false" multi:compartmentType="ct" />
  <compartment id="c2" multi:isType="false" multi:compartmentType="ct" />
  <compartment id="cc" multi:isType="false" multi:compartmentType="cct">
    <multi:listOfCompartmentReferences>
      <multi:compartmentReference multi:compartment="c1" />
      <multi:compartmentReference multi:compartment="c2" />
    </multi:listOfCompartmentReferences>
  </compartment>
</listOfCompartments>
<multi:listOfSpeciesTypes>
  <multi:speciesType multi:id="stX" multi:compartment="ct" multi:isBindingSite="true" />
  <multi:speciesType multi:id="stXXcis" multi:compartment="ct" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="stiX1" multi:speciesType="stX" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="stiX2" multi:speciesType="stX" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="stiX1" multi:bindingSite2="stiX2" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
  <multi:speciesType multi:id="stXXtrans" multi:compartment="cct" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:speciesTypeInstance multi:id="stiX1" multi:speciesType="stX"
        multi:compartmentReference="cr1" multi:occur="1" />
      <multi:speciesTypeInstance multi:id="stiX2" multi:speciesType="stX"
        multi:compartmentReference="cr2" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="stiX1" multi:bindingSite2="stiX2" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="spX" multi:speciesType="stX" compartment="ct" />
  <species id="spXXcis" multi:speciesType="stXXcis" compartment="ct" />
  <species id="spX1" multi:speciesType="stX" compartment="c1" /> <!-- Fully defined -->
  <species id="spX2" multi:speciesType="stX" compartment="c2" /> <!-- Fully defined -->
  <species id="spXXtrans" multi:speciesType="stXXtrans" compartment="cc" /> <!-- Fully defined -->
  <species id="spXXcis1" multi:speciesType="stXXcis" compartment="c1" /> <!-- Fully defined -->
  <species id="spXXcis2" multi:speciesType="stXXcis" compartment="c2" /> <!-- Fully defined -->
</listOfSpecies>
```

## 4.2 *Simmune* example: the Ecad model

The *Simmune* toolset (<http://go.usa.gov/QeH>) has some example models including the published *Ecad model* [Angermann et al. (2012)]. The Ecad model describes the interactions between E-cadherin receptors that can associate either with other E-cadherin receptors within the same membrane (in “cis”) or with E-cadherin receptors on adjacent membranes (in “trans”). This model is transformed into the SBML Level 3 format with use of the *Multi* package.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">
  <model name="E-cadherin_mediated_adhesion">

    <!-- Definitions -->
    <listOfUnitDefinitions>
      <unitDefinition id="litre_per_mole_per_sec">
        <listOfUnits>
          <unit kind="litre" exponent="1" scale="0" multiplier="1" />
          <unit kind="mole" exponent="-1" scale="0" multiplier="1" />
          <unit kind="second" exponent="-1" scale="0" multiplier="1" />
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="micron_square_per_sec">
        <listOfUnits>
          <unit kind="metre" exponent="2" scale="-6" multiplier="1" />
          <unit kind="second" exponent="-1" scale="0" multiplier="1" />
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="micrometre_per_sec">
        <listOfUnits>
          <unit kind="metre" exponent="1" scale="-6" multiplier="1" />
          <unit kind="second" exponent="-1" scale="0" multiplier="1" />
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="per_sec">
        <listOfUnits>
          <unit kind="second" exponent="-1" scale="0" multiplier="1" />
        </listOfUnits>
      </unitDefinition>
    </listOfUnitDefinitions>

    <!-- Compartments -->
    <listOfCompartments>
      <compartment id="membrane" constant="true" multi:isType="true" />
      <compartment id="inter_membrane" constant="true" multi:isType="true">
        <multi:listOfCompartmentReferences>
          <multi:compartmentReference multi:id="m1" multi:compartment="membrane" />
          <multi:compartmentReference multi:id="m2" multi:compartment="membrane" />
        </multi:listOfCompartmentReferences>
      </compartment>
    </listOfCompartments>

    <!-- SpeciesTypes -->
    <multi:listOfSpeciesTypes>

      <!-- Ecad with cis-binding site and trans-binding site: -->
      <multi:speciesType multi:id="st_Cis_Interface" multi:isBindingSite="true" />
      <multi:speciesType multi:id="st_Trans_Interface" multi:isBindingSite="true" />
      <multi:speciesType multi:id="st_Ecad" multi:compartment="membrane"
        multi:isBindingSite="false">
        <multi:listOfSpeciesTypeInstances>
          <multi:speciesTypeInstance multi:id="cis" multi:speciesType="st_Cis_Interface"
            multi:occur="1" />
          <multi:speciesTypeInstance multi:id="trans" multi:speciesType="st_Trans_Interface"
            multi:occur="1" />
        </multi:listOfSpeciesTypeInstances>
      </multi:speciesType>

      <!-- cis dimer: -->
      <multi:speciesType multi:id="st_Ecad_cis_dimer" multi:compartment="membrane"
        multi:isBindingSite="false">
        <multi:listOfSpeciesTypeInstances>
          <multi:speciesTypeInstance multi:id="Ecad1" multi:speciesType="st_Ecad"
            multi:occur="1" />
          <multi:speciesTypeInstance multi:id="Ecad_2" multi:speciesType="st_Ecad"

```



```

        multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
        <multi:speciesTypeComponentIndex multi:id="Ecad1cis"
            multi:component="cis" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2cis"
            multi:component="cis" multi:identifyingParent="Ecad2" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad1trans"
            multi:component="trans" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2trans"
            multi:component="trans" multi:identifyingParent="Ecad2" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
    <multi:listOfInSpeciesTypeBonds>
        <multi:inSpeciesTypeBond multi:bindingSite1="Ecad1cis"
            multi:bindingSite2="Ecad2cis" />
    </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- trans dimer: -->
<multi:speciesType multi:id="st_Ecad_trans_dimer" multi:compartment="inter_membrane"
    multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
        <multi:speciesTypeInstance multi:id="Ecad1" multi:speciesType="st_Ecad"
            multi:compartmentReference="m1" multi:occur="1" />
        <multi:speciesTypeInstance multi:id="Ecad2" multi:speciesType="st_Ecad"
            multi:compartmentReference="m2" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
        <multi:speciesTypeComponentIndex multi:id="Ecad1trans"
            multi:component="trans" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2trans"
            multi:component="trans" multi:identifyingParent="Ecad2" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad1cis"
            multi:component="cis" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2cis"
            multi:component="cis" multi:identifyingParent="Ecad2" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
    <multi:listOfInSpeciesTypeBonds>
        <multi:inSpeciesTypeBond multi:bindingSite1="Ecad1trans"
            multi:bindingSite2="Ecad2trans" />
    </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- trimer: -->
<multi:speciesType multi:id="st_Ecad_trimer" multi:compartment="inter_membrane"
    multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
        <multi:speciesTypeInstance multi:id="Ecad1" multi:speciesType="st_Ecad"
            multi:compartmentReference="m1" multi:occur="1" />
        <multi:speciesTypeInstance multi:id="Ecad2" multi:speciesType="st_Ecad"
            multi:compartmentReference="m1" multi:occur="1" />
        <multi:speciesTypeInstance multi:id="Ecad3" multi:speciesType="st_Ecad"
            multi:compartmentReference="m2" multi:occur="1" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
        <multi:speciesTypeComponentIndex multi:id="Ecad1cis"
            multi:component="cis" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad1trans"
            multi:component="trans" multi:identifyingParent="Ecad1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2cis"
            multi:component="cis" multi:identifyingParent="Ecad2" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad2trans"
            multi:component="trans" multi:identifyingParent="Ecad2" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad3cis"
            multi:component="cis" multi:identifyingParent="Ecad3" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="Ecad3trans"

```

```

        multi:component="trans" multi:identifyingParent="Ecad3" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
    <multi:listOfInSpeciesTypeBonds>
        <multi:inSpeciesTypeBond multi:bindingSite1="Ecad1cis"
            multi:bindingSite2="Ecad2cis" />
        <multi:inSpeciesTypeBond multi:bindingSite1="Ecad1trans"
            multi:bindingSite2="Ecad3trans" />
    </multi:listOfInSpeciesTypeBonds>
    </multi:speciesType>
</multi:listOfSpeciesTypes>

<!-- Species -->
<listOfSpecies>

    <!-- free Ecad -->
    <species id="sp_Ecad_unbound" name="Ecad_unbound" compartment="membrane"
        hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
        multi:speciesType="st_Ecad">
        <multi:listOfOutwardBindingSites>
            <multi:outwardBindingSite multi:component="cis"
                multi:bindingStatus="unbound" />
            <multi:outwardBindingSite multi:component="trans"
                multi:bindingStatus="unbound" />
        </multi:listOfOutwardBindingSites>
    </species>

    <!-- Pattern species: Ecad trans unbd -->
    <species id="sp_Ecad_trans_unbd" name="Ecad_trans_unbd" compartment="membrane"
        hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
        multi:speciesType="st_Ecad">
        <multi:listOfOutwardBindings>
            <multi:outwardBindingSite multi:component="trans"
                multi:bindingStatus="unbound" />
        </multi:listOfOutwardBindings>
    </species>

    <!-- Pattern species: Ecad trans bnd -->
    <species id="sp_Ecad_trans_bnd" name="Ecad_trans_bnd" compartment="membrane"
        hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
        multi:speciesType="st_Ecad">
        <multi:listOfOutwardBindings>
            <multi:outwardBindingSite multi:component="trans"
                multi:bindingStatus="bound" />
        </multi:listOfOutwardBindings>
    </species>

    <!-- Pattern species: Ecad all -->
    <species id="sp_Ecad_all" name="Ecad_all" compartment="membrane"
        hasOnlySubstanceUnits="false"
        boundaryCondition="false" constant="false" multi:speciesType="st_Ecad" />

    <!-- Pattern species: Ecad cis unbd -->
    <species id="sp_Ecad_cis_unbd" name="Ecad_cis_unbd" compartment="membrane"
        hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
        multi:speciesType="st_Ecad">
        <multi:listOfOutwardBindings>
            <multi:outwardBindingSite multi:component="cis"
                multi:bindingStatus="unbound" />
        </multi:listOfOutwardBindings>
    </species>

    <!-- Pattern species: Ecad cis unbd, trans bnd -->
    <species id="sp_Ecad_6" name="Ecad_6" compartment="membrane" hasOnlySubstanceUnits="false"
        boundaryCondition="false" constant="false" multi:speciesType="st_Ecad">
        <multi:listOfOutwardBindings>
            <multi:outwardBindingSite multi:component="cis"

```

```

        multi:bindingStatus="unbound" />
        <multi:outwardBindingSite multi:component="trans"
        multi:bindingStatus="bound" />
    </multi:listOfOutwardBindings>
</species>

<!-- Pattern species: Ecad cis bnd, trans unbnd -->
<species id="sp_Ecad_7" name="Ecad_7" compartment="membrane" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false" multi:speciesType="st_Ecad">
    <multi:listOfOutwardBindings>
        <multi:outwardBindingSite multi:component="cis"
        multi:bindingStatus="bound" />
        <multi:outwardBindingSite multi:component="trans"
        multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindings>
</species>

<!-- Pattern species: Ecad cis dimer -->
<species id="sp_Ecad_cis_dimer" name="Ecad_cis_dimer" compartment="membrane"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
    multi:speciesType="st_Ecad_cis_dimer" />

<!-- Pattern species: Ecad cis dimer: all trans bnd -->
<species id="sp_EcadEcad_2" name="Ecad.Ecad_2" compartment="membrane"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
    multi:speciesType="st_Ecad_cis_dimer">
    <multi:listOfOutwardBindings>
        <multi:outwardBindingSite multi:component="Ecad1trans"
        multi:bindingStatus="bound" />
        <multi:outwardBindingSite multi:component="Ecad2trans"
        multi:bindingStatus="bound" />
    </multi:listOfOutwardBindings>
</species>

<!-- Pattern species: Ecad trans dimer -->
<species id="sp_EcadEcad_1" name="Ecad.Ecad_1" compartment="inter_membrane"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
    multi:speciesType="st_Ecad_trans_dimer" />

<!-- Pattern species: Ecad trans dimer: all cis bnd -->
<species id="sp_Ecad_trans_dimer_2" name="Ecad_trans_dimer_2" compartment="inter_membrane"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false"
    multi:speciesType="st_Ecad_trans_dimer">
    <multi:listOfOutwardBindings>
        <multi:outwardBindingSite multi:component="Ecad1cis"
        multi:bindingStatus="bound" />
        <multi:outwardBindingSite multi:component="Ecad2cis"
        multi:bindingStatus="bound" />
    </multi:listOfOutwardBindings>
</species>

<!-- Pattern species: Ecad True Trimer -->
<species id="sp_Ecad_True_Tramer" compartment="inter_membrane" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false" multi:speciesType="st_Ecad_trimer">
    <multi:listOfOutwardBindings>
        <multi:outwardBindingSite multi:component="Ecad2trans"
        multi:bindingStatus="unbound" />
        <multi:outwardBindingSite multi:component="Ecad3cis"
        multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindings>
</species>

<!-- Pattern species: Ecad All Trimer -->
<species id="sp_Ecad_All_Tramer" compartment="inter_membrane" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false" multi:speciesType="st_Ecad_trimer" />
</listOfSpecies>

```

```

<!-- Reactions -->
<listOfReactions>

  <!-- cis association: -->
  <reaction id="rc_Cis_Association" name="Cis_Association" reversible="false" fast="false"
    compartment="membrane">
    <listOfReactants>
      <speciesReference id="Cis_Association_r1" species="sp_Ecad_6"
        stoichiometry="1" constant="false" />
      <speciesReference id="Cis_Association_r2" species="sp_Ecad_6"
        stoichiometry="1" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_EcadEcad_2" constant="false" />
    </listOfProducts>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci> kon </ci>
          <ci multi:speciesReference="Cis_Association_r1"> sp_Ecad_6 </ci>
          <ci multi:speciesReference="Cis_Association_r2"> sp_Ecad_6 </ci>
        </apply>
      </math>
      <listOfLocalParameters>
        <localParameter id="kon" value="9000" units="litre_per_mole_per_sec" />
      </listOfLocalParameters>
    </kineticLaw>
  </reaction>

  <!-- In species cis association: Here the model requires that the two interacting molecules
    are part of one connected complex already prior to the association. Since the necessary
    connectivity can only be mediated by the trans binding sites here, these sites must be
    bound to the subcomplex (not shown) linking the two interacting molecules.
  -->
  <reaction id="rc_Intra_Complex_Cis_Association" name="Intra-Complex_Cis_Association"
    reversible="false" fast="false" compartment="membrane"
    multi:isIntraSpeciesReaction="true">
    <listOfReactants>
      <speciesReference id="Intra_Complex_Cis_Association_r1" species="sp_Ecad_6"
        stoichiometry="1" constant="false" />
      <speciesReference id="Intra_Complex_Cis_Association_r2" species="sp_Ecad_6"
        stoichiometry="1" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_EcadEcad_2" constant="false" />
    </listOfProducts>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci> kon </ci>
          <ci multi:speciesReference="Intra_Complex_Cis_Association_r1"> sp_Ecad_6 </ci>
        </apply>
      </math>
      <listOfLocalParameters>
        <localParameter id="kon" value="100" units="per_sec" />
      </listOfLocalParameters>
    </kineticLaw>
  </reaction>

  <!-- trans association: -->
  <reaction id="rc_Trans_Association" name="Trans_Association" reversible="false" fast="false"
    compartment="inter_membrane">
    <listOfReactants>

```

```

    <speciesReference id="Trans_Association_r1" species="sp_Ecad_trans_unbnd"
      compartmentReference="m1" constant="false" />
    <speciesReference id="Trans_Association_r2" species="sp_Ecad_trans_unbnd"
      compartmentReference="m2" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="sp_EcadEcad_1" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> kon </ci>
        <ci multi:speciesReference="Trans_Association_r1"> sp_Ecad_trans_unbnd </ci>
        <ci multi:speciesReference="Trans_Association_r2"> sp_Ecad_trans_unbnd </ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="kon" value="90000" units="litre_per_mole_per_sec" />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>

<!-- In complex trans association: Here the model requires that the two interacting molecules
are part of one connected complex already prior to the association. Since the necessary
connectivity can only be mediated by the cis binding sites here, these sites must be bound
to the subcomplex (not shown) linking the two interacting molecules.
-->
<reaction id="rc_Intra_Complex_Trans_Association" name="Intra-Complex_Trans_Association"
  reversible="false" fast="false" compartment="inter_membrane"
  multi:isIntraSpeciesReaction="true">
  <listOfReactants>
    <speciesReference id="Intra_Complex_Trans_Association_r1" species="sp_Ecad_7"
      compartmentReference="m1" constant="false" />
    <speciesReference id="Intra_Complex_Trans_Association_r2" species="sp_Ecad_7"
      compartmentReference="m2" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="sp_Ecad_trans_dimer_2" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> kon </ci>
        <ci multi:speciesReference="Intra_Complex_Trans_Association_r1"> sp_Ecad_7 </ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="kon" value="100" units="per_sec" />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>

<!-- cis dissociation: -->
<reaction id="rc_Cis_dissociation" name="Cis_dissociation" reversible="false" fast="false"
  compartment="membrane">
  <listOfReactants>
    <speciesReference species="sp_Ecad_cis_dimer" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference id="Cis_dissociation_p1" species="sp_Ecad_cis_unbnd"
      stoichiometry="1" constant="false" />
    <speciesReference id="Cis_dissociation_p2" species="sp_Ecad_cis_unbnd"
      stoichiometry="1" constant="false" />
  </listOfProducts>

```

```

<kineticLaw>
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
      <times />
      <ci> koff </ci>
      <ci> sp_Ecad_cis_unbnd </ci>
    </apply>
  </math>
  <listOfLocalParameters>
    <localParameter id="koff" value="1" units="per_sec" />
  </listOfLocalParameters>
</kineticLaw>
</reaction>

<!-- In-species cis dissociation: By specifying that this reaction breaks only an inner bond,
the model limits the application of this reaction to dissociations that result in only one
reaction product. The complex is still connected through a subcomplex that is not shown
here but that links the two molecules involved in the reaction at their trans binding
sites. Note that the modeler application has to ensure the correct application of this
rule and its consistent definition. For instance, specifying the one or both of the trans
binding sites to be unbound would lead to a rule that could never be applied because the
trans bindings are required for the connectivity of the result complex.
-->
<reaction id="rc_Intra_Complex_Cis_dissociation" name="Intra-Complex_Cis_dissociation"
  reversible="false" fast="false" compartment="membrane"
  multi:isIntraSpeciesReaction="true">
  <listOfReactants>
    <speciesReference species="sp_EcadEcad_2" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference id="Intra_Complex_Cis_dissociation_p1" species="sp_Ecad_6"
      stoichiometry="2" constant="false" />
    <speciesReference id="Intra_Complex_Cis_dissociation_p2" species="sp_Ecad_6"
      stoichiometry="2" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> koff </ci>
        <ci> sp_Ecad_6 </ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="koff" value="0.01" units="per_sec" />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>

<!-- trans dissociation: -->
<reaction id="rc_Trans_dissociation" name="Trans_dissociation" reversible="false"
  fast="false" compartment="inter_membrane">
  <listOfReactants>
    <speciesReference species="sp_EcadEcad_1" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference id="Trans_dissociation_p1" species="sp_Ecad_trans_unbnd"
      multi:compartmentReference="m1" constant="false" />
    <speciesReference id="Trans_dissociation_p2" species="sp_Ecad_trans_unbnd"
      multi:compartmentReference="m2" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> koff </ci>

```

```

        <ci> sp_Ecad_trans_unbnd </ci>
      </apply>
    </math>
  <listOfLocalParameters>
    <localParameter id="koff" value="1" units="per_sec" />
  </listOfLocalParameters>
</kineticLaw>
</reaction>

<!-- In species trans dissociation: By specifying that this reaction breaks only an inner
bond, the model limits the application of this reaction to dissociations that result in
only one reaction product. The complex is still connected through a subcomplex that is
not shown here but that links the two molecules involved in the reaction at their cis
binding sites. Note that the modeler application has to ensure the correct application
of this rule and its consistent definition. For instance, specifying the one or both of
the cis binding sites to be unbound would lead to a rule that could never be applied
because the cis bindings are required for the connectivity of the result complex.
-->
<reaction id="rc_Intra_Complex_Trans_dissociation" name="Intra-Complex_Trans_dissociation"
reversible="false" fast="false" compartment="inter_membrane"
multi:isIntraSpeciesReaction="true">
  <listOfReactants>
    <speciesReference species="sp_Ecad_trans_dimer_2" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference id="Intra_Complex_Trans_dissociation_p1" species="sp_Ecad_7"
      compartmentReference="m1" constant="false" />
    <speciesReference id="Intra_Complex_Trans_dissociation_p2" species="sp_Ecad_7"
      compartmentReference="m2" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci> koff </ci>
        <ci> sp_Ecad_7 </ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="koff" value="0.01" units="per_sec" />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>
</listOfReactions>
</model>
</sbml>

```

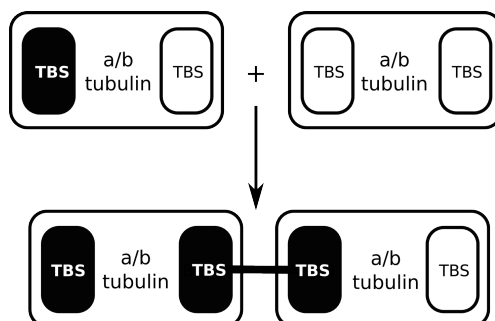
### 4.3 The “microtubule” example in the previous *Multi* proposal (2010)

This example is from the previous *Multi* proposal [Novère and Oellrich (2010)] and Figure 19 on the next page is the figure 3.2 in the previous *Multi* proposal (2010). An empty “TBS” icon represents an “unbound” binding site and a “TBS” icon filled with black represents a “bound” binding site. The example can be transformed into the SBML Level 3 format with use of the *Multi* package described in this specification as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">
  <model name="microtubule_elongation">
    <!-- Compartment -->
    <listOfCompartments>
      <compartment id="cell" constant="true" size="1" multi:isType="false" />
    </listOfCompartments>
  </model>
</sbml>

```



**Figure 19:** The figure 3.2 in the previous *Multi* proposal [Novère and Oellrich (2010)]: “A microtubule of undefined length can be extended through a polymerization”

```

<!-- SpeciesType -->
<multi:listOfSpeciesTypes>

  <!-- TBS: tubulin binding site -->
  <multi:speciesType multi:id="st_tbs" multi:isBindingSite="true" />

  <!-- Tubulin -->
  <multi:speciesType multi:id="st_tubulin" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:component multi:id="tbs" multi:speciesType="st_tbs" multi:occur="2" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
      <multi:speciesTypeComponentIndex multi:id="b1"
        multi:component="tbs" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="b2"
        multi:component="tbs" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
  </multi:speciesType>

  <!-- Microtubule -->
  <multi:speciesType multi:id="st_microtubule" multi:isBindingSite="false">
    <multi:listOfSpeciesTypeInstances>
      <multi:component multi:id="tubulin" multi:speciesType="st_tubulin"
        multi:occur="2" />
    </multi:listOfSpeciesTypeInstances>
    <multi:listOfSpeciesTypeComponentIndexes>
      <multi:speciesTypeComponentIndex multi:id="tubulin1"
        multi:component="tubulin" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="tubulin2"
        multi:component="tubulin" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="t1b1"
        multi:component="tbs" multi:identifyingParent="tubulin1" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="t1b2"
        multi:component="tbs" multi:identifyingParent="tubulin1" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="t2b1"
        multi:component="tbs" multi:identifyingParent="tubulin2" multi:occur="1" />
      <multi:speciesTypeComponentIndex multi:id="t2b2"
        multi:component="tbs" multi:identifyingParent="tubulin2" multi:occur="1" />
    </multi:listOfSpeciesTypeComponentIndexes>
    <multi:listOfInSpeciesTypeBonds>
      <multi:inSpeciesTypeBond multi:bindingSite1="t1b2"
        multi:bindingSite2="t2b1" />
    </multi:listOfInSpeciesTypeBonds>
  </multi:speciesType>
</multi:listOfSpeciesTypes>

<!-- Species -->
<listOfSpecies>

```



```

<!-- Fully defined free tubulin species -->
<species id="sp_free_tubulin" name="free_tubulin" multi:speciesType="st_tubulin"
  compartment="cell" hasOnlySubstanceUnits="false" boundaryCondition="false"
  constant="false" initialAmount="100">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="b1"
      multi:bindingStatus="unbound" />
    <multi:outwardBindingSite multi:component="b2"
      multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
</species>

<!-- Fully defined free microtubule species: 2 tubulin bound together -->
<species id="sp_microtubule_free" name="microtubule_free" multi:speciesType="st_microtubule"
  compartment="cell" hasOnlySubstanceUnits="false" boundaryCondition="false"
  constant="false" initialAmount="200">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="t1b1"
      multi:bindingStatus="unbound" />
    <multi:outwardBindingSite multi:component="t2b2"
      multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
</species>

<!-- Pattern tubulin species: one binding site is free and another binding site is
dont care -->
<species id="sp_tubulin_pattern" name="tubulin_pattern" multi:speciesType="st_tubulin"
  compartment="cell" hasOnlySubstanceUnits="false" boundaryCondition="false"
  constant="false">
  <multi:listOfOutwardBindingSites>
    <!-- b1: dont care -->
    <multi:outwardBindingSite multi:component="b2"
      multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
</species>

<!-- Pattern microtubule species: 2 tubulin bound together, one end is free and another
end is dont care -->
<species id="sp_microtubule_pattern" name="microtubule_pattern"
  multi:speciesType="st_microtubule" compartment="cell" hasOnlySubstanceUnits="false"
  boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="t2b2"
      multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
</species>
</listOfSpecies>

<!-- Reaction -->
<listOfReactions>

  <!-- Pattern tubulin elongated with a free tubulin -->
  <reaction id="tubulin_binding" reversible="false" fast="false">
    <listOfReactants>
      <speciesReference species="sp_free_tubulin" constant="false" />
      <speciesReference species="sp_tubulin_pattern" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_microtubule_pattern" constant="false">
        <multi:listSpeciesTypeComponentMapsInProduct>
          <multi:speciesComponentMapInProduct
            multi:reactantComponent="b1" multi:reactant="sp_tubulin_pattern"
            multi:productComponent="t1b1" />
          <multi:speciesComponentMapInProduct
            multi:reactantComponent="b2" multi:reactant="sp_tubulin_pattern"
            multi:productComponent="t1b2" />
        </multi:listSpeciesTypeComponentMapsInProduct>
      </speciesReference>
    </listOfProducts>
  </reaction>
</listOfReactions>

```

```

        <multi:speciesComponentMapInProduct
          multi:reactantComponent="b1" multi:reactant="sp_free_tubulin"
          multi:productComponent="t2b1" />
        <multi:speciesComponentMapInProduct
          multi:reactantComponent="b2" multi:reactant="sp_free_tubulin"
          multi:productComponent="t2b2" />
      </multi:listSpeciesTypeComponentMapsInProduct>
    </speciesReference>
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci>cell</ci>
        <ci>kon</ci>
        <ci>sp_free_tubulin</ci>
        <ci>sp_tubulin_pattern</ci>
      </apply>
    </math>
    <listOfLocalParameters>
      <localParameter id="kon" value="1" />
    </listOfLocalParameters>
  </kineticLaw>
</reaction>
</listOfReactions>
</model>
</sbml>

```

## 4.4 A BioNetGen example from its user manual

egfr\_simple.bngl ([http://bionetgen.org/index.php/BNGManual:Listing\\_1](http://bionetgen.org/index.php/BNGManual:Listing_1))

```

begin parameters
  NA 6.02e23          # Avogadro's number (molecules/mol)
  f 1                 # Fraction of the cell to simulate
  Vo f*1.0e-10        # Extracellular volume=1/cell_density (L)
  V f*3.0e-12         # Cytoplasmic volume (L)

  EGF_init 20*1e-9*NA*Vo # Initial amount of ligand (20 nM)
                        # converted to copies per cell

  # Initial amounts of cellular components (copies per cell)
  EGFR_init f*1.8e5
  Grb2_init f*1.5e5
  Sos1_init f*6.2e4

  # Rate constants
  # Divide by NA*V to convert bimolecular rate constants
  # from /M/sec to /(molecule/cell)/sec
  kp1 9.0e7/(NA*Vo)    # ligand-monomer binding
  km1 0.06             # ligand-monomer dissociation
  kp2 1.0e7/(NA*V)     # aggregation of bound monomers
  km2 0.1              # dissociation of bound monomers
  kp3 0.5              # dimer transphosphorylation
  km3 4.505            # dimer dephosphorylation
  kp4 1.5e6/(NA*V)     # binding of Grb2 to receptor
  km4 0.05             # dissociation of Grb2 from receptor
  kp5 1.0e7/(NA*V)     # binding of Grb2 to Sos1
  km5 0.06             # dissociation of Grb2 from Sos1
  deg 0.01             # degradation of receptor dimers
end parameters

begin molecule types
  EGF(R)

```

```

EGFR(L,CR1,Y1068~U~P)
Grb2(SH2,SH3)
Sos1(PxxP)
Trash()
end molecule types

begin seed species
  EGF(R)          0
  EGFR(L,CR1,Y1068~U) EGFR_init
  Grb2(SH2,SH3)   Grb2_init
  Sos1(PxxP)       Sos1_init
end seed species

begin observables
  1 Molecules EGFR_tot EGFR()
  2 Molecules Lig_free EGF(R)
  3 Species Dim EGFR(CR1!+)
  4 Molecules RP EGFR(Y1068~P!?)
  5 Molecules Grb2Sos1 Grb2(SH2,SH3!1).Sos1(PxxP!1)
  6 Molecules Sos1_act EGFR(Y1068!1).Grb2(SH2!1,SH3!2).Sos1(PxxP!2)
end observables

begin reaction rules
  # Ligand-receptor binding
  1 EGFR(L,CR1) + EGF(R) <-> EGFR(L!1,CR1).EGF(R!1) kp1, km1

  # Receptor-aggregation
  2 EGFR(L!+,CR1) + EGFR(L!+,CR1) <-> EGFR(L!+,CR1!1).EGFR(L!+,CR1!1) kp2,km2

  # Transphosphorylation of EGFR by RTK
  3 EGFR(CR1!+,Y1068~U) -> EGFR(CR1!+,Y1068~P) kp3

  # Dephosphorylation
  4 EGFR(Y1068~P) -> EGFR(Y1068~U) km3

  # Grb2 binding to pY1068
  5 EGFR(Y1068~P) + Grb2(SH2) <-> EGFR(Y1068~P!1).Grb2(SH2!1) kp4,km4

  # Grb2 binding to Sos1
  6 Grb2(SH3) + Sos1(PxxP) <-> Grb2(SH3!1).Sos1(PxxP!1) kp5,km5

  # Receptor dimer internalization/degradation
  7 EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3) -> Trash()
end reaction rules

#actions
generate_network({overwrite=>1});

# Equilibration
simulate_ode({suffix=>equil,t_end=>100000,n_steps=>10,sparse=>1,steady_state=>1});
setConcentration("EGF(R)","EGF_init");
saveConcentrations(); # Saves concentrations for future reset

# Kinetics
writeSBML({});
simulate_ode({t_end=>120,n_steps=>120});
resetConcentrations(); # reverts to saved Concentrations
simulate_ssa({suffix=>ssa,t_end=>120,n_steps=>120});

```

The SBML code can be as follows. Please note, the SBML code does not cover the content other than the model in the bngl file, such as the “actions”, “Equilibration” and “Kinetics” sections.

```

<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">

```

```

<model name="bionetgen_example_egfr_simple">

  <listOfUnitDefinitions>
    <unitDefinition id="molecules_per_mol">
      <listOfUnits>
        <unit kind="mole" scale="0" multiplier="1" exponent="-1" />
      </listOfUnits>
    </unitDefinition>
  </listOfUnitDefinitions>

  <!-- compartments -->
  <listOfCompartments>
    <compartment id="Vo" constant="true" spatialDimensions="3" units="liter"
      multi:isType="false" />
    <compartment id="V" constant="true" spatialDimensions="3" units="liter"
      multi:isType="false" />
  </listOfCompartments>

  <!-- speciesType -->
  <multi:listOfSpeciesTypes>

    <!-- EGF(R) -->
    <multi:speciesType multi:id="st_EGF_bs_R" multi:isBindingSite="true" />
    <multi:speciesType multi:id="st_EGF" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:component multi:id="R" multi:speciesType="st_EGF_bs_R" multi:occur="1" />
      </multi:listOfSpeciesTypeInstances>
    </multi:speciesType>

    <!-- EGFR(L,CR1,Y1068~U~P) -->
    <multi:speciesType multi:id="st_EGFR_bs_L" multi:isBindingSite="true" />
    <multi:speciesType multi:id="st_EGFR_bs_CR1" multi:isBindingSite="true" />
    <multi:speciesType multi:id="st_EGFR_bs_Y1068" multi:isBindingSite="true">
      <multi:listOfSpeciesFeatureTypes>
        <multi:speciesFeatureType multi:id="sft_Y1068">
          <multi:possibleSpeciesFeatureValue multi:id="U" />
          <multi:possibleSpeciesFeatureValue multi:id="P" />
        </multi:speciesFeatureType>
      </multi:listOfSpeciesFeatureTypes>
    </multi:speciesType>
    <multi:speciesType multi:id="st_EGFR" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:component multi:id="L" multi:speciesType="st_EGFR_bs_L" multi:occur="1" />
        <multi:component multi:id="CR1" multi:speciesType="st_EGFR_bs_CR1" multi:occur="1" />
        <multi:component multi:id="Y1068" multi:speciesType="st_EGFR_bs_Y1068"
          multi:occur="1" />
      </multi:listOfSpeciesTypeInstances>
    </multi:speciesType>

    <!-- EGFR dimer: [EGFR(CR1!1).EGFR(CR1!1)] -->
    <multi:speciesType multi:id="st_EGFR_dimer" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:component multi:id="EGFR" multi:speciesType="st_EGFR" multi:occur="2" />
      </multi:listOfSpeciesTypeInstances>
      <multi:listOfSpeciesTypeComponentIndexes>
        <multi:speciesTypeComponentIndex multi:id="EGFR1"
          multi:component="EGFR" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="EGFR2"
          multi:component="EGFR" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="EGFR1CR1"
          multi:component="CR1" identifyingParent="EGFR1" multi:occur="1" />
        <multi:speciesTypeComponentIndex multi:id="EGFR2CR1"
          multi:component="CR1" identifyingParent="EGFR2" multi:occur="1" />
      </multi:listOfSpeciesTypeComponentIndexes>
    </multi:speciesType>
  </multi:listOfInSpeciesTypeBonds>

```

```

    <multi:inSpeciesTypeBond multi:bindingSite1="EGFR1CR1"
      multi:bindingSite2="EGFR2CR1" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- EGFR-EGF dimer: [EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3)] -->
<multi:speciesType multi:id="st_EGFR_EGF_dimer" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="EGF" multi:speciesType="st_EGF" multi:occur="2" />
    <multi:component multi:id="EGFR" multi:speciesType="st_EGFR" multi:occur="2" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfSpeciesTypeComponentIndexes>
    <multi:speciesTypeComponentIndex multi:id="EGFR1"
      multi:component="EGFR" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGFR2"
      multi:component="EGFR" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGF1"
      multi:component="EGF" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGF2"
      multi:component="EGF" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGF1R"
      multi:component="R" identifyingParent="EGF1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGF2R"
      multi:component="R" identifyingParent="EGF2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGFR1L"
      multi:component="L" identifyingParent="EGFR1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGFR2L"
      multi:component="L" identifyingParent="EGFR2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGFR1CR1"
      multi:component="CR1" identifyingParent="EGFR1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="EGFR2CR1"
      multi:component="CR1" identifyingParent="EGFR2" multi:occur="1" />
  </multi:listOfSpeciesTypeComponentIndexes>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="EGFR1CR1" multi:bindingSite2="EGFR2CR1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="EGF1R" multi:bindingSite2="EGFR1L" />
    <multi:inSpeciesTypeBond multi:bindingSite1="EGF2R" multi:bindingSite2="EGFR2L" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- Grb2(SH2, SH3) -->
<multi:speciesType multi:id="st_Grb2_bs_SH2" multi:isBindingSite="true" />
<multi:speciesType multi:id="st_Grb2_bs_SH3" multi:isBindingSite="true" />
<multi:speciesType multi:id="st_Grb2" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="SH2" multi:speciesType="st_Grb2_bs_SH2" multi:occur="1" />
    <multi:component multi:id="SH3" multi:speciesType="st_Grb2_bs_SH3" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
</multi:speciesType>

<!-- Sos1 -->
<multi:speciesType multi:id="st_Sos1_bs_PxxP" multi:isBindingSite="true" />
<multi:speciesType multi:id="st_Sos1" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="PxxP" multi:speciesType="st_Sos1_bs_PxxP"
      multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
</multi:speciesType>

<!-- Trash -->
<multi:speciesType multi:id="trash" multi:isBindingSite="false" />

<!-- Grb2-Sos1 -->
<multi:speciesType multi:id="st_Grb2_Sos1" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="Grb2" multi:speciesType="st_Grb2" multi:occur="1" />

```

```

    <multi:component multi:id="Sos1" multi:speciesType="st_Sos1" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="SH3" multi:bindingSite2="PxxP" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- EGFR(Y1068!1).Grb1(SH2!1,SH3!2).Sos1(PxxP!2) -->
<multi:speciesType multi:id="st_EGFR_Grb2_Sos1" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="EGFR" multi:speciesType="st_EGFR" multi:occur="1" />
    <multi:component multi:id="Grb2" multi:speciesType="st_Grb2" multi:occur="1" />
    <multi:component multi:id="Sos1" multi:speciesType="st_Sos1" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="Y1068" multi:bindingSite2="SH2" />
    <multi:inSpeciesTypeBond multi:bindingSite1="SH3" multi:bindingSite2="PxxP" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- EGFR(L!1).EGF(R!1) -->
<multi:speciesType multi:id="st_EGFR_EGF" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="EGFR" multi:speciesType="st_EGFR" multi:occur="1" />
    <multi:component multi:id="EGF" multi:speciesType="st_EGF" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="L" multi:bindingSite2="R" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<!-- EGFR(Y1068!1).Grb2(SH2!1) -->
<multi:speciesType multi:id="st_EGFR_Grb2" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:component multi:id="EGFR" multi:speciesType="st_EGFR" multi:occur="1" />
    <multi:component multi:id="Grb2" multi:speciesType="st_Grb2" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="Y1068" multi:bindingSite2="SH2" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
</multi:listOfSpeciesTypes>

<!-- species -->
<listOfSpecies>

  <species id="sp_EGF_free" name="EGF(R)" multi:speciesType="st_EGF"
    hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="R" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>

  <species id="sp_EGFR_free_U" name="EGFR(L,CR1,Y1068~U)" multi:speciesType="st_EGFR"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="L" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="Y1068" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
        <multi:speciesFeatureValue multi:value="U" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
  </species>

```

```

    </multi:listOfSpeciesFeatures>
  </species>
  <species id="sp_Grb2_free" name="Grb2(SH2,SH3)" multi:speciesType="st_Grb2"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="SH2" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="SH3" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_Grb2_SH2" name="Grb2(SH2)" multi:speciesType="st_Grb2"
    hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="SH2" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_Grb2_SH3" name="Grb2(SH3)" multi:speciesType="st_Grb2"
    hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="SH3" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_Sos1_free" name="Sos1(PxxP)" multi:speciesType="st_Sos1"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="PxxP" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_EGF_tot" name="EGF()" multi:speciesType="st_EGF"
    hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false" />
  <species id="sp_EGFR_dimerized" name="EGFR(CR1!+)" multi:speciesType="st_EGFR"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="bound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_EGFR_U" name="EGFR(Y1068~P!?)" multi:speciesType="st_EGFR"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfSpeciesFeatures>
      <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
        <multi:speciesFeatureValue multi:value="P" />
      </multi:speciesFeature>
    </multi:listOfSpeciesFeatures>
  </species>
  <species id="sp_EGFR_L_CR1" name="EGFR(L,CR1)" multi:speciesType="st_EGFR"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="L" multi:bindingStatus="unbound" />
      <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_EGFR_EGF_CR1" name="EGFR(L!1,CR1).EGF(R!1)" multi:speciesType="st_EGFR_EGF"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="sp_EGFR_bL_CR1" name="EGFR(L!+,CR1)" multi:speciesType="st_EGFR"
    hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite multi:component="L" multi:bindingStatus="bound" />
      <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="unbound" />
    </multi:listOfOutwardBindingSites>
  </species>

```

```

<species id="sp_EGFR_dimer_bL" name="EGFR(L!+,CR1!1).EGFR(L!+,CR1!1)"
  multi:speciesType="st_EGFR_dimer"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="EGFR1L"
      multi:bindingStatus="bound" />
    <multi:outwardBindingSite multi:component="EGFR2L"
      multi:bindingStatus="bound" />
  </multi:listOfOutwardBindingSites>
</species>
<species id="sp_EGFR_EGF_dimer" name="EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3)"
  multi:speciesType="st_EGFR_EGF_dimer" hasOnlySubstanceUnits="false"
  boundaryCondition="false"
  constant="false" />
<species id="sp_EGFR_bCR1_Y1068_U" name="EGFR(CR1!+,Y1068~U)" multi:speciesType="st_EGFR"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="bound" />
    <multi:outwardBindingSite multi:component="Y1068" multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
      <multi:speciesFeatureValue multi:value="U" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="sp_EGFR_bCR1_Y1068_P" name="EGFR(CR1!+,Y1068~P)" multi:speciesType="st_EGFR"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="CR1" multi:bindingStatus="bound" />
    <multi:outwardBindingSite multi:component="Y1068" multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
      <multi:speciesFeatureValue multi:value="P" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="sp_EGFR_Y1068_P" name="EGFR(Y1068~P)" multi:speciesType="st_EGFR"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="Y1068" multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
      <multi:speciesFeatureValue multi:value="P" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="sp_EGFR_Y1068_U" name="EGFR(Y1068~U)" multi:speciesType="st_EGFR"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="Y1068" multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
      <multi:speciesFeatureValue multi:value="U" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="sp_EGFR_Grb2_P" name="EGFR(Y1068~P!1).Grb2(SH2!1)"
  multi:speciesType="st_EGFR_Grb2"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="sft_Y1068">
      <multi:speciesFeatureValue multi:value="P" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>

```



```

    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
<species id="sp_Grb2_Sos1" name="Grb2(SH3!1).Sos1(PxxP!1)" multi:speciesType="st_Grb2_Sos1"
  hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false" />

  <species id="sp_Trash" name="Trash()" multi:speciesType="st_Trash"
    hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false" />
</listOfSpecies>

<!-- parameters -->
<listOfParameters>
  <parameter id="NA" value="6.02e23" constant="true" units="molecules_per_mol" />
  <parameter id="f" value="1" constant="true" />
  <parameter id="kp1" constant="true" />
  <parameter id="km1" value="0.06" constant="true" />
  <parameter id="kp2" constant="true" />
  <parameter id="km2" value="0.1" constant="true" />
  <parameter id="kp3" value="0.5" constant="true" />
  <parameter id="km3" value="4.505" constant="true" />
  <parameter id="kp4" constant="true" />
  <parameter id="km4" value="0.05" constant="true" />
  <parameter id="kp5" constant="true" />
  <parameter id="km5" value="0.06" constant="true" />
  <parameter id="deg" value="0.01" constant="true" />
</listOfParameters>

<!-- initialAssignments -->
<listOfInitialAssignments>

  <initialAssignment symbol="Vo">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci>f</ci>
        <cn> 1e-10 </cn>
      </apply>
    </math>
  </initialAssignment>
  <initialAssignment symbol="V">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci>f</ci>
        <cn> 3e-12 </cn>
      </apply>
    </math>
  </initialAssignment>
  <initialAssignment symbol="kp1">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <divide />
        <cn>9.02e7</cn>
        <apply>
          <times />
          <ci>NA</ci>
          <ci>Vo</ci>
        </apply>
      </apply>
    </math>
  </initialAssignment>
  <initialAssignment symbol="kp2">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <divide />

```

```

        <cn>1.0e7</cn>
        <apply>
          <times />
          <ci>NA</ci>
          <ci>V</ci>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="kp4">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <divide />
          <cn>1.5e6</cn>
          <apply>
            <times />
            <ci>NA</ci>
            <ci>V</ci>
          </apply>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="kp5">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <divide />
          <cn>1.0e7</cn>
          <apply>
            <times />
            <ci>NA</ci>
            <ci>V</ci>
          </apply>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="sp_EGF_free">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <cn>20</cn>
          <cn>1e-9</cn>
          <ci>NA</ci>
          <ci>Vo</ci>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="sp_EGFR_free_U">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci>f</ci>
          <cn>1.8e5</cn>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="sp_Grb2_free">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci>f</ci>
          <cn>1.5e5</cn>
        </apply>
      </math>
    </initialAssignment>
    <initialAssignment symbol="sp_Sos1_free">
      <math xmlns="http://www.w3.org/1998/Math/MathML">

```

```

        <apply>
          <times />
          <ci>f</ci>
          <cn>6.2e4</cn>
        </apply>
      </math>
    </initialAssignment>
  </listOfInitialAssignments>

  <!-- reactions -->
  <listOfReactions>

    <!-- # Ligand-receptor binding -->
    <!-- 1 EGFR(L,CR1) + EGF(R) <-> EGFR(L!1,CR1).EGF(R!1) kp1, km1 -->
    <reaction id="rc_Ligand_receptor_binding" reversible="true" fast="false">
      <listOfReactants>
        <speciesReference species="sp_EGFR_L_CR1" constant="false" />
        <speciesReference species="sp_EGF_free" constant="false" />
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="sp_EGFR_EGF_CR1" constant="false" />
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <minus />
            <apply>
              <times />
              <ci> kp1 </ci>
              <ci> sp_EGFR_L_CR1 </ci>
              <ci> sp_EGF_free </ci>
            </apply>
            <apply>
              <times />
              <ci> km1 </ci>
              <ci> sp_EGFR_EGF_CR1 </ci>
            </apply>
          </apply>
        </math>
      </kineticLaw>
    </reaction>

    <!-- # Receptor-aggregation -->
    <!-- 2 EGFR(L!+,CR1) + EGFR(L!+,CR1) <-> EGFR(L!+,CR1!1).EGFR(L!+,CR1!1) kp2,km2 -->
    <reaction id="rc_Receptor_aggregation" reversible="true" fast="false">
      <listOfReactants>
        <speciesReference species="sp_EGFR_bL_CR1" constant="false" stoichiometry="2" />
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="sp_EGFR_dimer_bL" constant="false" />
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <minus />
            <apply>
              <times />
              <ci> kp2 </ci>
              <ci> sp_EGFR_bL_CR1 </ci>
              <ci> sp_EGFR_bL_CR1 </ci>
            </apply>
            <apply>
              <times />
              <ci> km3 </ci>
              <ci> sp_EGFR_dimer_bL </ci>
            </apply>
          </apply>
        </math>
      </kineticLaw>
    </reaction>
  </listOfReactions>

```

```

        </apply>
      </math>
    </kineticLaw>
  </reaction>

  <!-- # Transphosphorylation of EGFR by RTK -->
  <!-- 3 EGFR(CR1!+,Y1068~U) -> EGFR(CR1!+,Y1068~P) kp3 -->
  <reaction id="rc_Transphosphorylation" reversible="false" fast="false">
    <listOfReactants>
      <speciesReference species="sp_EGFR_bCR1_Y1068_U" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_EGFR_bCR1_Y1068_P" constant="false" />
    </listOfProducts>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci>kp3</ci>
          <ci>sp_EGFR_bCR1_Y1068_U</ci>
        </apply>
      </math>
    </kineticLaw>
  </reaction>

  <!-- # Dephosphorylation -->
  <!-- 4 EGFR(Y1068~P) -> EGFR(Y1068~U) km3 -->
  <reaction id="rc_Dephosphorylation" reversible="false" fast="false">
    <listOfReactants>
      <speciesReference species="sp_EGFR_Y1068_P" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_EGFR_Y1068_U" constant="false" />
    </listOfProducts>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <times />
          <ci>km3</ci>
          <ci>sp_EGFR_Y1068_P</ci>
        </apply>
      </math>
    </kineticLaw>
  </reaction>

  <!-- # Grb2 binding to pY1068 -->
  <!-- 5 EGFR(Y1068~P) + Grb2(SH2) <-> EGFR(Y1068~P!1).Grb2(SH2!1) kp4,km4 -->
  <reaction id="rc_Grb2_binding_to_pY1068" reversible="true" fast="false">
    <listOfReactants>
      <speciesReference species="sp_EGFR_Y1068_P" constant="false" />
      <speciesReference species="sp_Grb2_SH2" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_EGFR_Grb2_P" constant="false" />
    </listOfProducts>
    <kineticLaw>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <minus />
          <apply>
            <times />
            <ci>kp4</ci>
            <ci>sp_EGFR_Y1068_P</ci>
            <ci>sp_Grb2_SH2</ci>
          </apply>
        </apply>
      </math>
    </kineticLaw>
  </reaction>

```

```

        <times />
        <ci> km4 </ci>
        <ci> sp_EGFR_Grb2_P </ci>
      </apply>
    </math>
  </kineticLaw>
</reaction>

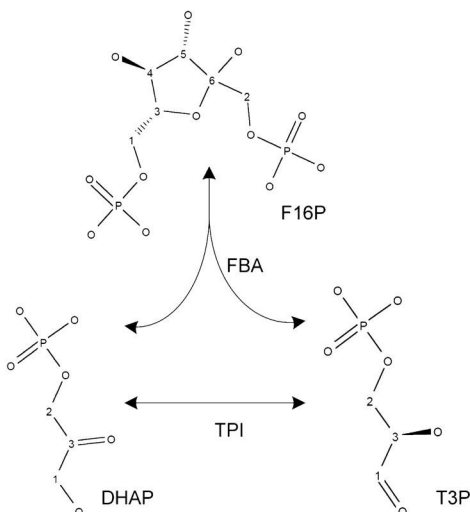
<!-- # Grb2 binding to Sos1 -->
<!-- 6 Grb2(SH3) + Sos1(PxxP) <-> Grb2(SH3!1).Sos1(PxxP!1) kp5,km5 -->
<reaction id="rc_Grb2_binding_to_Sos1" reversible="true" fast="false">
  <listOfReactants>
    <speciesReference species="sp_Grb2_SH3" constant="false" />
    <speciesReference species="sp_Sos1_free" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="sp_Grb2_Sos1" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <minus />
        <apply>
          <times />
          <ci> kp5 </ci>
          <ci> sp_Grb2_SH3 </ci>
          <ci> sp_Sos1_free </ci>
        </apply>
        <apply>
          <times />
          <ci> km5 </ci>
          <ci> sp_Grb2_Sos1 </ci>
        </apply>
      </apply>
    </math>
  </kineticLaw>
</reaction>

<!-- # Receptor dimer internalization/degradation -->
<!-- 7 EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3) -> Trash() -->
<reaction id="rc_EGFR_EGF_dimer_degradation" reversible="false" fast="false">
  <listOfReactants>
    <speciesReference species="sp_EGFR_EGF_dimer" constant="false" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="sp_Trash" constant="false" />
  </listOfProducts>
  <kineticLaw>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times />
        <ci>deg</ci>
        <ci>sp_EGFR_EGF_dimer</ci>
      </apply>
    </math>
  </kineticLaw>
</reaction>
</listOfReactions>
</model>
</sbml>

```

## 4.5 Mappings between *reactants* and *products* in a published model

This example coding the mappings between *reactants* and *products* is taken from a published *model* (Mu et al. (2007)).



**Figure 20:** The reactions of F16P, DHAP and T3P described in the paper by Mu et al. (2007)

Figure 20 is cited from the original paper and has the mappings between “F16P” and “DHAP + T3P” as follows:

1. Carbon atom 1 in F16P corresponds to Carbon atom 2 in T3P
2. Carbon atom 2 in F16P corresponds to Carbon atom 2 in DHAP
3. Carbon atom 3 in F16P corresponds to Carbon atom 3 in T3P
4. Carbon atom 4 in F16P corresponds to Carbon atom 1 in T3P
5. Carbon atom 5 in F16P corresponds to Carbon atom 1 in DHAP
6. Carbon atom 6 in F16P corresponds to Carbon atom 3 in DHAP

The SBML code in the format of the *Multi* package can be as follows.

**Note:**

In the biochemical structure diagram in Figure 20, all “binding sites” of oxygen and carbon atoms without explicit binding partners are understood to be bound to hydrogens. But those bonds are not important here for the purpose illustrating the mappings of carbon atoms and therefore the “binding sites” are simply assigned a “bound” state in the SBML code.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">
  <model name="Carbon-fate">
    <multi:listOfSpeciesTypes>
      <multi:speciesType multi:id="stCbindingSite" multi:isBindingSite="true" />
      <multi:speciesType multi:id="stC" multi:isBindingSite="false">
        <multi:listOfSpeciesTypeInstances>
          <multi:speciesTypeInstance multi:id="cbs" multi:speciesType="stCbindingSite"
            multi:occur="4" />
        </multi:listOfSpeciesTypeInstances>
      </multi:speciesType>
    </multi:listOfSpeciesTypes>
  </model>
</sbml>
```

```

</multi:speciesType>

<multi:speciesType multi:id="stObindingSite" multi:isBindingSite="true" />
<multi:speciesType multi:id="stO" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="obs" multi:speciesType="stObindingSite"
      multi:occur="2" />
  </multi:listOfSpeciesTypeInstances>
</multi:speciesType>

<multi:speciesType multi:id="stPbindingSite" multi:isBindingSite="true" />
<multi:speciesType multi:id="stP" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="pbs" multi:speciesType="stPbindingSite"
      multi:occur="5" />
  </multi:listOfSpeciesTypeInstances>
</multi:speciesType>

<multi:speciesType multi:id="stPhosphate" multi:isBindingSite="false">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="P" multi:speciesType="stP" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="O" multi:speciesType="stO" multi:occur="4" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfSpeciesTypeComponentIndexes>

    <!-- oxygens -->
    <multi:speciesTypeComponentIndex multi:id="O1" multi:component="O" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="O2" multi:component="O" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="O3" multi:component="O" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="O4" multi:component="O" multi:occur="1" />

    <!-- binding sites of oxygens -->
    <multi:speciesTypeComponentIndex multi:id="P01b1"
      multi:component="obs" multi:identifyingParent="O1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P01b2"
      multi:component="obs" multi:identifyingParent="O1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P02b1"
      multi:component="obs" multi:identifyingParent="O2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P02b2"
      multi:component="obs" multi:identifyingParent="O2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P03b1"
      multi:component="obs" multi:identifyingParent="O3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P03b2"
      multi:component="ob2" multi:identifyingParent="O3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="P04bb"
      multi:component="obs" multi:identifyingParent="O4" multi:occur="2" />

    <!-- binding sites of phosphorus -->
    <multi:speciesTypeComponentIndex multi:id="pb1"
      multi:component="pbs" multi:identifyingParent="P" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="pb2"
      multi:component="pbs" multi:identifyingParent="P" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="pb3"
      multi:component="pbs" multi:identifyingParent="P" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="pbb"
      multi:component="pbs" multi:identifyingParent="P" multi:occur="2" />

  </multi:listOfSpeciesTypeComponentIndexes>

  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="P01b1" multi:bindingSite2="pb1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="P02b1" multi:bindingSite2="pb2" />
    <multi:inSpeciesTypeBond multi:bindingSite1="P03b1" multi:bindingSite2="pb3" />
    <multi:inSpeciesTypeBond multi:bindingSite1="P04bb" multi:bindingSite2="pbb" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

```

```

<multi:speciesType multi:id="stDHAP">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="phosphate"
      multi:speciesType="stPhosphate" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="C2" multi:speciesType="stC" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="C3" multi:speciesType="stC" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="Oc3" multi:speciesType="stO" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="C1" multi:speciesType="stC" multi:occur="1" />
    <multi:speciesTypeInstance multi:id="Oc1" multi:speciesType="stO" multi:occur="1" />
  </multi:listOfSpeciesTypeInstances>
  <multi:listOfSpeciesTypeComponentIndexes>

    <!-- binding sites of C2 -->
    <multi:speciesTypeComponentIndex multi:id="C2b1"
      multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C2b23"
      multi:component="cbs" multi:identifyingParent="C2" multi:occur="2" />
    <multi:speciesTypeComponentIndex multi:id="C2b4"
      multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />

    <!-- binding sites of C3 -->
    <multi:speciesTypeComponentIndex multi:id="C3b1"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C3b23"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="2" />
    <multi:speciesTypeComponentIndex multi:id="C3b4"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />

    <!-- binding sites of C1 -->
    <multi:speciesTypeComponentIndex multi:id="C1b1"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C1b23"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="2" />
    <multi:speciesTypeComponentIndex multi:id="C1b4"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />

    <!-- binding sites of Oc3 -->
    <multi:speciesTypeComponentIndex multi:id="Oc3bb"
      multi:component="obs" multi:identifyingParent="Oc3" multi:occur="2" />

    <!-- binding sites of Oc1 -->
    <multi:speciesTypeComponentIndex multi:id="Oc1b1"
      multi:component="obs" multi:identifyingParent="Oc1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="Oc1b2"
      multi:component="obs" multi:identifyingParent="Oc1" multi:occur="1" />

  </multi:listOfSpeciesTypeComponentIndexes>

  <multi:listOfInSpeciesTypeBonds>
    <multi:inSpeciesTypeBond multi:bindingSite1="P01b2"
      multi:bindingSite2="C2b1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="C2b2"
      multi:bindingSite2="C3b1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="C3b23"
      multi:bindingSite2="Oc3bb" />
    <multi:inSpeciesTypeBond multi:bindingSite1="C3b4"
      multi:bindingSite2="C1b1" />
    <multi:inSpeciesTypeBond multi:bindingSite1="C1b2"
      multi:bindingSite2="Oc1b1" />
  </multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

<multi:speciesType multi:id="stT3P">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="phosphate"

```



```

    multi:speciesType="stPhosphate" multi:occur="1" />
<multi:speciesTypeInstance multi:id="C2" multi:speciesType="stC"
  multi:occur="1" />
<multi:speciesTypeInstance multi:id="C3" multi:speciesType="stC"
  multi:occur="1" />
<multi:speciesTypeInstance multi:id="Oc3" multi:speciesType="stO"
  multi:occur="1" />
<multi:speciesTypeInstance multi:id="C1" multi:speciesType="stC"
  multi:occur="1" />
<multi:speciesTypeInstance multi:id="Oc1" multi:speciesType="stO"
  multi:occur="1" />
</multi:listOfSpeciesTypeInstances>
<multi:listOfSpeciesTypeComponentIndexes>

  <!-- binding sites of C2 -->
  <multi:speciesTypeComponentIndex multi:id="C2b1"
    multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="C2b23"
    multi:component="cbs" multi:identifyingParent="C2" multi:occur="2" />
  <multi:speciesTypeComponentIndex multi:id="C2b4"
    multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />

  <!-- binding sites of C3 -->
  <multi:speciesTypeComponentIndex multi:id="C3b1"
    multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="C3b2"
    multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="C3b3"
    multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="C3b4"
    multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />

  <!-- binding sites of C1 -->
  <multi:speciesTypeComponentIndex multi:id="C1b1"
    multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="C1b23"
    multi:component="cbs" multi:identifyingParent="C1" multi:occur="2" />
  <multi:speciesTypeComponentIndex multi:id="C1b4"
    multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />

  <!-- binding sites of Oc3 -->
  <multi:speciesTypeComponentIndex multi:id="Oc3b1"
    multi:component="obs" multi:identifyingParent="Oc3" multi:occur="1" />
  <multi:speciesTypeComponentIndex multi:id="Oc3b2"
    multi:component="obs" multi:identifyingParent="Oc3" multi:occur="1" />

  <!-- binding sites of Oc1 -->
  <multi:speciesTypeComponentIndex multi:id="Oc1bb"
    multi:component="obs" multi:identifyingParent="Oc1" multi:occur="2" />

</multi:listOfSpeciesTypeComponentIndexes>

<multi:listOfInSpeciesTypeBonds>
  <multi:inSpeciesTypeBond multi:bindingSite1="P01b2"
    multi:bindingSite2="C2b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C2b2"
    multi:bindingSite2="C3b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C3b2"
    multi:bindingSite2="Oc3b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C3b4"
    multi:bindingSite2="C1b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C1b23"
    multi:bindingSite2="Oc1bb" />
</multi:listOfInSpeciesTypeBonds>
</multi:speciesType>

```

```

<multi:speciesType multi:id="stF16P">
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="phosphate1" multi:speciesType="stPhosphate"
      multi:occur="1" /> <!-- T3P phosphate -->
    <multi:speciesTypeInstance multi:id="C1" multi:speciesType="stC"
      multi:occur="1" /> <!-- T3P C2 -->
    <multi:speciesTypeInstance multi:id="C3" multi:speciesType="stC"
      multi:occur="1" /> <!-- T3P C3 -->
    <multi:speciesTypeInstance multi:id="Oc3" multi:speciesType="stO"
      multi:occur="1" /> <!-- T3P Oc3 -->
    <multi:speciesTypeInstance multi:id="C4" multi:speciesType="stC"
      multi:occur="1" /> <!-- T3P C1 -->
    <multi:speciesTypeInstance multi:id="Oc4" multi:speciesType="stO"
      multi:occur="1" /> <!-- T3P Oc1 -->
    <multi:speciesTypeInstance multi:id="phosphate2" multi:speciesType="stPhosphate"
      multi:occur="1" /> <!-- DHAP phosphate -->
    <multi:speciesTypeInstance multi:id="C2" multi:speciesType="stC"
      multi:occur="1" /> <!-- DHAP C2 -->
    <multi:speciesTypeInstance multi:id="C6" multi:speciesType="stC"
      multi:occur="1" /> <!-- DHAP C3 -->
    <multi:speciesTypeInstance multi:id="Oc6" multi:speciesType="stO"
      multi:occur="1" /> <!-- DHAP Oc3 -->
    <multi:speciesTypeInstance multi:id="C5" multi:speciesType="stC"
      multi:occur="1" /> <!-- DHAP C1 -->
    <multi:speciesTypeInstance multi:id="Oc5" multi:speciesType="stO"
      multi:occur="1" /> <!-- DHAP Oc1 -->
  </multi:listOfSpeciesTypeInstances>

  <multi:listOfSpeciesTypeComponentIndexes>
    <!-- Oxygen binding sites in phosphate1 -->
    <multi:speciesTypeComponentIndex multi:id="P101b2"
      multi:component="P01b2" multi:identifyingParent="phosphate1" />
    <multi:speciesTypeComponentIndex multi:id="P102b2"
      multi:component="P02b2" multi:identifyingParent="phosphate1" />
    <multi:speciesTypeComponentIndex multi:id="P103b2"
      multi:component="P03b2" multi:identifyingParent="phosphate1" />

    <!-- Oxygen binding sites in phosphate2 -->
    <multi:speciesTypeComponentIndex multi:id="P201b2"
      multi:component="P01b2" multi:identifyingParent="phosphate2" />
    <multi:speciesTypeComponentIndex multi:id="P202b2"
      multi:component="P02b2" multi:identifyingParent="phosphate2" />
    <multi:speciesTypeComponentIndex multi:id="P203b2"
      multi:component="P03b2" multi:identifyingParent="phosphate2" />

    <!-- binding sites of C1 -->
    <multi:speciesTypeComponentIndex multi:id="C1b1"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C1b23"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="2" />
    <multi:speciesTypeComponentIndex multi:id="C1b4"
      multi:component="cbs" multi:identifyingParent="C1" multi:occur="1" />

    <!-- binding sites of C3 -->
    <multi:speciesTypeComponentIndex multi:id="C3b1"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C3b2"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C3b3"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />
    <multi:speciesTypeComponentIndex multi:id="C3b4"
      multi:component="cbs" multi:identifyingParent="C3" multi:occur="1" />

    <!-- binding sites of Oc3 -->
    <multi:speciesTypeComponentIndex multi:id="Oc3b1"
      multi:component="obs" multi:identifyingParent="Oc3" multi:occur="1" />

```

```

<multi:speciesTypeComponentIndex multi:id="Oc3b2"
  multi:component="obs" multi:identifyingParent="Oc3" multi:occur="1" />

<!-- binding sites of C4 -->
<multi:speciesTypeComponentIndex multi:id="C4b1"
  multi:component="cbs" multi:identifyingParent="C4" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C4b2"
  multi:component="cbs" multi:identifyingParent="C4" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C4b3"
  multi:component="cbs" multi:identifyingParent="C4" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C4b4"
  multi:component="cbs" multi:identifyingParent="C4" multi:occur="1" />

<!-- binding sites of Oc4 -->
<multi:speciesTypeComponentIndex multi:id="Oc4b1"
  multi:component="ob1" multi:identifyingParent="Oc4" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="Oc4b2"
  multi:component="ob2" multi:identifyingParent="Oc4" multi:occur="1" />

<!-- binding sites of C5 -->
<multi:speciesTypeComponentIndex multi:id="C5b1"
  multi:component="cbs" multi:identifyingParent="C5" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C5b2"
  multi:component="cbs" multi:identifyingParent="C5" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C5b3"
  multi:component="cbs" multi:identifyingParent="C5" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C5b4"
  multi:component="cbs" multi:identifyingParent="C5" multi:occur="1" />

<!-- binding sites of Oc5 -->
<multi:speciesTypeComponentIndex multi:id="Oc5b1"
  multi:component="obs" multi:identifyingParent="Oc5" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="Oc5b2"
  multi:component="obs" multi:identifyingParent="Oc5" multi:occur="1" />

<!-- binding sites of C6 -->
<multi:speciesTypeComponentIndex multi:id="C6b1"
  multi:component="cbs" multi:identifyingParent="C6" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C6b2"
  multi:component="cbs" multi:identifyingParent="C6" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C6b3"
  multi:component="cbs" multi:identifyingParent="C6" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C6b4"
  multi:component="cbs" multi:identifyingParent="C6" multi:occur="1" />

<!-- binding sites of Oc6 -->
<multi:speciesTypeComponentIndex multi:id="Oc6b1"
  multi:component="ob1" multi:identifyingParent="Oc6" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="Oc6b2"
  multi:component="ob2" multi:identifyingParent="Oc6" multi:occur="1" />

<!-- binding sites of C2 -->
<multi:speciesTypeComponentIndex multi:id="C2b1"
  multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />
<multi:speciesTypeComponentIndex multi:id="C2b2"
  multi:component="cbs" multi:identifyingParent="C2" multi:occur="1" />

</multi:listOfSpeciesTypeComponentIndexes>

<multi:listOfInSpeciesTypeBonds>
  <multi:inSpeciesTypeBond multi:bindingSite1="P101b2"
    multi:bindingSite2="C1b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C1b4"
    multi:bindingSite2="C3b1" />
  <multi:inSpeciesTypeBond multi:bindingSite1="C3b2"
    multi:bindingSite2="Oc3b1" />

```

```

<multi:inSpeciesTypeBond multi:bindingSite1="C3b4"
  multi:bindingSite2="C4b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C4b2"
  multi:bindingSite2="Oc4b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C4b4"
  multi:bindingSite2="C5b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C5b2"
  multi:bindingSite2="Oc5b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C5b4"
  multi:bindingSite2="C6b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C6b2"
  multi:bindingSite2="Oc3b2" />
<multi:inSpeciesTypeBond multi:bindingSite1="C6b3"
  multi:bindingSite2="Oc6b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C6b4"
  multi:bindingSite2="C2b1" />
<multi:inSpeciesTypeBond multi:bindingSite1="C2b2"
  multi:bindingSite2="P201b2" />
</multi:listOfInSpeciesTypeBonds>
</multi:speciesType>
</multi:listOfSpeciesTypes>
<listOfSpecies>
  <species id="F16P" multi:speciesType="stF16P" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:listOfOutwardBindingSites>
      <multi:outwardBindingSite component="C1b23" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="C3b3" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="Oc4b2" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="C5b3" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="Oc6b2" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="C2b23" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="P102b2" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="P103b2" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="P202b2" multi:bindingStatus="bound" />
      <multi:outwardBindingSite component="P203b2" multi:bindingStatus="bound" />
    </multi:listOfOutwardBindingSites>
  </species>
  <species id="DHAP" multi:speciesType="stDHAP" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:outwardBindingSite component="P02b2" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="P03b2" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="C2b23" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="C1b23" multi:bindingStatus="bound" />
  </species>
  <species id="T3P" multi:speciesType="stT3P" hasOnlySubstanceUnits="false"
    boundaryCondition="false" constant="false">
    <multi:outwardBindingSite component="P02b2" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="P03b2" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="C2b23" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="C3b3" multi:bindingStatus="bound" />
    <multi:outwardBindingSite component="C1b4" multi:bindingStatus="bound" />
  </species>
  <species id="FBA" ... />
  <species id="TPI" ... />
</listOfSpecies>
<listOfReactions>
  <reaction id="rc" reversible="true" fast="false">
    <listOfReactants>
      <speciesReference species="F16P" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="DHAP" constant="false">
        <multi:listOfSpeciesTypeComponentMapsInProduct>
          <multi:speciesTypeComponentMapInProduct
            multi:reactantComponent="C5" multi:reactant="F16P"
            multi:productComponent="C1" />
        </multi:listOfSpeciesTypeComponentMapsInProduct>
      </speciesReference>
    </listOfProducts>
  </reaction>

```

```

        <multi:speciesTypeComponentMapInProduct
          multi:reactantComponent="C2" multi:reactant="F16P"
          multi:productComponent="C2" />
        <multi:speciesTypeComponentMapInProduct
          multi:reactantComponent="C6" multi:reactant="F16P"
          multi:productComponent="C3" />
      </multi:listOfSpeciesTypeComponentMapsInProduct>
    </speciesReference>
    <speciesReference species="T3P" constant="false">
      <multi:listOfSpeciesTypeComponentMapsInProduct>
        <multi:speciesTypeComponentMapInProduct
          multi:reactantComponent="C4" multi:reactant="F16P"
          multi:productComponent="C1" />
        <multi:speciesTypeComponentMapInProduct
          multi:reactantComponent="C1" multi:reactant="F16P"
          multi:productComponent="C2" />
        <multi:speciesTypeComponentMapInProduct
          multi:reactantComponent="C3" multi:reactant="F16P"
          multi:productComponent="C3" />
      </multi:listOfSpeciesTypeComponentMapsInProduct>
    </speciesReference>
  </listOfProducts>
  <listOfModifierSpeciesReferences>
    <ModifierSpeciesReference species="FBA" />
  </listOfModifierSpeciesReferences>
  ...
</reaction>
<reaction reversible="true" fast="false">
  <listOfReactants>
    <speciesReference species="DHAP" />
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="T3P">
      <multi:listOfSpeciesTypeComponentMapsInProduct>
        <multi:speciesTypeComponentMapInProduct multi:reactantComponent="C1"
          multi:reactant="DHAP" multi:productComponent="C1" />
        <multi:speciesTypeComponentMapInProduct multi:reactantComponent="C2"
          multi:reactant="DHAP" multi:productComponent="C2" />
        <multi:speciesTypeComponentMapInProduct multi:reactantComponent="C3"
          multi:reactant="DHAP" multi:productComponent="C3" />
      </multi:listOfSpeciesTypeComponentMapsInProduct>
    </speciesReference>
  </listOfProducts>
  <listOfModifierSpeciesReferences>
    <ModifierSpeciesReference species="TPI" />
  </listOfModifierSpeciesReferences>
  ...
</reaction>
</listOfReactions>
</model>
</sbml>

```

## 4.6 Example from *Kappa*'s documentation

Here is the example “An Introduction to Kappa Syntax” at *Kappa* website (<http://www.kappalanguage.org/syntax.html>).

Rule in English: “Unphosphorylated Site1 of A binds to Site1 of B”

Kappa Rule: A(Site1 u),B(Site1) -> A(Site1 u!1),B(Site1!1)

```

<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" multi:required="true">

```

```

<model name="An_Introduction_to_Kappa_Syntax">
  ...
  <!-- speciesType -->
  <multi:listOfSpeciesTypes>

    <!-- A:Site1 -->
    <multi:speciesType multi:id="st_A_Site1" multi:isBindingSite="true">
      <multi:listOfSpeciesFeatureTypes>
        <multi:speciesFeatureType multi:id="phosphorylation">
          <multi:possibleSpeciesFeatureValue multi:id="U" />
          <multi:possibleSpeciesFeatureValue multi:id="P" />
        </multi:speciesFeatureType>
      </multi:listOfSpeciesFeatureTypes>
    </multi:speciesType>

    <!-- A -->
    <multi:speciesType multi:id="st_A" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:speciesTypeInstance multi:id="Asite1" multi:speciesType="st_A_Site1"
          multi:occur="1" />
      </multi:listOfSpeciesTypeInstances>
    </multi:speciesType>

    <!-- B:Site1 -->
    <multi:speciesType multi:id="st_B_Site1" multi:isBindingSite="true" />

    <!-- B -->
    <multi:speciesType multi:id="st_B" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:speciesTypeInstance multi:id="Bsite1" multi:speciesType="st_B_Site1"
          multi:occur="1" />
      </multi:listOfSpeciesTypeInstances>
    </multi:speciesType>

    <!-- A.B -->
    <multi:speciesType multi:id="st_AB" multi:isBindingSite="false">
      <multi:listOfSpeciesTypeInstances>
        <multi:speciesTypeInstance multi:id="A" multi:speciesType="st_A" multi:occur="1" />
        <multi:speciesTypeInstance multi:id="B" multi:speciesType="st_B" multi:occur="1" />
      </multi:listOfSpeciesTypeInstances>
      <multi:listOfInSpeciesTypeBonds>
        <multi:inSpeciesTypeBond multi:bindingSite1="Asite1"
          multi:bindingSite2="Bsite1" />
      </multi:listOfInSpeciesTypeBonds>
    </multi:speciesType>
    ...
  </multi:listOfSpeciesTypes>

  <!-- species -->
  <listOfSpecies>
    <!-- species A with free unphosphorylated Site1 -->
    <species id="sp_A" name="A_with_Unphosphorylated_Site_1" multi:speciesType="st_A"
      hasOnlySubstanceUnits="false" boundaryCondition="false" constant="false">
      <multi:listOfOutwardBindingSites>
        <multi:outwardBindingSite multi:component="Asite1"
          multi:bindingStatus="unbound" />
      </multi:listOfOutwardBindingSites>
      <multi:listOfSpeciesFeatures>
        <multi:speciesFeature multi:speciesFeatureType="phosphorylation">
          <multi:speciesFeatureValue multi:value="U" />
        </multi:speciesFeature>
      </multi:listOfSpeciesFeatures>
    </species>

    <!-- species B with free Site 1 -->

```

```

<species id="sp_B" name="B" multi:speciesType="st_B" hasOnlySubstanceUnits="false"
  boundaryCondition="false" constant="false">
  <multi:listOfOutwardBindingSites>
    <multi:outwardBindingSite multi:component="sti_B_Site1"
      multi:bindingStatus="unbound" />
  </multi:listOfOutwardBindingSites>
</species>

<!-- species AB: unphosphorylated -->
<species id="sp_AB" name="AB" multi:speciesType="st_AB" hasOnlySubstanceUnits="false"
  boundaryCondition="false" constant="false">
  <multi:listOfSpeciesFeatures>
    <multi:speciesFeature multi:speciesFeatureType="phosphorylation">
      <multi:speciesFeatureValue multi:value="U" />
    </multi:speciesFeature>
  </multi:listOfSpeciesFeatures>
</species>
...
</listOfSpecies>

<!-- reactions -->
<listOfReactions>

  <!-- Unphosphorylated Site1 of A binds to Site1 of B -->
  <!-- Kappa Rule: A(Site1~u),B(Site1) -> A(Site1~u!1),B(Site1!1) -->
  <reaction id="rc_AB" reversible="false" fast="false">
    <listOfReactants>
      <speciesReference species="sp_A" constant="false" />
      <speciesReference species="sp_B" constant="false" />
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="sp_AB" constant="false" />
    </listOfProducts>
    <kineticLaw>
      ...
    </kineticLaw>
  </reaction>
  ...
</listOfReactions>
</model>
</sbml>

```

## A Validation of SBML documents using *Multi* constructs

---

To be added later.

1

2



---

# Acknowledgments

---

1

This work was supported by the Intramural Research Program of the US National Institute of Allergy and Infectious Diseases of the National Institutes of Health.

2

3

## References

- Angermann, B. R., Klauschen, F., Garcia, A. D., Prustel, T., Zhang, F., Germain, R. N., and Meier-Schellersheim, M. (2012). Computational modeling of cellular signaling processes embedded into dynamic spatial contexts. *Nat Methods*, 9(3):283–9.
- Barik, D., Baumann, W. T., Paul, M. R., Novak, B., and Tyson, J. J. (2010). A model of yeast cell-cycle regulation based on multisite phosphorylation. *Molecular Systems*, 6:1–18.
- Danos, V. and Laneve, C. (2004). Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110.
- Faeder, J. R., Blinov, M. L., and Hlavacek, W. S. (2009). Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol Biol.*, 500:113–67.
- Faeder, J. R., Blinov, M. L., Hlavacek, W. S., Posner, R. G., Hucka, M., and Fontana, W. (2006). Rules for Modeling Signal-Transduction Systems. *Sci. STKE*, 344:ref6.
- Feret, J., Danos, V., Krivine, J., Harmer, R., and Fontana, W. (2009). Internal coarse-graining of molecular systems. *PNAS*, 106:6453–6458.
- Hucka, M., Bergmann, F., Hoops, S., Keating, S., Sahle, S., Schaff, J., Smith, L., and Wilkinson, D. (2010). The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core. Available via the World Wide Web at <http://sbml.org/Documents/Specifications>.
- Malleshaiah, M. K., Shahrezaei, V., Swain, P. S., and Michnick, S. W. (2010). The scaffold protein Ste5 directly controls a switch-like mating decision in yeast. *Nature*, 465:101–105.
- Meier-Schellersheim, M., Xu, X., Angermann, B., Kunkel, E. J., Jin, T., and Germain, R. N. (2006). Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method. *PLoS Comput Biol*, 2(7):e82.
- Mu, F., Williams, R. F., Unkefer, C. J. U. P. J., Faeder, J. R., and Hlavacek, W. S. (2007). Carbon-fate maps for metabolic reactions. *Bioinformatics*, 23:3193–3199.
- Novère, N. L. and Oellrich, A. (2010). Multistate and Multicomponent Species (multi). Available via the World Wide Web at [http://sbml.org/Community/Wiki/SBML\\_Level\\_3\\_Proposals/Multistate\\_and\\_Multicomponent\\_Species\\_Proposal](http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Multistate_and_Multicomponent_Species_Proposal).
- Zhang, F., Angermann, B. R., and Meier-Schellersheim, M. (2012). Draft for Discussion: SBML Proposals for "Revised Multi", "Simple Spatial" and "Multi-Spatial" Extensions. In *COMBINE 2012* (<http://goo.gl/hWV52F>).
- Zhang, F., Angermann, B. R., and Meier-Schellersheim, M. (2013). The Simmune Modeler visual interface for creating signaling networks based on bi-molecular interactions. *Bioinformatics*, 29 (9):1229–1230.
- Zhang, F. and Meier-Schellersheim, M. (2013a). Multistate, Multicomponent and Multicompartment Species Package for SBML Level 3 (Multi), rev 280. Available via the World Wide Web at <http://goo.gl/2375K>.
- Zhang, F. and Meier-Schellersheim, M. (2013b). Multistate, Multicomponent and Multicompartment Species Package for SBML Level 3 (SBML-Multi). In *HARMONY 2013* (<http://goo.gl/DIyEy>).