Multistate and Multicomponent Species (*multi*)

Editors:

Anika Oellrich   *EMBL European Bioinformatics Institute, UK*
Nicolas Le Novère   *EMBL European Bioinformatics Institute, UK*

Contributors:

Michael Blinov
Department of Genetics
Developmental Biology
University of Connecticut Health Center
263 Farmington Avenue, Farmington
CT 06030-3301, USA

James Faeder
3082 Biomedical Science Tower 3
University of Pittsburgh School of Medicine
Pittsburgh, PA 15260, USA

Andrew Finney
Oxford computer consultants
Oxford, UK

William S. Hlavacek
Mail Stop K710, Drop Point 03041003U
Los Alamos National Laboratory
Los Alamos, New Mexico 87545 , USA

Stefan Hoops
Virginia Bioinformatics Institute
0477 Virginia Tech Bioinformatics Facility I
Blacksburg
Va 24061, USA06030-3301, USA

Bin Hu
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545, USA

Nicolas Le Novère
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge CB10 1SD, UK

Martin Meier-Schellersheim
NIAID, NIH
Building 10, Room 11N311
10 Center Drive MSC 1892
Bethesda, MD 20892-1892, USA

Anika Oellrich
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge CB10 1SD, UK

Nicolas Rodriguez
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge CB10 1SD, UK

Date: November 29, 2010

Disclaimer: This is a working draft of the SBML Level 3 *multi* package It is not a normative document.

To discuss any aspect of SBML multi, please send your messages to the mailing list sbml-multi@lists.sf.net.

summary

Contents

Chapter 1

Introduction

The current document defines the package *multi* Version 1 of SBML Level 3 Version 1.

1.1 Graphical and typographical conventions

We use the following typographical conventions to distinguish objects and data types from other entities:

Class: Names of classes begin with a capital letter and are printed in a bold, sans-serif typeface.

attribute: Names of attributes begin with a lowercase letter and are printed in a bold, italic, sans-serif typeface.

value: CDATA (character data (see http://en.wikipedia.org/wiki/CDATA)), that is the textual content of an element or the value of an attribute, is printed in a italic, sans-serif typeface.

`code`: Examples of XML code are printed in monotype typeface.

Some **SpeciesTypes** can represent binding sites. Those binding sites can be linked to other binding sites. In this document, a binding site can be represented in four different contexts:

A binding site declared as explicitly unbound is represented with a white background.

A binding site declared as either bound or unbound is represented with a grey background.

A binding site declared as bound to another undefined binding site is represented with a black background.

A binding site bound declared as bound to a defined binding site is represented with a black background and a black edge linking it to the other binding site.

Throughout the text, the American spelling is used rather than the British one.
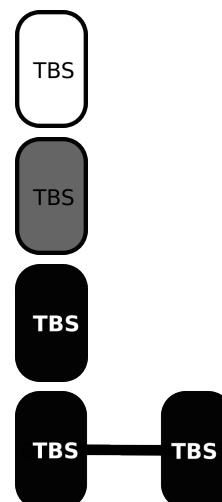
1.2 Motivation

This package *multi* addresses two different — though related — problems, commonly encountered when trying to model biological processes:

4

- The representation of entities that can exist under different states affecting their behaviours (multi-state entities). Those entities carry state features, sometimes many of them, each able to take different values. This may result in a combinatorial explosion of alternative states taken by the entities.

- The creation and behaviour of complexes made up of different components (multi-component entities). The rules of assembly may lead to an unbounded list of species, with the number of components and their topology impossible to precise before the simulation.

As a simple example of multi-state multi-component entity, let's consider a ligand-gated ion channel with only one feature, the *pore*, that can adopt three different values, *closed*, *opened* and *desensitized*. In addition to this state feature, the channel can be bound to a scaffold through an *anchor*.
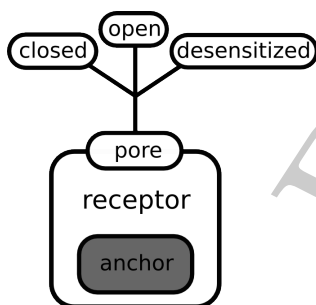
Figure 1.1: *Example of a molecular entity carrying a multi-valued variable and containing another entity.*

Taking into account the different values for the state feature, plus the status of the anchoring site (bound or not), this receptor can exist under six different states:

free, closed
free, open
free, desensitized
anchored, closed
anchored, open
anchored, desensitized

With the SBML core, any reaction involving the receptor, such as binding to a ligand, will have to be written six times. However, some of the state features and/or bonds may not affect the binding of the ligand, but the reactions have to be enumerated nevertheless, if we want to keep track of all the populations. Writing all the possibilities can be in the best case just exhausting, and in the worst case plainly impossible due to the combinatorial explosion. If an entity possesses 4 bivalued features and 2 trivalued features, the number of possible state is 24x32. A dodecamer of CaMKII with 5 different characteristics taking two values (e.g. activity, binding to calmodulin and to ATP, phosphorylations on threonin 286 and 306) exhibits 60 state features, and consequently a billion of billion possible states. Writing such a model by enumerating all possible states and reactions is plainly impossible, and one needs a way to describe only the relevant states of this species rather than all the possible ones.

Another problem addressed by the package *multi* is the unbounded list of multi-component entities. Let's imagine a situation where we would like to model the growth of microtubules from dimers of tubulin. We cannot possibly enumerate all the possible microtubules of different lengths. Furthermore, the length of a microtubule does not affect the rate with which a new

dimer of tubulin is incorporated. The only thing we need to encode is the binding between a tubulin dimer incorporated in a microtubule and a free tubulin dimer.
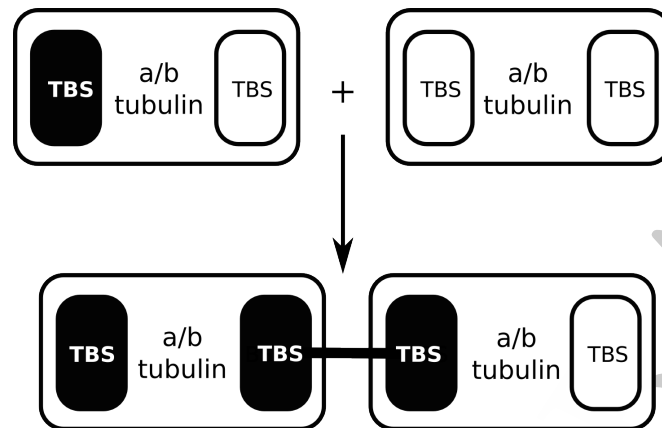
Figure 1.2: *Example of a reaction involving entities with undefined binding status.*

In the schema above, the grey binding site is either bound or unbound, and it can be bound to another tubulin dimer or a microtubule containing 100 tubulin dimers.

To handle the problems described above a field of modeling was developed, called rule-based modeling [**?**]. The main idea of rule-based modeling is to write down the rules that reactions must obey rather than the reactions themselves. And example of language used to describe rule-based models in biology is BioNetGen [**?**]. Another approach to avoid the combinatorial explosion of possible cases, is to use multi-agent modeling, where one represent all the interacting entities individually rather than pools. It the number of possible cases exceeds the number of entities, this method, otherwise verbose, become parsimonious. Example of multi-agent software used in biology are StochSim [**? ?**] and Simmune [**? ?**].

The graphical equivalent of the package *multi* Version 1 of SBML Level 3 Version 1 is the SBGN Entity Relationship language.

Finally, the package *multi* permit to encode entities that are made of components belonging to different compartments. Those entities cannot be species, since a species is attached to a single compartment, and the compartments in SBML Level 3 Version 1 are not overlapping. In *multi*, species types, representing species attached to different compartments, can be linked.

1.3 Past work on this problem or similar topics

Proposals for supporting multistates and multicomponent species have a long history in SBML. Here is a reconstruction in chronological order:

- 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 sligthly 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 Novère 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 develope the Level 3 package multi.

Chapter 2

Syntax and semantics

The proposal for extending SBML to carry the information for multistate multicomponent species relies on the extension of the following core SBML elements: **Species**, **InitialAssignment**, **Rule**, **Reaction**, **SimpleSpeciesReference**, **EventAssignment**. In addition, the multistate multi-component package requires the creation of new main elements: **SpeciesType**, **Selector**.

A software that does not understand *multi* Version 1 can ignore entirely all elements in the corresponding namespace. The remaining SBML model is a valid SBML Level 3 Version 1 core.

In the UML diagrams contained in this specification, the elements and attributes specific to the package *multi* are drawn in red, while the elements and attributes belonging to SBML core are drawn in black. Not all of the latter are drawn, but only the ones needed to "plug" the package *multi*. Please consult the specification of the core for a deeper understanding of SBML at large.

2.1 Extension of the model element

In order to encode the structures needed to define and use multistate and multi-component complexes, the element **model** is extended to be linked to a list of **SpeciesType**s and a list of **Selector**s.
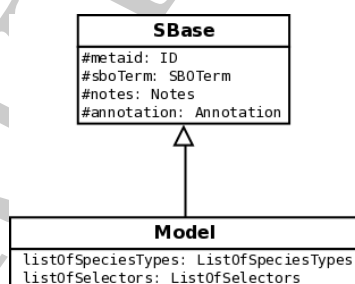
Figure 2.1: *Definition of **Model** and its relation with **SBase**.*

```
<model xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <!-- some compartments -->
  <multi:listOfSpeciesTypes>
    <!-- some species types -->
  </multi:listOfSpeciesTypes>
  <multi:listOfSelectors>
    <!-- some selectors -->
  </multi:listOfSelectors>
```

```
   <!-- some species, initialAssignments, rules,  reactions, events -->
</model>
```

2.2 Definition of the species types and their state features

In order to build multi-component multistate entities, one needs to define the building blocks that will be combined. A given type of component, a **SpeciesType**, can carry several **StateFeature**s, which are multi-valued characteristics of the component. One can also specify if instances of a **SpeciesType** can act as binding sites. Apart from a unique identifier, a name, and several annotations, it is furthermore possible to define state features of a component, and the values taken by those state features. The following UML diagram shows how the structure of the **SpeciesType** looks like.
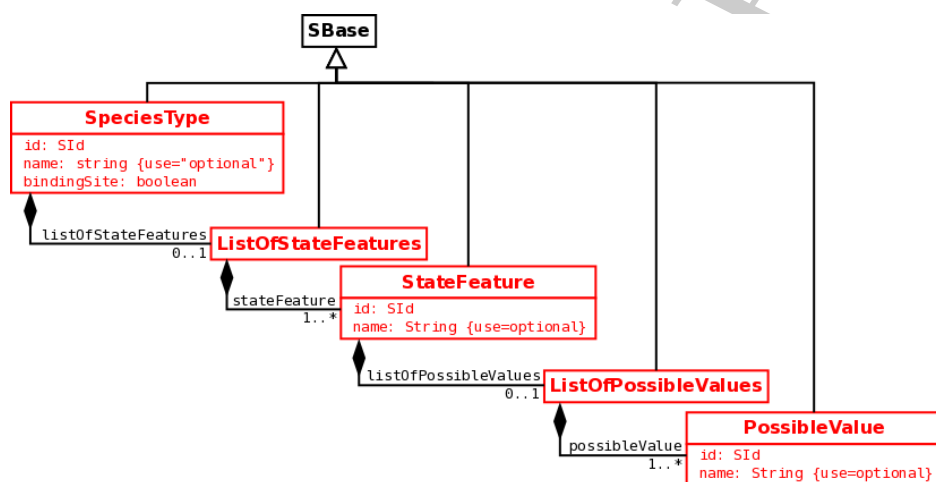
Figure 2.2: *SpeciesType and all the associated classes.*

2.2.1 SpeciesType

The element **SpeciesType**, which is part of SBML Level 2 Version 4 specification, is not part of SBML Level 3 Version 1 any more. Instead, it will be defined in the *multi* package. The **SpeciesType** element carries not only the basic attributes which it had in SBML Level 2 Version 4 (**metaid**, **id**, **name**), but is also extended for the needs of describing multi-component entities with the attribute **bindingSite** and for the needs of multistate entities by linking it to a list of **StateFeature**s.

9
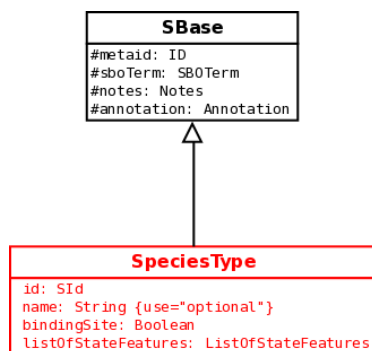
Figure 2.3: *Definition of **SpeciesType** and its relation with **SBase**.*

A species type can be used to describe a component of a supra-macromolecular assembly, but also a domain of a macromolecule. Such a domain can be a portion of the macromolecule, a non-connex set of atoms forming a functional domain, or just a conceptual construct suiting the needs of the modeler. The type of component can be specified by referring terms from the subbranch *functional entity* of the Systems Biology Ontology (http://biomodels.net/sbo/, [**?**]) through the optional **sboTerm** attribute. The following table provides typical examples of component or domains (the list is absolutely not complete).

SBO identifier	definition
SBO:0000242	channel
SBO:0000244	receptor
SBO:0000284	transporter
SBO:0000280	ligand
SBO:0000493	functional domain
SBO:0000494	binding site
SBO:0000495	catalytic site
SBO:0000496	transmembrane domain

The example below encodes a species type representing a ligand-gated ion channel, that is a transmembrane macromolecule that can form a ionic channel stabilised by the binding of ligands. Notice the notes and annotation that belong to the namespace of SBML Level 3 Version 1 *core*.

```
<multi:speciesType xmlns:core="http://www.sbml.org/sbml/level3/version1"
                   xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                   xmlns:xhtml="http://www.w3.org/1999/xhtml"
                   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                   xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
                   multi:id="speciesType1"
                   multi:name="LGIC"
                   multi:bindingSite="false"
                   multi:sboTerm="SBO:0000242">
  <core:notes>
    <xhtml:body>
      <xhtml:p>LGIC is a Ligand-Gated Ion Channel</xhtml:p>
    </xhtml:body>
  </core:notes>
  <core:annotation>
    <rdf:RDF>
```

```
      <rdf:Description rdf:about="#_000003">
        <bqbiol:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="urn:miriam:interpro:IPR002394"/>
          </rdf:Bag>
        </bqbiol:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </core:annotation>
  <multi:listOfStateFeatures>
    <!-- some state features -->
  </multi:listOfStateFeatures>
</multi:speciesType>
```

The example below encodes a species type representing the binding site for a ligand, that could be used for instance in conjunction with the previous example.

```
<multi:speciesType xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                   multi:id="speciesType2"
                   multi:name="AgonistSite"
                   multi:bindingSite="true"
                   multi:sboTerm="SBO:0000494" />
```

2.2.2 StateFeature

A species type can carry any number of state features, which are characteristic properties specific for this type of species. The element **StateFeature** of SBML Level 3 Version 1 *multi* Version 1 corresponds to the "state variable" of the SBGN Entity Relationship language. A **StateFeature** is identified by an **id** and an optional **name**. A **StateFeature** is linked to a list of **PossibleValue**s.
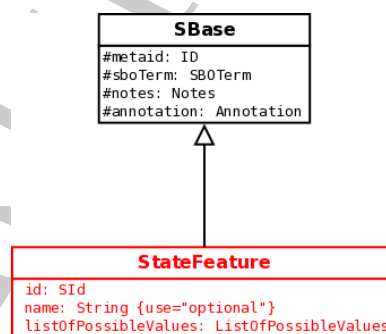
Figure 2.4: *Definition of **StateFeatureClass** and its relation with **SBase**.*

As all elements derived from **SBase**, **StateFeature** can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. The value *SBO:0000497* of **sboTerm** used in the example below corresponds to a "ternary switch".

```
<multi:stateFeature
            xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
```

```
                multi:id="stateFeature1"
                multi:name="pore"
                multi:sboTerm="SBO:0000497">
  <multi:listOfPossibleValues>
    <!-- some possible values -->
  </multi:listOfPossibleValues>
</multi:stateFeature>
```

2.2.3 PossibleValue

Each state feature also requires the definition of all the possible values it can take. Those values will be used within a selector, to define the states an entity is allowed to take. A state feature is not obligatory a boolean property, but can carry any number of **PossibleValue**. A **PossibleValue** is identified by an **id** and an optional **name**.
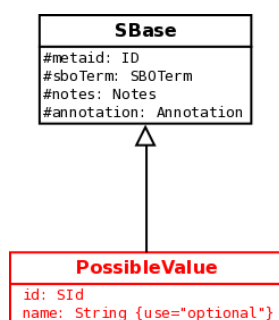
Figure 2.5: *Definition of **PossibleValue** and its relation with **SBase**.*

As all elements derived from **SBase**, **PossibleValue** can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. The **sboTerm** used in the example below corresponds to a "ternary switch".

```
<multi:possibleValue
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="possibleValue1"
                multi:name="open"
                multi:sboTerm="0000416" />
```

2.2.4 Complete definition of a species type

The following example describes the SpeciesType presented on Figure 1.1 on page 5.

```
<multi:speciesType xmlns:core="http://www.sbml.org/sbml/level3/version1"
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                xmlns:xhtml="http://www.w3.org/1999/xhtml"
                xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
                multi:id="speciesType1"
                multi:bindingSite="false"
                multi:name="LGIC"
```

```
                        multi:sboTerm="SBO:0000242">
  <core:notes>
    <xhtml:body>
      <xhtml:p>
        LGIC is a Ligand-Gated Ion Channel. It contains a pore that can be open or closed.
      </xhtml:p>
    </xhtml:body>
  </core:notes>
  <core:annotation>
    <rdf:RDF>
      <rdf:Description rdf:about="#_000003">
        <bqbiol:isVersionOf>
          <rdf:Bag>
            <rdf:li rdf:resource="urn:miriam:interpro:IPR002394"/>
          </rdf:Bag>
        </bqbiol:isVersionOf>
      </rdf:Description>
    </rdf:RDF>
  </core:annotation>
  <multi:listOfStateFeatures>
    <multi:stateFeature multi:id="stateFeature1"
                        multi:name="pore"
                        multi:sboTerm="SBO:0000497">
      <multi:listOfPossibleValues>
        <multi:possibleValue id="possibleValue1"
                             multi:name="open"
                             multi:sboTerm="SBO:0000416" />
        <multi:possibleValue id="possibleValue2"
                             multi:name="closed"
                             multi:sboTerm="SBO:0000417" />
        <multi:possibleValue id="possibleValue3"
                             multi:name="desensitised"
                             multi:sboTerm="SBO:0000417" />
      </multi:listOfPossibleValues>
    </multi:stateFeature>
  </multi:listOfStateFeatures>
</multi:speciesType>
```

2.3 Definition of filters to select states and topologies

A selector is a mask describing the rules that an entity has to pass in order to be used or rejected. This selector is built of different components, carrying state features. The components can be bound together or not. In a population-based model, the selector, when applied to a pool of entities, permits to filter it, and to obtain a further, more refined, entity pool.
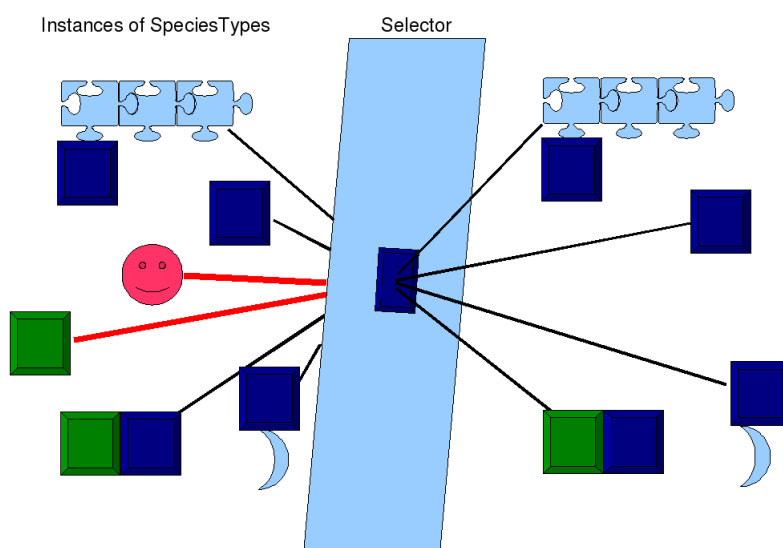
Figure 2.6: *The above selector accepts everything containing a blue crystal, but rejects the red face and the isolated green crystal.*

A selector can be reused in various places of a model, to restrict the application of a procedure to a certain set of topologies and states. Selectors can be used to refine the initial conditions of a species, for instance to specify the initial distribution of different states and topologies. They can also be used in a reaction to decide if a this reaction happens, or to modulate its velocity, in function of the state or topology of a reactant.

A selector defines the list of components composing the mask, that are species type existing under a given state (that can be an ensemble of elementary states). In addition to the components, the selector lists the possible or mandatory bonds, as well as the components that must not be bound. It is to be noted that a selector must not necessarily be the most parsimonious. One can use the selectors to describe the fine-grained topology of complexes, even if this topology is not used to decide upon particular reactions. The general structure of the selector is provided in Figure 2.7 on the next page and the general structure of the species type state is provided in Figure 2.8 on the following page.
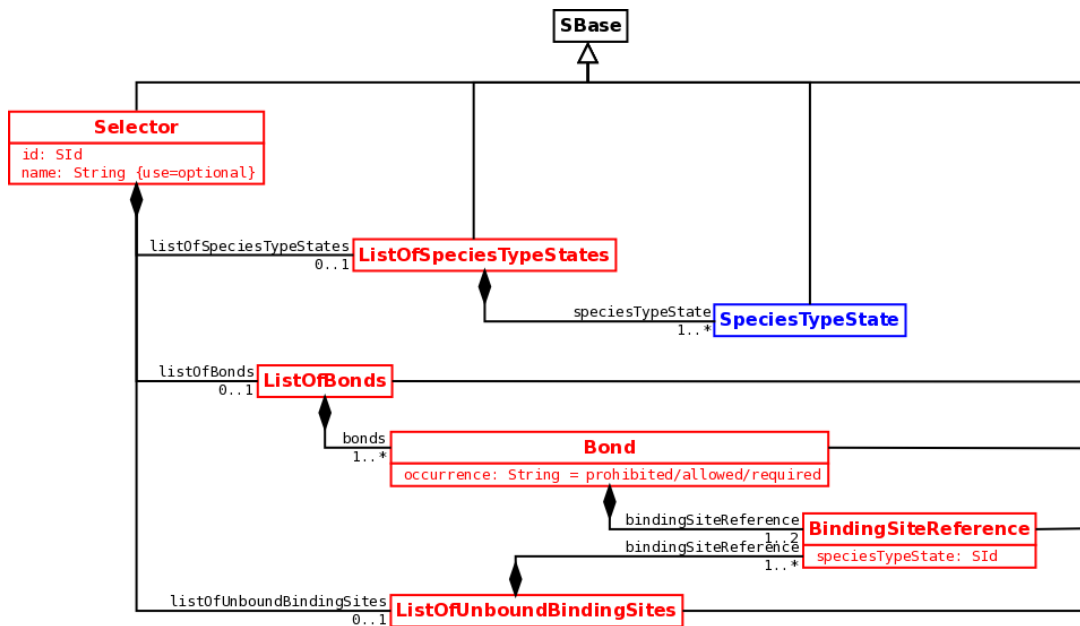
Figure 2.7: *Selector* *and all the associated classes.*
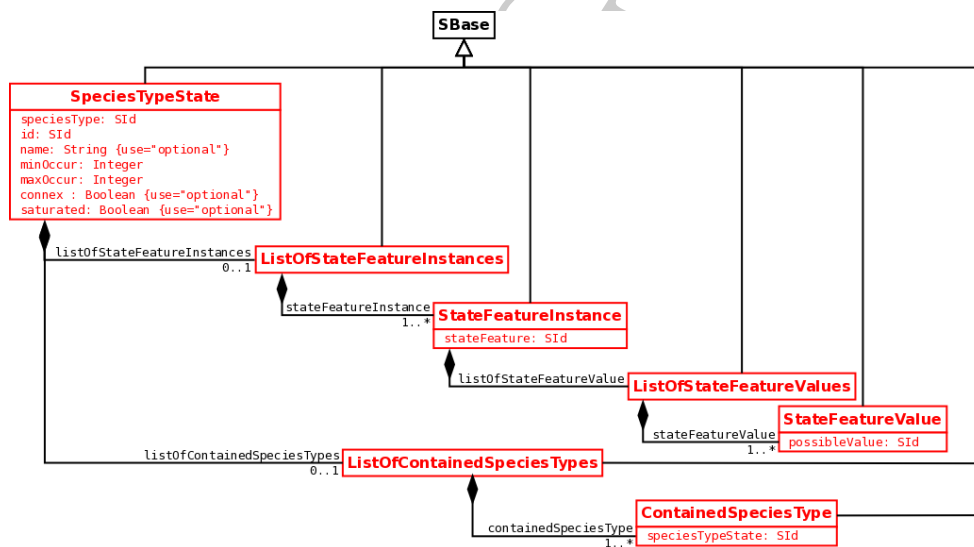
Figure 2.8: *SpeciesTypeState* *and all the associated classes.*

A graphical representation of how to build a selector to encode a complex entity is represented on Figure 2.9 on the next page.
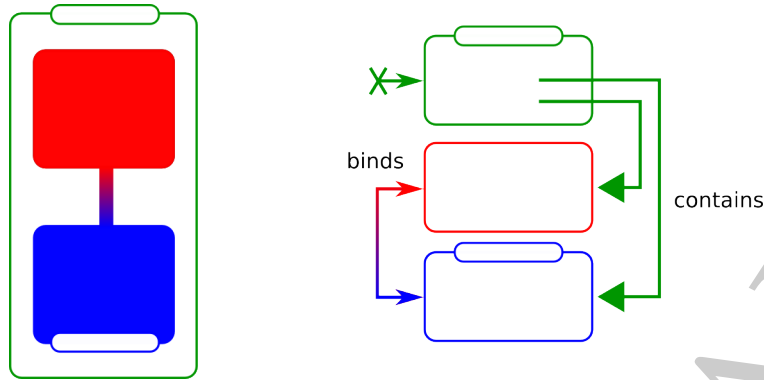
Figure 2.9: *This figure illustrates the procedure by which a selector builds the representation of a multi-state, multi-component complex. To encode the left part, representing the complex following the conventions used in this document, we write part represented on the right, listing the different components and their relationships.*

The **id** of **SpeciesTypeState** belong to a namespace local to the containing selector. Therefore, one can use the same values for **id** of **SpeciesTypeState** in different selectors. However, within a selector, all the values for **id** of **SpeciesTypeState** must be different.

2.3.1 Selector

A **Selector** is identified by an **id** and an optional **name**. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. In addition, a **Selector** is linked to a list of **SpeciesTypeState**s, a list of **Bond**s and a list of **BindingSiteReference**s.
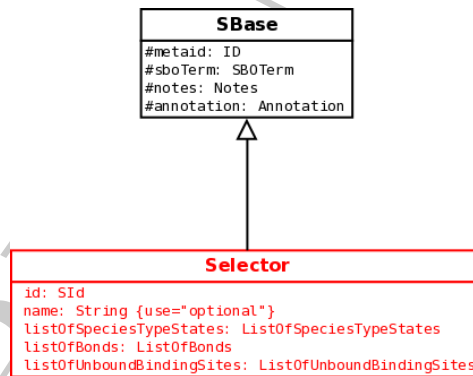
Figure 2.10: *Definition of **Selector** and its relation with **SBase**.*

```
<multi:selector
               xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
               multi:id="selector1"
               multi:name="unbound_receptor">
  <multi:listOfSpeciesTypeStates>
    <!-- some species type state -->
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds>
    <!-- some bonds -->
  </multi:listOfBonds>
```

```
    <multi:listOfUnboundBindingSites>
      <!-- some unbound binding sites -->
    </multi:listOfUnboundBindingSites>
</multi:selector>
```

2.3.2 SpeciesTypeState

A species type state describes an ensemble of instances of a species type can be into, in order
to fulfill the requirements of the selector. In order to build complex multi-component species,
an instance of a species type can contain other instances of species types, that have to be
declared in the selector also, with their allowed states. A species type state is then defined
by the values of the different state features carried by the species type, the list of species type
states it "contains", and their topology. A **SpeciesTypeState** is identified by an **id** and an
optional **name** plus an attribute **speciesType**, pointing to the **SpeciesType** it instantiates. As
all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a **metaid**, and
an **sboTerm**. In addition, a **SpeciesTypeState** can be linked to a list of **StateFeatureInstance**s
and a list of **ContainedSpeciesType**s. A **SpeciesTypeState** can be instantiated several times
in a selector, using the attributes **minOccur** and **maxOccur**. The attribute **connex** precises
that all those instances must be part of a continuous network through bonds formed by the
ContainedSpeciesTypes. The attribute **saturated** precises that all **ContainedSpeciesType**s that
are instances of **SpeciesType** with their attribute **bindingSite** set to true must be involved in
bonds.

Figure 2.11: *Definition of **SpeciesTypeState** and its relation with **SBase**.*

```
<multi:speciesTypeState
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="speciesTypeState1" multi:name="open_receptor"
                multi:speciesType="speciesType1"
                multi:minOccur="2" multi:maxOccur="4"
                multi:connex="true" multi:saturated="true" >
  <multi:listOfStateFeatureInstances>
    <!-- some state feature instances -->
  </multi:listOfStateFeatureInstances>
  <multi:listOfContainedSpeciesTypes>
    <!-- some contained species types -->
  </multi:listOfContainedSpeciesTypes>
</multi:speciesTypeState>
```

17

2.3.3 StateFeatureInstance

The possible states of an instance of species type are described using the state features of that species type. Only the meaningful state features, that are used to specified the state, must be listed. The other are assumed to take any value, to be wildcards. A **StateFeatureInstance** is identified by an attribute *StateFeature*, pointing to the **StateFeature** it instantiates. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. In addition, a **StateFeatureInstance** can be linked to a list of **StateFeatureValue**s.

```
                         ┌─────────────────────────────────┐
                         │              SBase               │
                         ├─────────────────────────────────┤
                         │ #metaid: ID                      │
                         │ #sboTerm: SBOTerm                │
                         │ #notes: Notes                    │
                         │ #annotation: Annotation          │
                         └─────────────────────────────────┘
                                         △
                                         │
                         ┌─────────────────────────────────────────────┐
                         │           StateFeatureInstance              │
                         ├─────────────────────────────────────────────┤
                         │ id: SId                                      │
                         │ name: String {use="optional"}               │
                         │ stateFeature: SId                           │
                         │ listOfStateFeatureValues: ListOfStateFeatureValues │
                         └─────────────────────────────────────────────┘
```
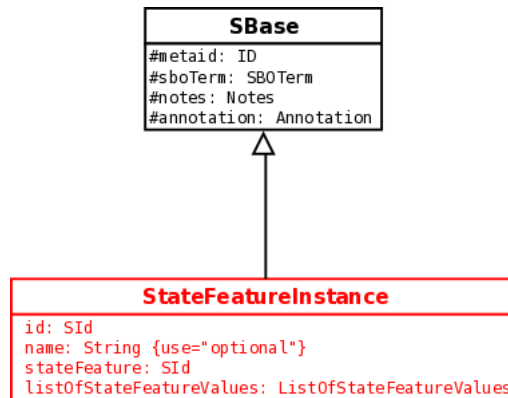
Figure 2.12: *Definition of **StateFeatureInstance** and its relation with **SBase**.*

```
<multi:stateFeatureInstance
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:stateFeature="stateFeature1">
  <!-- some state feature values -->
</multi:stateFeatureInstance>
```

2.3.4 StateFeatureValue

The selector specifies the values that a state feature of an instance of species type can take. A **StateFeatureValue** points to the relevant **PossibleValue**s defined in the instantiated **SpeciesTypes** using an attribute *possibleValue*. As all elements derived from **SBase**, **StateFeatureValue** can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*.

```
                    ┌─────────────────────────────────┐
                    │              SBase               │
                    ├─────────────────────────────────┤
                    │ #metaid: ID                      │
                    │ #sboTerm: SBOTerm                │
                    │ #notes: Notes                    │
                    │ #annotation: Annotation          │
                    └─────────────────────────────────┘
                                    △
                                    │
                    ┌─────────────────────────────────┐
                    │        StateFeatureValue         │
                    ├─────────────────────────────────┤
                    │ possibleValue: SId               │
                    └─────────────────────────────────┘
```

Figure 2.13: *Definition of **StateFeatureValue** and its relation with **SBase**.*

```
<multi:stateFeatureValue
                 xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                 multi:possibleValue="possibleValue1" />
```

2.3.5 ContainedSpeciesType

In order to build complex nested multi-component species, an instance of a species type can contain other instances of species types, that have to be declared in the same selector. As all elements derived from **SBase**, **ContainedSpeciesType** can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. In addition, a **ContainedSpeciesType** points to the relevant **SpeciesTypeState** using an attribute *speciesTypeState*.
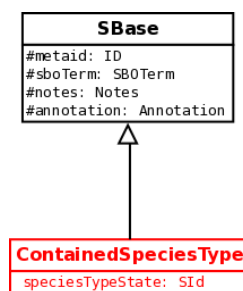
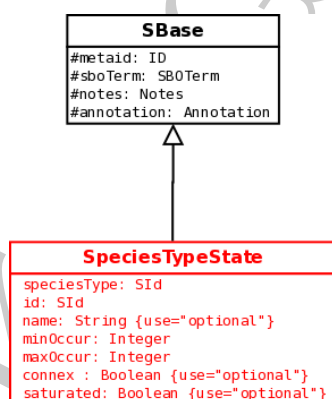**Figure 2.14:** *Definition of **ContainedSpeciesType** and its relation with **SBase**.*

```
<multi:containedSpeciesType
            xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
            multi:speciesTypeState="speciesType1" />
```

2.3.6 Bond

The connectivity between the components of a selector is described by listing the bonds that are possible, mandatory or forbidden. As all elements derived from **SBase**, **Bond** can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. An attribute *occurrence* specifies if a bond is *required*, *allowed* or *prohibited*. In addition, a **Bond** can be linked to a one or two **BindingReference**s.
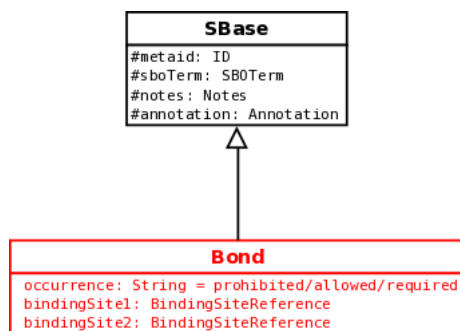
Figure 2.15: *Definition of **Bond** and its relation with **SBase**.*

```
<multi:bond xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
            multi:occurrence="required" />
  <!-- some binding site references -->
</multi:bond>
```

2.3.7 BindingSiteReference

A component involved in a bond is specified by a **BindingSiteReference**. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. In addition, a **BindingSiteReference** refers to the component involved in the bond through the attribute **speciesTypeState**, that points to the value of the **id** attribute of the relevant **SpeciesTypeState**.
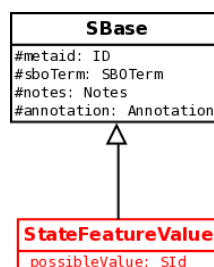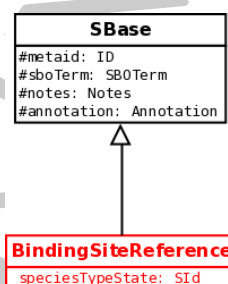

Figure 2.16: *Definition of **BindingSiteReference** and its relation with **SBase**.*

```
<multi:bindingSiteReference
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:speciesTypeState="speciesTypeState1" />
```

2.3.8 Defining the components and the states allowed by the selector

The following code contains a portion of selector describing two species type states, **speciesTypeState1** and **speciesTypeState2**. **SpeciesTypeState1** has a feature **stateFeature1** that can take the values **possibleValue1** or **possibleValue2** to pass the selection. In addition **speciesTypeState2** contains 4 instances of **speciesType2** (more exactly, it can contain between 4 and 4 instances

of *speciesTypeState2* ...). The boolean attributes **connex** and **saturated** are of no use in this particular example, but are to be interpreted in conjunction with the bond definitions.

```xml
<multi:listOfSpeciesTypeStates
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:speciesTypeState multi:id="speciesTypeState1" multi:speciesType="speciesType1"
                          multi:minOccur="4" multi:maxOccur="4"
                          multi:connex="true" multi:saturated="true">
    <multi:listOfStateFeatureInstances>
      <multi:stateFeatureInstance multi:stateFeature="stateFeature1">
        <multi:listOfStateFeatureValues>
          <multi:stateFeatureValue multi:possibleValue="possibleValue1" />
          <multi:stateFeatureValue multi:possibleValue="possibleValue2" />
        </multi:listOfStateFeatureValues>
      </multi:stateFeatureInstance>
    </multi:listOfStateFeatureInstances>
  </multi:speciesTypeState>
  <multi:speciesTypeState multi:id="speciesTypeState2" multi:speciesType="speciesType2"
                          multi:minOccur="1" multi:maxOccur="1">
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeState1" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
</multi:listOfSpeciesTypeStates>
```


Figure 2.17: *Generation of a nested selector from the definition of the two species type states, and the containment of one by the other.*

It is up to the model designer to avoid impossible structures, such as a species type state 1 that contains a species type 2, 2 itself containing 1. The following example is forbidden.

21

```
<multi:listOfSpeciesTypeStates
                  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:speciesTypeState multi:id="speciesTypeState1" multi:speciesType="speciesType1"
                          multi:minOccur="1" multi:maxOccur="1" >
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeState2" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
  <multi:speciesTypeState multi:id="speciesTypeState2" multi:speciesType="speciesType2"
                          multi:minOccur="1" multi:maxOccur="1" >
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeState1" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
</multi:listOfSpeciesTypeStates>
```

If a species type is used in two different contexts, requiring for instance different numbers of occurrences, two different species type states must be defined. For instance, if a species type A is used on its own in a species type C, or as contained twice in another species type B itself in C, we need one A with min/maxOccur=1 and one A with min/maxOccur=2.

```
<multi:listOfSpeciesTypeStates xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:speciesTypeState multi:id="speciesTypeStateA1" multi:speciesType="speciesTypeA"
                          multi:minOccur="1" multi:maxOccur="1" />
  <multi:speciesTypeState multi:id="speciesTypeStateA2" multi:speciesType="speciesTypeA"
                          multi:minOccur="2" multi:maxOccur="2" />
  <multi:speciesTypeState multi:id="speciesTypeStateB" multi:speciesType="speciesTypeB"
                          multi:minOccur="1" multi:maxOccur="1">
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateA2" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
  <multi:speciesTypeState multi:id="speciesTypeStateC" multi:speciesType="speciesTypeC"
                          multi:minOccur="1" multi:maxOccur="1" >
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateA1" />
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateB" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
</multi:listOfSpeciesTypeStates>
```

Figure 2.18: *Nested selector where a given species type is used in two different contexts.*

The same species type can also be declared several time with different state feature values. The following example depicts a species type B containing two instances of species type A with a state feature displaying alternative values. Although in both case, the species type states refer to the same species type, we need to declare them separately.

```xml
<multi:listOfSpeciesTypeStates
                  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:speciesTypeState multi:id="speciesTypeStateA1" multi:speciesType="speciesTypeA"
                      multi:minOccur="1" multi:maxOccur="1"
    <multi:listOfStateFeatureInstances>
      <multi:stateFeatureInstance multi:stateFeature="stateFeature1">
        <multi:listOfStateFeatureValues>
          <multi:stateFeatureValue multi:possibleValue="possibleValue1" />
        </multi:listOfStateFeatureValues>
      </multi:stateFeatureInstance>
    </multi:listOfStateFeatureInstances>
  </multi:speciesTypeState>
  <multi:speciesTypeState multi:id="speciesTypeStateA2" multi:speciesType="speciesTypeA"
                      multi:minOccur="1" multi:maxOccur="1"
    <multi:listOfStateFeatureInstances>
      <multi:stateFeatureInstance multi:stateFeature="stateFeature1">
        <multi:listOfStateFeatureValues>
          <multi:stateFeatureValue multi:possibleValue="possibleValue2" />
        </multi:listOfStateFeatureValues>
      </multi:stateFeatureInstance>
    </multi:listOfStateFeatureInstances>
  </multi:speciesTypeState>
  <multi:speciesTypeState multi:id="speciesTypeStateB" multi:speciesType="speciesTypeB"
                      multi:minOccur="1" multi:maxOccur="1">
    <multi:listOfContainedSpeciesTypes>
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateA1" />
      <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateA2" />
    </multi:listOfContainedSpeciesTypes>
  </multi:speciesTypeState>
</multi:listOfSpeciesTypeStates>
```
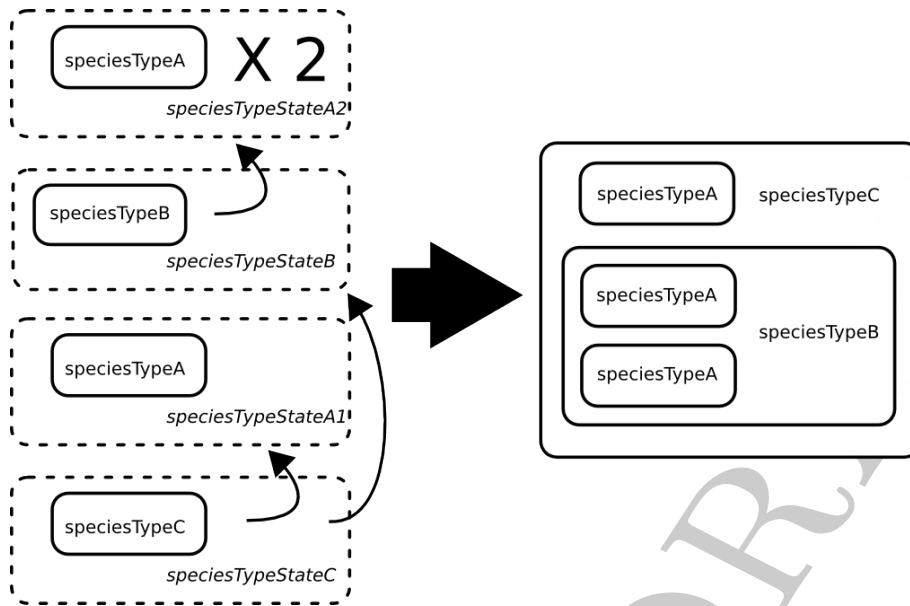
Figure 2.19: *Nested selector where a given species type is used with two different state feature values.*

2.3.9 Defining the connectivities allowed by the selector

In a selector, instances of species types possessing an attribute **bindingSite** set to *true* can be referred to when they are involved in bonds, or explicitly listed as unbound. As a result such a species type can be used in four different contexts:

- In a declared explicit bond, with another specific instance of a species type (represented by a bond linking two black entities in the example graphs).

```
<multi:bond xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
            multi:occurrence="required" >
  <multi:bindingSiteReference multi:speciesTypeState="speciesTypeState1" />
  <multi:bindingSiteReference multi:speciesTypeState="speciesTypeState2" />
</multi:bond>
```

- In a declared generic bond without another specific instance of a species type (represented by an unlinked black entity in the example graphs).

```
<multi:bond xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
            multi:occurrence="required" >
  <multi:bindingSiteReference multi:speciesTypeState="speciesTypeState1" />
</multi:bond>
```

- Declared as explicitly unbound (represented by a white entity in the example graphs).

```
<multi:listOfUnboundBindingSites
            xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" />
  <multi:bindingSiteReference multi:speciesTypeState="speciesTypeState1" />
</multi:listOfUnboundBindingSites>
```

24

- Undeclared, meaning that one does not care if it is bound or not (represented by a gray entity in the example graphs).

A given binding site can only be bound to another binding site. If three instances of species types are bound together, they have to be bound through different contained binding sites. Those binding sites can be different instances of the same species type ("identical" binding sites) or instances of different species types (different binding sites). The following example is forbidden:

```xml
<multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:bond multi:occurrence="required" >
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA" />
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateB" />
  </multi:bond>
  <multi:bond multi:occurrence="required" >
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA" />
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
  </multi:bond>
</multi:listOfBonds>
```

Instead the following code should be used, where *speciesTypeStateA1* and *speciesTypeStateA2* are two different species type states of the same species type.

```xml
<multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <multi:bond multi:occurrence="required" >
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA1" />
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateB" />
  </multi:bond>
  <multi:bond multi:occurrence="required" >
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA2" />
    <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
  </multi:bond>
</multi:listOfBonds>
```


Figure 2.20: *The connectivity on the left is forbidden, and the connectivity on the right should be used instead.*

The description of the connectivity through the list of bonds and the list of unbound binding sites is greedy. In other words, the possible connectivity is the sum of all the different listed possibilities. If in a selector, an instance of a species type is involved in at least one allowed bond, then it must be bound to something. It cannot be unbound. Furthermore, if an instance of a species type is involved in listed bonds, only the listed bonds can fulfill the selection.

The attribute **occurrence**, in combination with explicit, generic and undeclared bonds, provides for a complete boolean logic to select the context of bindingSites. Note that **occurrence** is only present on **Bond**. In order to forbid a binding site to be unbound, we create a generic bond with **occurrence** set to *required* (to say that X cannot be free is equivalent to say that X had to be bound to something, whatever it is).

To exemplify this boolean logic, let's take the example of an instance of a species type X interacting with instances of species type A, B or C.

In the following code, X is not declared at all, therefore X can be bound to A, B, C or be unbound:

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateA" speciesType="speciesTypeA"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateB" speciesType="speciesTypeB"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
</multi:selector>
```

With the following code, X can only be bound to A, or unbound:

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateA" speciesType="speciesTypeA"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" >
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```
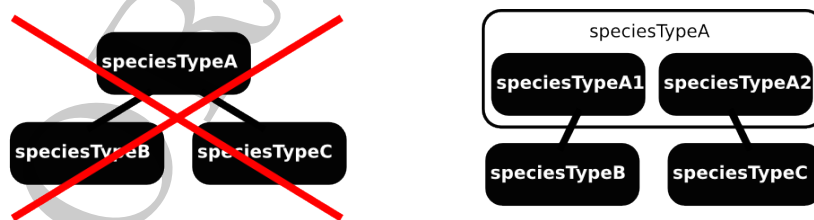
With the following code X can only be bound either to A or B, but not C:

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateA" speciesType="speciesTypeA"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateB" speciesType="speciesTypeB"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA" />
```

```
      </multi:bond>
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateB" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The following code defines a generic bond with X, which mean that X can be bound to A, B or C, but has to be bound to something:

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
               multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                           multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" >
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The following code says that X cannot be bound to C (The bond X-C is prohibited), and therefore has to be bound to A or B or unbound.

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
               multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                           multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateC" speciesType="speciesTypeC"
                           multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
    <multi:bond multi:occurrence="prohibited" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The following code says that X cannot be bound to C (The bond X-C is prohibited), however a generic bond specifies that X has to bound to something, therefore effectively to A or B. In this particular example, **occurrence** on the generic bond can be set indifferently to *allowed* or *required*.

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
               multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
```

```
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateC" speciesType="speciesTypeC"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" >
    <multi:bond multi:occurrence="prohibited" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
    </multi:bond>
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The following case is overdetermined, since the impossibility for X to bind C is implicit in the restriction of its binding to A or B.

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateX" speciesType="speciesTypeX"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateA" speciesType="speciesTypeA"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateB" speciesType="speciesTypeB"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateC" speciesType="speciesTypeC"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateA" />
    </multi:bond>
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateB" />
    </multi:bond>
    <multi:bond multi:occurrence="prohibited" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateX" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The boolean attributes **connex** and **saturated** precise the topology of a multimeric assembly of the same component (same species type state, that is same species type with the same values allowed for all their state features). One can then declare the component only once, and each of the bond types or unbound binding sites only once. Attribute **saturated** set to *true* means that all the contained species types which **bindingSite** is set to *true* have to be bound. Attribute **connex** means that all the instances of the species type state have to be connected within the same complex. The following example presents a tetramer of identical subunits.

```
<multi:selector xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
```

```
                   multi:id="selector1" />
  <multi:listOfSpeciesTypeStates>
    <multi:speciesTypeState id="speciesTypeStateA" speciesType="speciesTypeA"
                            multi:minOccur="4" multi:maxOccur="4"
                            connex="true|false" saturated="true|false">
      <multi:listOfContainedSpeciesTypes>
        <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateB" />
        <multi:containedSpeciesType multi:speciesTypeState="speciesTypeStateC" />
      </multi:listOfContainedSpeciesTypes>
    </multi:speciesTypeState>
    <multi:speciesTypeState id="speciesTypeStateB" speciesType="speciesTypeB"
                            multi:minOccur="1" multi:maxOccur="1" />
    <multi:speciesTypeState id="speciesTypeStateC" speciesType="speciesTypeC"
                            multi:minOccur="1" multi:maxOccur="1" />
  </multi:listOfSpeciesTypeStates>
  <multi:listOfBonds xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
    <multi:bond multi:occurrence="allowed" >
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateB" />
      <multi:bindingSiteReference multi:speciesTypeState="speciesTypeStateC" />
    </multi:bond>
  </multi:listOfBonds>
</multi:selector>
```

The following graphs are examples of the various combinations possible of **connex** and **saturated** attributes.
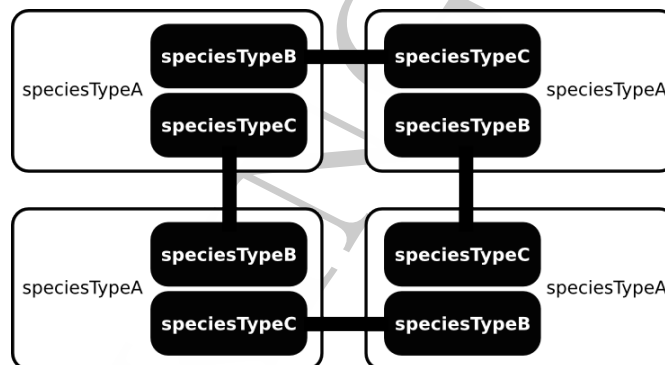
Figure 2.21: `connex="true" saturated="true"`
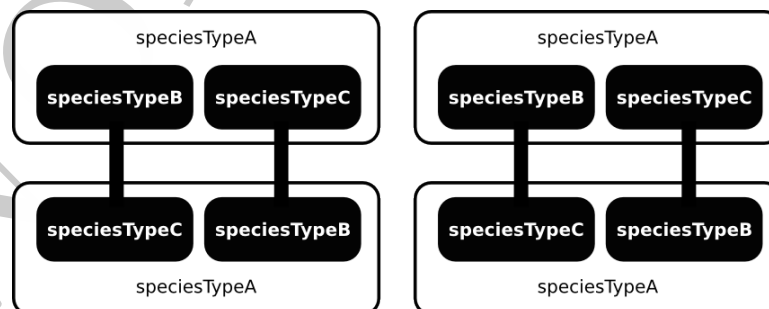
Figure 2.22: `connex="false" saturated="true"`

29
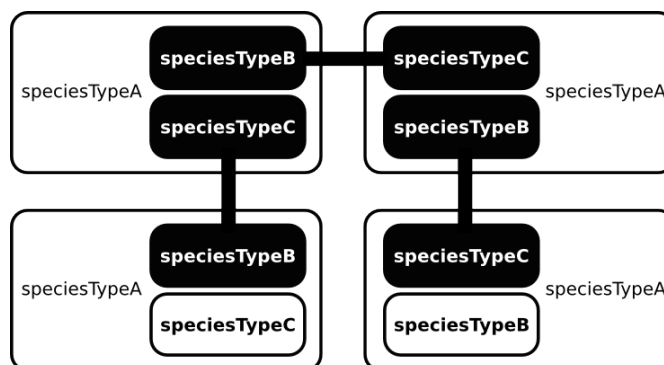
Figure 2.23: `connex="true" saturated="false"`
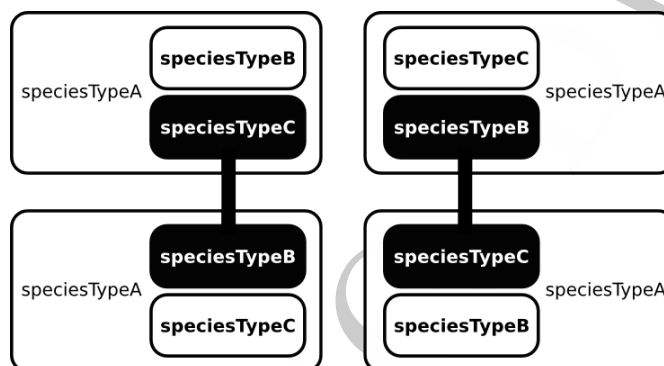
Figure 2.24: `connex="false" saturated="false"`

2.4 Definition of the instances and pools of instances

Once the multistate and multi-component types of entities used by the model have been described, using **SpeciesType**s and **Selector**s, one needs to define quantitatively actual states and connectivity of the elementary components within the **Species** of the SBML core. The new entity pool instances will then be used either as initial conditions or in any SBML constructs refering to **Species**. In order to specify these states and partnerships, we extend **Species** to link to **SpeciesTypeInstance**s, themselves refering to a list of **Selector**s. A **SpeciesTypeInstance** describes an entity that fulfils all the listed **Selector**s. A **Species** is effectively the ensemble of all the **SpeciesTypeInstance**s belonging to the same compartment. Note that it is the responsibility of the person or software generating the SBML file to ensure that the selectors used to precise the **SpeciesTypeInstance**s are not incompatible. The initial conditions, that is the intial states and connectivities of each species have to be fully specified.

Note that a "filtered" species is also an entity pool, and therefore is located in a given compartment. However, this species subset can be generated, selected, with a **Selector** that is using several SpeciesTypes instantiated by species in different compartments. Another species, located in another compartment, could be selected by a different "portion" of the same selector. This provides a mechanism to build multi-compartment entities.

2.4.1 Species

In order to encode the structures needed to refine entity pools based on their state and connectivity, we precise instances of which **SpeciesType** compose the **Species**. The element **Species** is
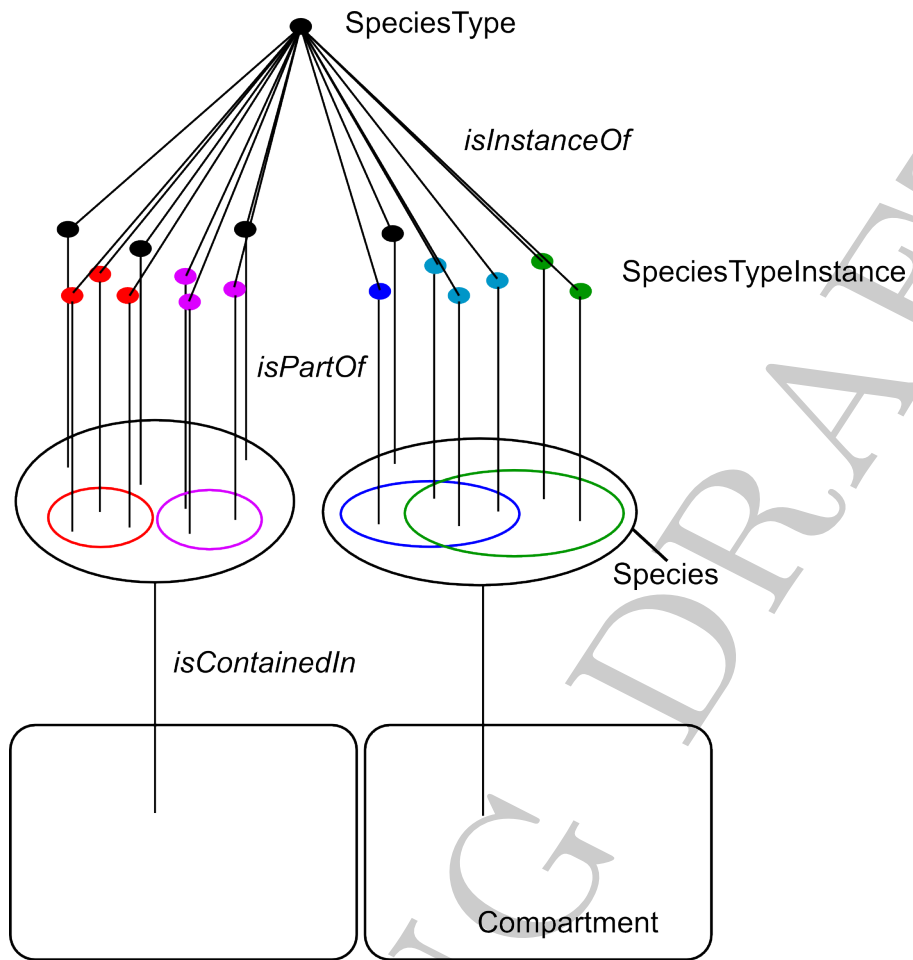
Figure 2.25: *Relationships between a conceptual species type, the actual species type instances and the different pools of entities defined in a model description.*
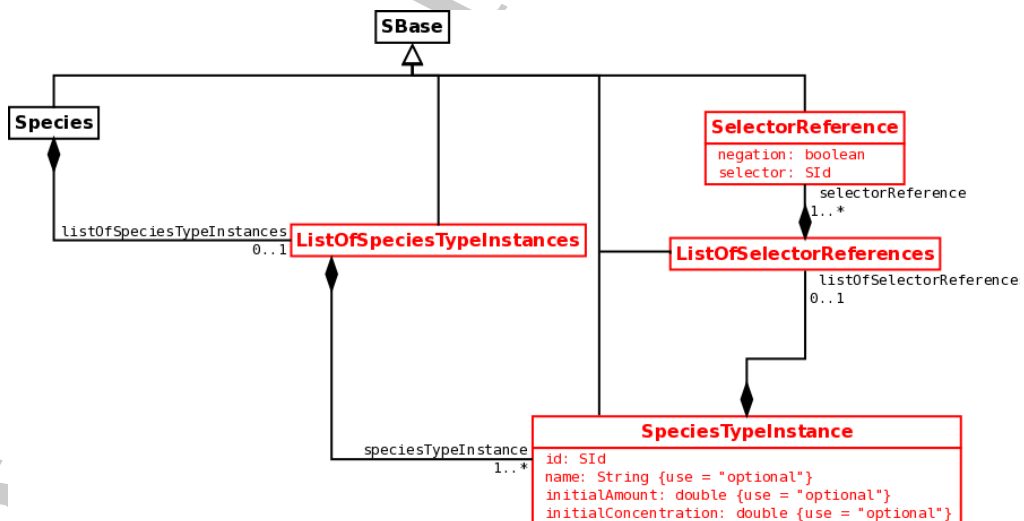
Figure 2.26: *Species and all the associated classes of* multi *Version 1.*

then linked to a list of **SpeciesTypeInstance**s, each of one describing an entity part of a different pool.
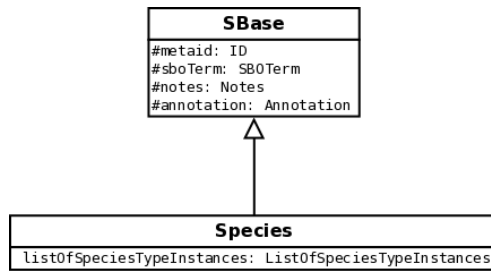
Figure 2.27: *Definition of the extended version of **Species** and its relation with **SBase**.*

```
<species id="species1" name="LGIC" sboTerm="SBO:0000245"
        boundaryCondition="false" hasOnlySubstanceUnit="false" constant="false"
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
        multi:speciesType="speciesType1" >
  <multi:listOfSpeciesTypeInstances>
    <!-- some definition of subpools -->
  </multi:listOfSpeciesTypeInstances>
</species>
```

2.4.2 SpeciesTypeInstance

A **SpeciesTypeInstance** is identified by an *id* and an optional *name*. A **SpeciesTypeInstance** is linked to a list of **SelectorReference**s. When used in the SBML model, the *id* represent the either the size of the pool formed by all the **SpeciesTypeInstance**s that fulfil the selection, or the species type restriction. The context decides. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. In addition, a **SpeciesTypeInstance** may precise the size of the pool as an *initialAmount* or an *initialConcentration*. If neither *initialAmount* nor *initialConcentration* are precised, the instance is not to be used for initial conditions.
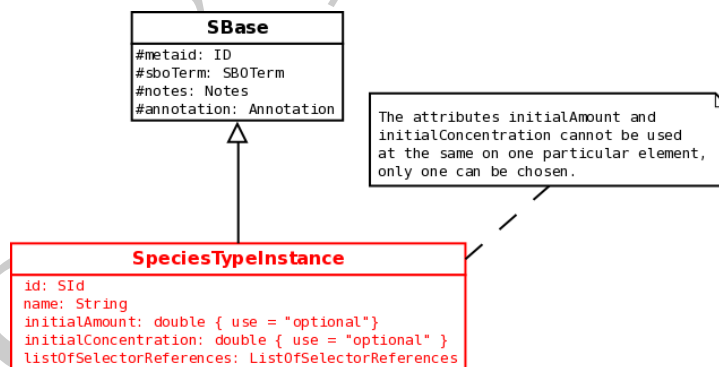
Figure 2.28: *Definition of **SpeciesTypeInstance** and its relation with **SBase**.*

```
<multi:speciesTypeInstance
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
        multi:id="SpeciesTypeInstance1"
        multi:name="LGIC_open"
```

```
        multi:initialAmount="1">
  <multi:listOfSelectorReferences>
    <!-- selectors to combine to filter out the instance -->
  </multi:listOfSelectorReferences>
</multi:speciesTypeInstance>
```

Note that because an instance can fulfill several selectors, the selectors used to create instances used as initial conditions may overlap. As a consequence, the sum of all the initial quantities may be larger than the initial quantity specified on the **Species**. It is up to the modeler generating the description to make sure there is no ambiguity and that whatever procedure is used to create the instances result in the same distribution.

SpeciesTypeInstances defined in a **Species** inherit the value of the attribute **hasOnlySubstanceUnit** carried by the **Species**. In other words, regardless of the presence of the attributes **initialAmount** and **initialConcentration** carried by a **SpeciesTypeInstance**, when used in a context requiring a quantity its **id** represents an amount if the **species**' **hasOnlySubstanceUnit** is set to *true*.

2.4.3 SelectorReference

As all elements derived from **SBase**, A **SelectorReference** can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. It targets a **Selector** through its **selector** attribute. A boolean attribute **negation** allows to precise that the entities fulfilling the rules described in the selector are NOT selected.

```
                    ┌─────────────────────────────┐
                    │          SBase              │
                    ├─────────────────────────────┤
                    │ #metaid: ID                 │
                    │ #sboTerm: SBOTerm           │
                    │ #notes: Notes               │
                    │ #annotation: Annotation     │
                    └─────────────────────────────┘
                                 △
                                 │
                    ┌─────────────────────────────┐
                    │     SelectorReference       │
                    ├─────────────────────────────┤
                    │ negation: boolean           │
                    │ selector: SId               │
                    └─────────────────────────────┘
```
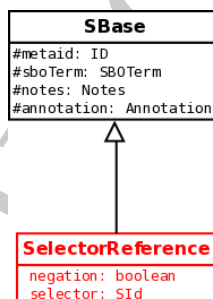
Figure 2.29: *Definition of* **SelectorReference** *and its relation with* **SBase**.

```
<multi:selectorReference
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
        multi:selector="selector1"
        multi:negation="true" />
```

2.4.4 Complete description of a species with pools of instances

The example below describes the encoding of two instances of the speciesType *speciesType1*.

```
<species id="species1"
        boundaryCondition=false" hasOnlySubstanceUnit=false" constant=false"
```

33

```
        compartment="compartment1" initialAmount="1000"
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
        multi:speciesType="speciesType1" >
  <multi:listOfSpeciesTypeInstances>
    <multi:SpeciesTypeInstance multi:id="speciesTypeInstance1"
                             multi:initialAmount="1">
      <multi:listOfSelectorReferences>
        <multi:selectorReference multi:selector="selector1" multi:negation="false"/>
      <multi:listOfSelectorReferences>
    </multi:speciesTypeInstance>
    <multi:speciesTypeInstance multi:id="speciesTypeInstance2">
      <multi:listOfSelectorReferences>
        <multi:selectorReference multi:selector="selector2" multi:negation="false"/>
        <multi:selectorReference multi:selector="selector3" multi:negation="true" />
      <multi:listOfSelectorReferences>
    </multi:speciesTypeInstance>
  </multi:listOfSpeciesTypeInstances>
</species>
```


Figure 2.30: *Procedure to build speciesTypeInstances from a speciesType and selectors. The dashed rectangle represent all the instances of a species, in all possible states and connectivity. Selector1 is used to generate the entity pool SpeciesTypeInstance1. Selector2 and Selector3 are used in combination to generate SpeciesTypeInstance2.*

2.5 Computing initial values for pools of instances

In SBML Level 3 Version 1, **InitialAssignment**s are used to assign initial values to variables of a model, that are **Compartment**s, **Species**, **SpeciesReference**, or global **Parameter**s. In order to assign the value of a specific type of species, **InitialAssignment**s in *multi* Version 1 also contain an element **SpeciesTypeInstanceChange**. The assignment sets the initial quantity of instances that fulfil a certain selection using the mathematical expression provided. The value provided by the **InitialAssignment** overides the value provided by the attributes *initialAmount* or *initialConcentration* of the relevant **SpeciesTypeInstance**.

2.5.1 InitialAssignment

In order to assign the initial values to entity subpools, defined by specific state and connectivity, the element **InitialAssignment** of SBML Level 3 Version 1 core is linked to a **SpeciesTypeInstanceChange**.

34
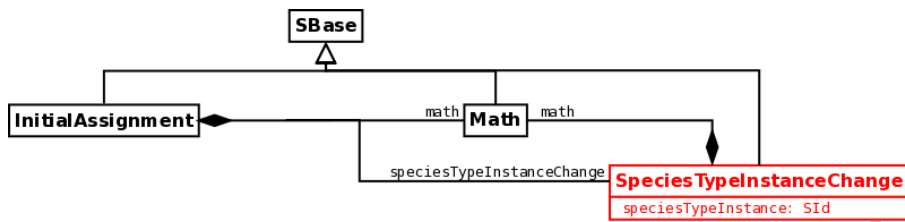
Figure 2.31: *InitialAssignment and all the associated classes of* multi *Version 1.*
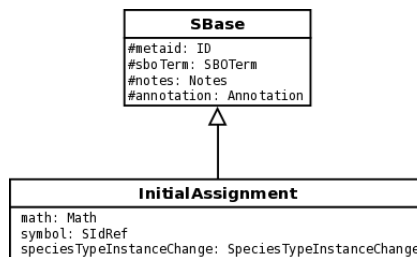
Figure 2.32: *Definition of the extended version of* **InitialAssignment** *and its relation with* **SBase**.

2.5.2 SpeciesTypeInstanceChange

As all elements derived from **SBase**, A **SpeciesTypeInstanceChange** can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. It targets a **SpeciesTypeInstance** through its *SpeciesTypeInstance* attribute. The change is computed using a MathML construct, as is always the case in SBML Level 3 Version 1.
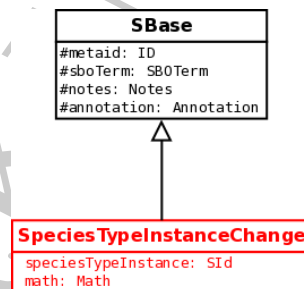
Figure 2.33: *Definition of* **SpeciesTypeInstanceChange** *and its relation with* **SBase**.

2.5.3 Complete example of an initial assignment

The following example presents the assignment of the initial value for the subpool of *species1* defined as *speciesTypeInstance1*. This value is computed as the product of the parameters *x* and *y*, defined elsewhere. Note the empty **Math**, used when the software is unable to use the package *multi* Version 1.

```
<species id="species1"
        boundaryCondition="false" hasOnlySubstanceUnit="false" constant="false"
        compartment="compartment1" initialAmount="1000"
```

```
          xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
          multi:speciesType="speciesType1" >
  <multi:listOfSpeciesTypeInstances>
    <multi:speciesTypeInstance multi:id="speciesTypeInstance1" multi:initialAmount="1">
      <multi:listOfSelectorReferences>
        <multi:selectorReference multi:selector="selector1" />
      <multi:listOfSelectorReferences>
    </multi:speciesTypeInstance>
  </multi:listOfSpeciesTypeInstances>
</species>
<initialAssignment symbol="species1">
  <multi:speciesTypeInstanceChange speciesTypeInstance="speciesTypeInstance1">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply>
        <times/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </math>
  </multi:speciesTypeInstanceChange>
  <math xmlns="http://www.w3.org/1998/Math/MathML" />
</initialAssignment>
```
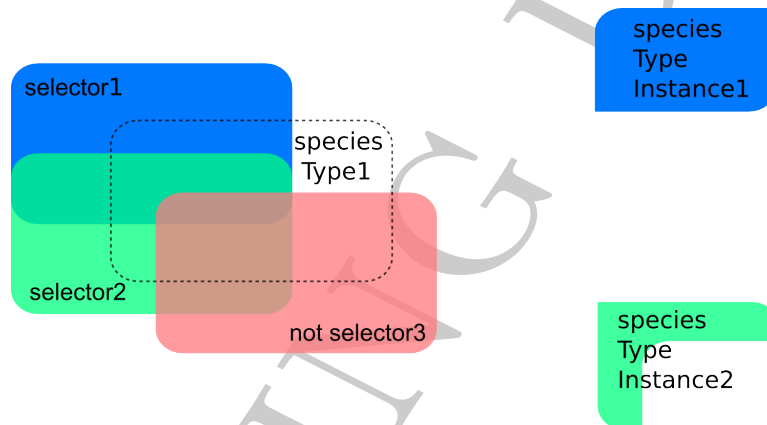
2.6   Rules

In SBML Level 3 Version 1, **AssignmentRule**s and **RateRule**s are used to assign values or change
of values respectively to variables of a model, that are **Compartment**s, **Species**, **SpeciesReference**,
or global **Parameter**s. In order to assign the value of a specific type of species, **AssignmentRule**s
and **RateRule**s in *multi* Version 1 also contain an element **SpeciesTypeInstanceChange**. The
assignment sets the quantity of instances that fulfil a certain selection using the mathematical
expression provided.
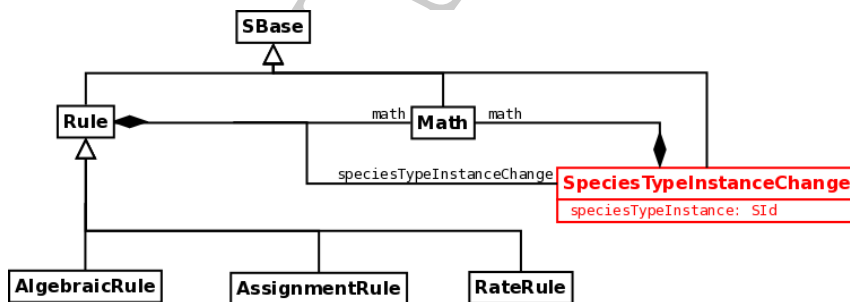
Figure 2.34: *Rule and all the associated classes of* multi *Version 1.*

2.6.1   Rule

In order to assign values, or value evolutions, to entity subpools, defined by specific state and
connectivity, the elements **AssignmentRule** and **RateRule** of SBML Level 3 Version 1 core are
linked to a **SpeciesTypeInstanceChange**.

36
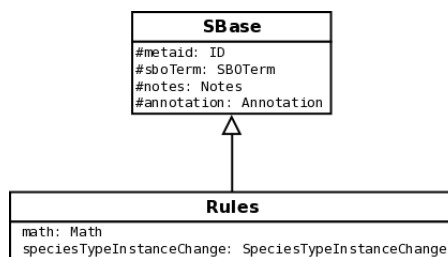
Figure 2.35: *Definition of the extended version of* **Rule** *and its relation with* **SBase**.

2.6.2 SpeciesTypeInstanceChange

For a definition of **SpeciesTypeInstanceChange**, see section 2.5.2.

2.7 Definition of reactions and reaction rules

Now that different instances of the species types have been defined using species and selectors, we can use them to choose between alternative conditional reactions, and to set-up the characteristics of the entities resulting from those conditional. This is done by creating, for each relevant reaction, a list of reaction rules that contains alternative kineticLaws.
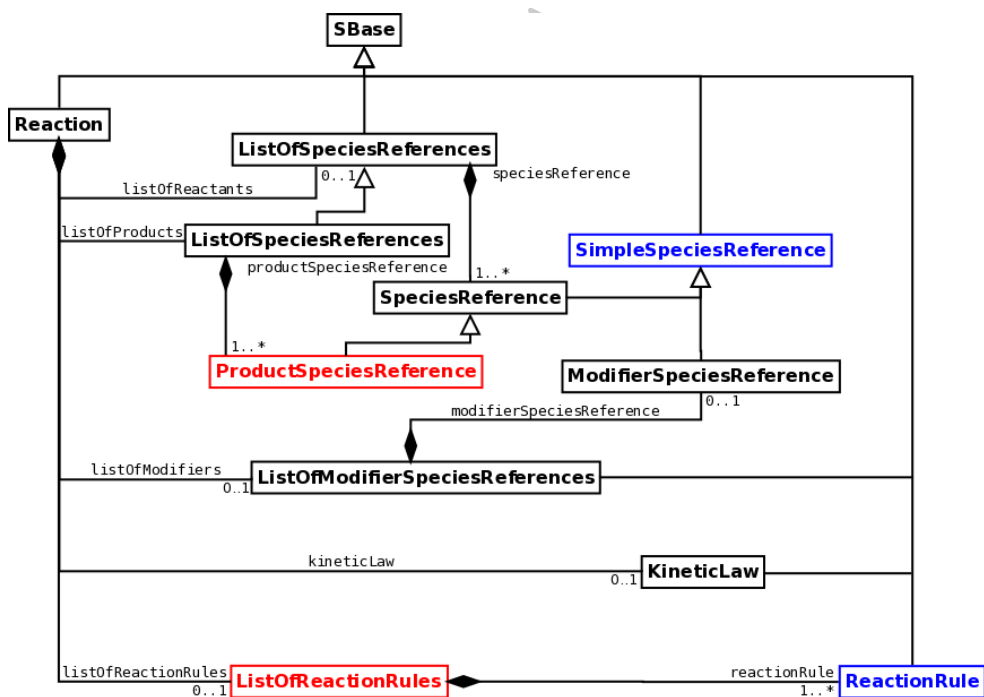
Figure 2.36: *Reaction* *and all the associated classes of* multi *Version 1.*

A reaction rule applies when a list of conditions is fulfiled by the reacting species, and a reaction rule produces outcomes described in a list of results. Reaction rules must not produce mass or cause unexplained disappearance of it. In other words, once the various selections are applied, what is on the left should be on the right. In order to ensure that, what is not explicitly represented must be left untouched by the reactions.

37
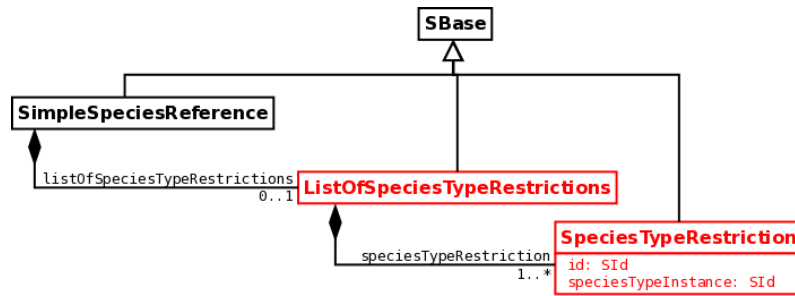
Figure 2.37: *SimpleSpeciesReference* *and all the associated classes of* multi *Version 1.*

$$* - A + B \rightarrow * - A - B$$

Can represent:

$$C - A + B \rightarrow C - A - B$$

or

$$D - A + B \rightarrow D - A - B$$

But not:

$$C - A + B \rightarrow D - A - B$$
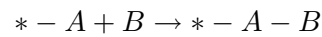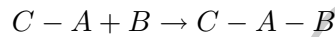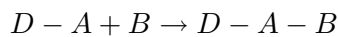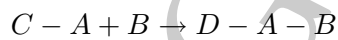
Such a complex reaction must be explicitly described if needed.
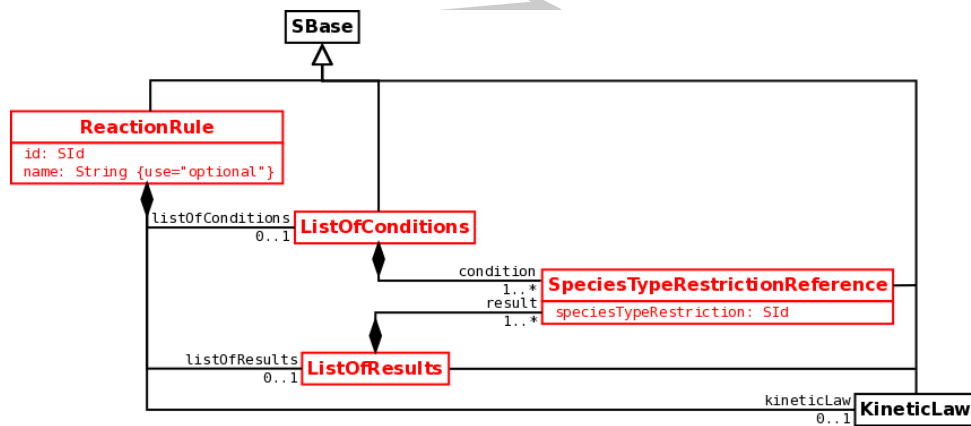
Figure 2.38: *ReactionRule* *and all the associated classes of* multi *Version 1.*

2.7.1 Reaction

In order to encode the structures needed to propose alternative kinetic laws selected based on the state and connectivity of involved partners, we extend the element **Reaction** of SBML Level 3 Version 1 core by linking it to a list of **ReactionRule**s.
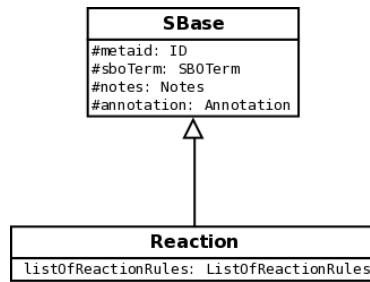
Figure 2.39: *Definition of the extended version of **Reaction** and its relation with **SBase**.*

```
<reaction id="react" reversible="false" fast="false"
          xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
  <listOfReactants>
    <!-- some reactants -->
  </listOfReactants>
  <listOfProducts>
    <!-- some products -->
  <kineticLaw />
  <multi:listOfReactionRules>
    <!-- some reaction rules-->
  </multi:listOfReactionRules>
</reaction>
```

2.7.2 SimpleSpeciesReference

In order to decide which kinetic law to choose, one needs to have a list of the necessary states and connectivities for the partners involved. This is done by linking the **SimpleSpeciesReference** of SBML Level 3 Version 1 core to a list of **SpeciesTypeRestriction**s. There can be any number of **SpeciesTypeRestriction**s per **SimpleSpeciesReference**. They always represent alternatives, and only one can be used per **ReactionRule**. However, if one wants to discriminate between two instances of the same **Species** involved in the same **ReactionRule** but with different roles in the reaction, one must create two **SimpleSpeciesReference**s pointing to the same **Species**.
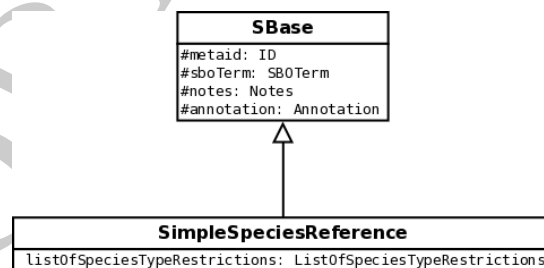
Figure 2.40: *Definition of the extended version of **SimpleSpeciesReference** and its relation with **SBase**.*

```
<speciesReference species="species1" stoichiometry="1"
                  xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1">
```

```
   <multi:listOfSpeciesTypeRestriction>
     <!-- some species type restrictions -->
   </multi:listOfSpeciesTypeRestriction>
</speciesReference>
```

2.7.3 ProductSpeciesReference

In the cases where several instances of the same molecule are use in a reaction, we must explicit the correspondence between the reactants in the product. Otherwise, the following reactions will be selected by the same reaction rule:

$$A1 - P + A2 \rightarrow A1 + A2 - P$$

$$A1 - P + A2 \rightarrow A1 - P + A2$$
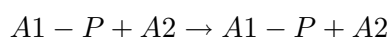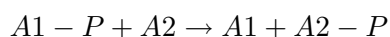
In order to do so, we create a new element **ProductSpeciesReference** that inherits from **SimpleSpeciesReference**. The element carries an attribute *correspondingReactant* that precises from which reactant instance the product originates. In an agent-based approach, this would apply to each molecule, while in a population-based framework, this would apply to pools.
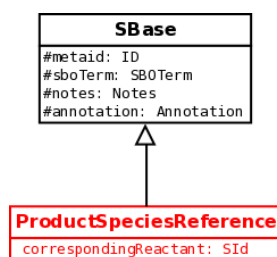
Figure 2.41: *Definition of the extended version of **ProductSpeciesReference** and its relation with **SBase**.*

```
<listOfReactants>
  <speciesReference id="R1" species="species1" stoichiometry="1"/>
  <speciesReference id="R2" species="species1" stoichiometry="1"/>
</listOfReactants>
<listOfProducts>
  <multi:productSpeciesReference species="species1" stoichiometry="1"
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:correspondingReactant="R1" />
  <multi:productSpeciesReference species="species1" stoichiometry="1"
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
                multi:correspondingReactant="R2" />
</listOfProducts>
```

2.7.4 SpeciesTypeRestriction

A **SpeciesTypeRestriction** points to a **SpeciesTypeInstances**, and creates a conditional species reference. Those various **SpeciesTypeRestriction**s can then be used to decide on the **ReactionRule** to use, and which **Result**s to produce. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*.
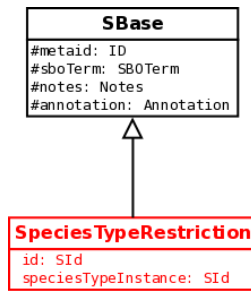
Figure 2.42: *Definition of **SpeciesTypeRestriction** and its relation with **SBase**.*

```
<multi:speciesTypeRestriction multi:id="speciesRest1"
                multi:speciesTypeInstance="speciesTypeInstance1"
                xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"/>
```

2.7.5 ReactionRule

The **ReactionRule** element is used to describe a process that is dependent on states or connectivity. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a *metaid*, and an *sboTerm*. A **ReactionRule** applies when the conditions described in a **ListOfConditions** are fulfilled. The **ReactionRule** replaces the the regular **Reaction**. The effect of a **ReactionRule** is described by a **ListOfResults**.
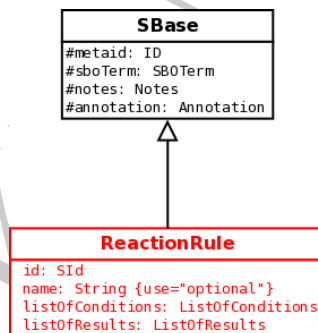
Figure 2.43: *Definition of **ReactionRule** and its relation with **SBase**.*

```
<multi:reactionRule multi:id="bindingNonPhospho"
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" >
  <multi:listOfConditions>
    <!-- some conditions for the rule to be used-->
  </multi:listOfConditions>
  <multi:listOfResults>
    <!-- some results of the application of the rule -->
  </multi:listOfResults>
  <kineticLaw />
</multi:reactionRule>
```

2.7.6 SpeciesTypeRestrictionReference

In order to precise the conditions for a reaction rule to apply, and describe the results to obtain, a **SpeciesTypeRestrictionReference** points to a species type restriction using the attribute **speciesTypeRestriction**. As all elements derived from **SBase**, it can link to **Notes** and **Annotation**, and carry a **metaid**, and an **sboTerm**. The **SpeciesTypeRestriction**s used to decide if a rule applies are called in **ListOfConditions**. The **SpeciesTypeRestriction**s used to decide if the result of a rule are called in **ListOfResults**.
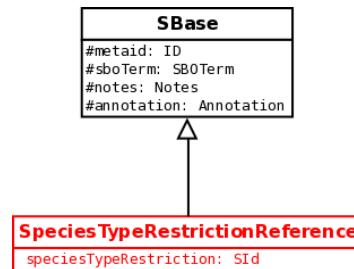
Figure 2.44: *Definition of **SpeciesTypeRestrictionReference** and its relation with **SBase**.*

```
<multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeRnonP"
        xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1" />
```

2.7.7 ReactionRule -specific KineticLaw

A **ReactionRule** optionally contains a **KineticLaw**. If the conditions apply, this **KineticLaw** is used to compute the results instead of the one contained in the reaction element of SBML Level 3 Version 1. Note that the semantic of the model stripped of all information belonging to *multi* Version 1 may be understood in very different ways if the reaction does not contains a kineticLaw or contains an empty one. In the former case, stripping the species restrictions could means that the reaction always occurs, while in the latter, the reaction would have a flux of 0, effectively never happening. In the kinetic law of a reaction rule, the quantities of species are represented by the **id** of the **SpeciesTypeRestriction**s, and not the **id** of the **Species**. This allows for multi-component reactions, where instances of the same species are playing different roles.

2.7.8 Complete example of a reaction with reaction rules

The following example describes a binding reaction that takes place ten times faster on the phosphorylated receptor than on the non-phosphorylated one.

```
<reaction id="receptLigBinding" reversible="false" fast="false">
  <listOfReactants>
    <speciesReference id="spRef_Rec" species="receptor" stoichiometry="1">
      <multi:listOfSpeciesRestriction>
        <multi:speciesRestriction multi:id="restriction1"
                              multi:speciesTypeInstance="receptorNP" />
        <multi:speciesRestriction multi:id="restriction2"
                              multi:speciesTypeInstance="receptorP" />
      </multi:listOfSpeciesRestriction>
```

```
        </speciesReference>
        <speciesReference species="ligand" stoichiometry="1" />
    </listOfReactants>
    <listOfProducts>
        <productSpeciesReference species="receptor" stoichiometry="1"
                        correspondingReactant="spRef_Rec">
            <multi:listOfSpeciesRestriction>
                <multi:speciesRestriction multi:id="restriction3"
                                        multi:speciesTypeInstance="receptorBound" />
            </multi:listOfSpeciesRestriction>
        </productSpeciesReference>
    </listOfProducts>
    <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML" />
    </kineticLaw>
    <multi:listOfReactionRules>
        <multi:reactionRule multi:id="reactionRule1">
            <multi:listOfConditions>
                <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="restriction1"/>
            </multi:listOfConditions>
            <multi:listOfResults>
                <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="restriction3"/>
            </multi:listOfResults>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML" >
                    <ci> parameter1 </ci>
                </math>
                <listOfLocalParameters>
                    <localParameter id="parameter1" value="1">
                </listOfLocalParameters>
            </kineticLaw>
        </multi:reactionRule>
        <multi:reactionRule multi:id="reactionRule2">
            <multi:listOfConditions>
                <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="restriction2"/>
            </multi:listOfConditions>
            <multi:listOfResults>
                <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="restriction3"/>
            </multi:listOfResults>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML" >
                    <ci> parameter2 </ci>
                </math>
                <listOfLocalParameters>
                    <localParameter id="parameter2" value="10">
                </listOfLocalParameters>
            </kineticLaw>
        </multi:reactionRule>
    </multi:listOfReactionRules>
</reaction>
```

2.8 Assignments following discrete events

In SBML Level 3 Version 1, **EventAssignment**s are used to assign values to variables of a model, that are **Compartment**s, **Species**, **SpeciesReference**, or global **Parameter**s when some conditions are fulfilled. In order to assign the value of a specific type of species, **EventAssignment**s in *multi* Version 1 also contain an element **SpeciesTypeInstanceChange**. The assignment sets the quantity of instances that fulfil a certain selection using the mathematical expression provided.
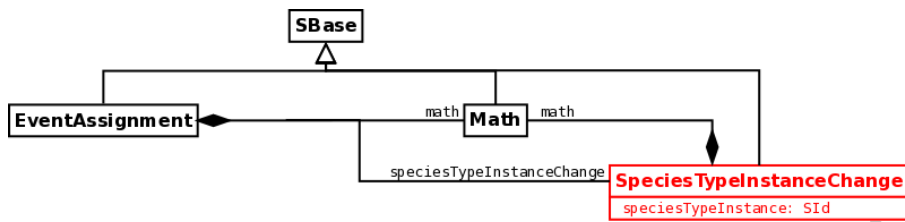
Figure 2.45: *EventAssignment and all the associated classes of* multi *Version 1.*

2.8.1 EventAssignment

In order to assign values to entity subpools, defined by specific state and connectivity, following a discrete event, the element **eventAssignment** of SBML Level 3 Version 1 core is linked to a **SpeciesTypeInstanceChange**.
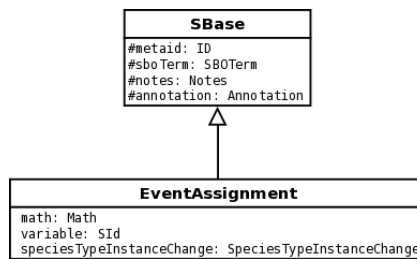
Figure 2.46: *Definition of the extended version of* **EventAssignment** *and its relation with* **SBase***.*

2.8.2 SpeciesTypeInstanceChange

For a definition of **SpeciesTypeInstanceChange**, see section 2.5.2.

44

Chapter 3

Examples

3.1 Receptor phosphorylation
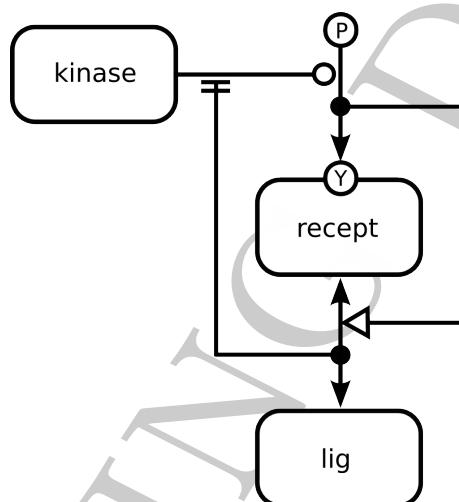
Figure 3.1: *A receptor can bind a ligand, and be phosphorylated by a kinase. The phosphorylation can only take place on the free receptor. The binding reaction is stimulated by the phosphorylation. The figure is drawn using the SBGN Entity Relationship language.*

```xml
<?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="phosphoreceptor">

    <listOfCompartments>
      <compartment id="cell" constant="true" size="1" />
    </listOfCompartments>

<!-- ---------------------------------------------------------------------- -->
<!-- We need to define two species types to be able to describe all the reactions. -->
<!-- There is not species type for the kinase, that is not used in selectors. -->
<!-- ---------------------------------------------------------------------- -->

    <multi:listOfSpeciesTypes>
```

```
        <multi:speciesType multi:id="st_recept"
                           multi:name="receptor"
                           multi:bindingSite="true">
      <multi:listOfStateFeatures>
        <multi:stateFeature multi:id="Y" multi:name="tyrosine">
          <multi:listOfPossibleValues>
            <multi:possibleValue multi:id="P" multi:name="phosphorylated" />
            <multi:possibleValue multi:id="nonP" multi:name="nonphosphorylated" />
          </multi:listOfPossibleValues>
        </multi:stateFeature>
      </multi:listOfStateFeatures>
    </multi:speciesType>

    <multi:speciesType multi:id="st_lig" multi:name="ligand" multi:bindingSite="true" />
  </multi:listOfSpeciesTypes>

  <multi:listOfSelectors>
```


recept

```
<!-- ------------------------------------------------------------------------------ -->
<!-- Description of receptor, unbound, unphosphorylated                              -->
<!-- ------------------------------------------------------------------------------ -->

      <multi:selector multi:id="sel_freereceptor_nonP">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_recept" multi:speciesType="st_recept"
                                  multi:minOccur="1" multi:maxOccur="1">
          <multi:listOfStateFeatureInstances>
            <multi:stateFeatureInstance multi:stateFeature="Y">
              <multi:stateFeatureValue multi:possibleValue="nonphosphorylated" />
            </multi:stateFeatureInstance>
          </multi:listOfStateFeatureInstances>
        </multi:speciesTypeState>
      </multi:listOfSpeciesTypeStates>
      <multi:listOfUnboundBindingSites>
        <multi:bindingSiteReference multi:speciesTypeState="sts_recept" />
      </multi:listOfUnboundBindingSites>
    </multi:selector>
```


recept

```
<!-- ------------------------------------------------------------------------------ -->
<!-- Description of receptor, unbound, phosphorylated                                -->
<!-- ------------------------------------------------------------------------------ -->
      <multi:selector multi:id="sel_freereceptor_P">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_recept" multi:speciesType="st_recept"
                                  multi:minOccur="1" multi:maxOccur="1">
          <multi:listOfStateFeatureInstances>
            <multi:stateFeatureInstance multi:stateFeature="Y">
              <multi:stateFeatureValue multi:possibleValue="phosphorylated" />
```

```
            </multi:stateFeatureInstance>
          </multi:listOfStateFeatureInstances>
        </multi:speciesTypeState>
      </multi:listOfSpeciesTypeStates>
      <multi:listOfUnboundBindingSites>
        <multi:bindingSiteReference multi:speciesTypeState="sts_recept" />
      </multi:listOfUnboundBindingSites>
    </multi:selector>
```

```
lig
```

```
<!-- ----------------------------------------------------------------------- -->
<!-- Description of ligand, unbound                                           -->
<!-- ----------------------------------------------------------------------- -->
    <multi:selector multi:id="sel_freeligand">
      <multi:listOfSpeciesTypeStates>
        <multi:speciesTypeState multi:id="sts_lig" multi:speciesType="st_lig"
                                multi:minOccur="1" multi:maxOccur="1" />
      </multi:listOfSpeciesTypeStates>
      <multi:listOfUnboundBindingSites>
        <multi:bindingSiteReference multi:speciesTypeState="sts_lig" />
      </multi:listOfUnboundBindingSites>
    </multi:selector>
```

```
  ?
recept

lig
```

```
<!-- ----------------------------------------------------------------------- -->
<!-- Description of the ligand-receptor complex                               -->
<!-- ----------------------------------------------------------------------- -->
    <multi:selector multi:id="sel_complex">
      <multi:listOfSpeciesTypeStates>
        <multi:speciesTypeState multi:id="sts_recept" multi:speciesType="st_recept"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_lig" multi:speciesType="st_lig"
                                multi:minOccur="1" multi:maxOccur="1" />
      </multi:listOfSpeciesTypeStates>
      <multi:listOfBonds>
        <multi:bond multi:occurrence="required">
          <multi:bindingSiteReference multi:speciesTypeState="sts_lig" />
          <multi:bindingSiteReference multi:speciesTypeState="sts_recept" />
        </multi:bond>
      </multi:listOfBonds>
    </multi:selector>

  </multi:listOfSelectors>
```

```xml
<listOfSpecies>
  <species id="kinase" compartment="cell" boundaryCondition="false"
          hasOnlySubstanceUnit="false" constant="false" initialAmount="100" />
  <species id="recept" compartment="cell" multi:speciesType="st_recept"
          boundaryCondition="false" hasOnlySubstanceUnit="false"
          constant="false" initialAmount="100" >
    <multi:listOfSpeciesTypeInstances>
      <multi:SpeciesTypeInstance multi:id="freereceptor_nonP"
                                 multi:initialAmount="100">
        <multi:listOfSelectorReferences>
          <multi:selectorReference multi:selector="sel_freereceptor_nonP" />
        </multi:listOfSelectorReferences>
      </multi:speciesTypeInstance>
      <multi:SpeciesTypeInstance multi:id="freereceptor_P"
                                 multi:initialAmount="0">
        <multi:listOfSelectorReferences>
          <multi:selectorReference multi:selector="sel_freereceptor_P" />
        </multi:listOfSelectorReferences>
      </multi:speciesTypeInstance>
      <multi:SpeciesTypeInstance multi:id="boundreceptor">
        <multi:listOfSelectorReferences>
          <multi:selectorReference multi:selector="sel_complex">
        </multi:listOfSelectorReferences>
      </multi:speciesTypeInstance>
    </multi:listOfSpeciesTypeInstances>
  </species>
  <species id="lig" compartment="cell" multi:speciesType="st_lig"
                  boundaryCondition="false"
          hasOnlySubstanceUnit="false" constant="false" initialAmount="1000" >
    <multi:listOfSpeciesTypeInstances>
      <multi:SpeciesTypeInstance multi:id="freeligand" multi:initialAmount="1000">
        <multi:listOfSelectorReferences>
          <multi:selectorReference multi:selector="sel_freeligand" />
        </multi:listOfSelectorReferences>
      </multi:speciesTypeInstance>
      <multi:SpeciesTypeInstance multi:id="boundligand" multi:initialAmount="0">
        <multi:listOfSelectorReferences>
          <multi:selectorReference multi:selector="sel_complex">
        </multi:listOfSelectorReferences>
      </multi:speciesTypeInstance>
    </multi:listOfSpeciesTypeInstances>
  </species>
</listOfSpecies>

<listOfReactions>
```



```xml
<!-- ------------------------------------------------------------------------- -->
<!-- Phosphorylation of the receptor by the kinase when unbound                -->
<!-- ------------------------------------------------------------------------- -->

    <reaction id="phosphorylation" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference species="recept" stoichiometry="1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="freeRnonP"
                                      multi:speciesTypeInstance="freereceptor_nonP" />
          </multi:listOfSpeciesRestriction>
```

```xml
          </speciesReference>
        </listOfReactants>
        <listOfProducts>
          <multi:productSpeciesReference species="recept" stoichiometry="1">
            <multi:listOfSpeciesRestriction>
              <multi:speciesRestriction multi:id="freeRP"
                                         multi:speciesTypeInstance="freereceptor_P" />
            </multi:listOfSpeciesRestriction>
          </multi:productSpeciesReference>
        </listOfProducts>
        <listOfModifiers>
          <modifierSpeciesReference species="kinase" />
        </listOfModifiers>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML" />
        </kineticLaw>
        <multi:listOfReactionRules>
          <multi:reactionRule multi:id="PhosphorylationUnbound">
            <multi:listOfConditions>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeRnonP"/>
            </multi:listOfConditions>
            <multi:listOfResults>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeRP"/>
            </multi:listOfResults>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> cell </ci>
                  <ci> Vm </ci>
                  <apply>
                    <divide />
                    <ci> recept </ci>
                    <apply>
                      <plus />
                      <ci> recept </ci>
                      <ci> Km </ci>
                    </apply>
                  </apply>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="Km" value="1" />
                <localParameter id="Vm" value="1" />
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>
        </multi:listOfReactionRules>

      </reaction>
```
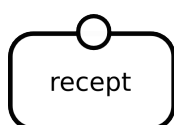
```
<!-- ---------------------------------------------------------------------------- -->
<!-- Ligand binding, when the receptor is phosphorylated or not                   -->
<!-- ---------------------------------------------------------------------------- -->

    <reaction id="receptLigBinding" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference id="react_recept" species="recept" stoichiometry="1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="freeRnonP"
                                      multi:speciesTypeInstance="freereceptor_nonP" />
            <multi:speciesRestriction multi:id="freeRP"
                                      multi:speciesTypeInstance="freereceptor_P" />
          </multi:listOfSpeciesRestriction>
        </speciesReference>
        <speciesReference id="react_lig" species="lig" stoichiometry="1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="freeL"
                                      multi:speciesTypeInstance="freeligand" />
          </multi:listOfSpeciesRestriction>
        </speciesReference>
      </listOfReactants>
      <listOfProducts>
        <multi:productSpeciesReference species="recept" stoichiometry="1"
                            correspondingReactant="react_recept">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="boundR"
                                      multi:speciesTypeInstance="boundreceptor" />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
        <multi:productSpeciesReference species="lig" stoichiometry="1"
                            correspondingReactant="react_lig">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="boundL"
                                      multi:speciesTypeInstance="boundligand" />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
      </listOfProducts>
      <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML" />
      </kineticLaw>
```
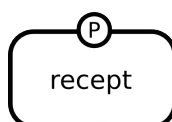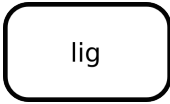
```xml
        <multi:listOfReactionRules>
          <multi:reactionRule multi:id="bindingNonPhospho">
            <multi:listOfConditions>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeRnonP" />
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeL" />
            </multi:listOfConditions>
            <multi:listOfResults>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="boundR" />
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="boundL" />
            </multi:listOfResults>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> cell </ci>
                  <ci> kon_nonphos </ci>
                  <ci> recept </ci>
                  <ci> lig </ci>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="kon_nonphos" value="1">
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>
          <multi:reactionRule multi:id="bindingPhospho">
            <multi:listOfConditions>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeRP" />
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="freeL" />
            </multi:listOfConditions>
            <multi:listOfResults>
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="boundR" />
              <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="boundL" />
            </multi:listOfResults>

            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> cell </ci>
                  <ci> kon_phos </ci>
                  <ci> recept </ci>
                  <ci> lig </ci>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="kon_phos" value="10">
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>
        </multi:listOfReactionRules>
      </reaction>
    </listOfReactions>

  </model>
</sbml>
```
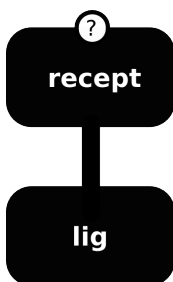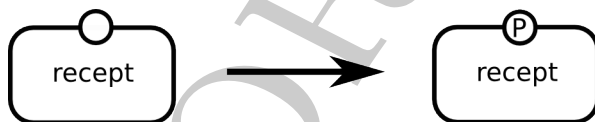
3.2 Microtubule polymerisation
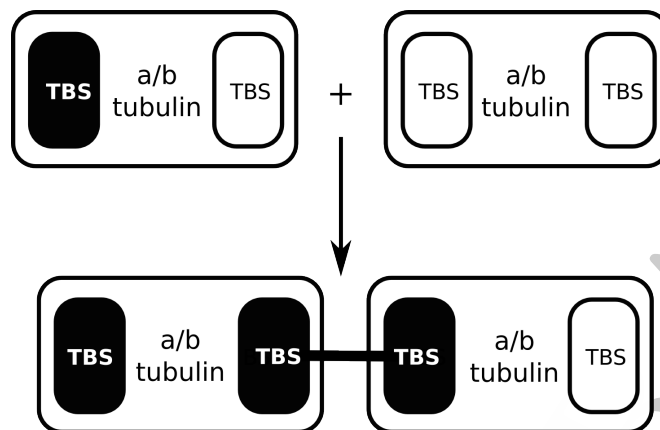
Figure 3.2: *A microtubule of undefined length can be extended through a polimerization reaction.*

Within the context of the example detailed below, where only microtubules and tubulin dimers can bind, the figure 3.2 is effectively equivalent to the figure , both representing the same reactions.
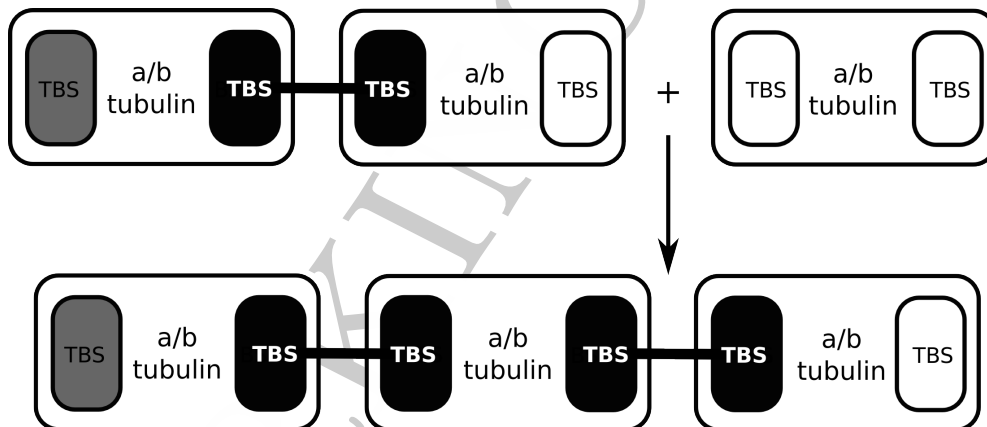
Figure 3.3: *A microtubule of undefined length can be extended through a polimerization reaction.*

```xml
<?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">

    <listOfCompartments>
      <compartment id="cell" constant="true" size="1" />
    </listOfCompartments>
```

```
<multi:listOfSpeciesTypes>

    <multi:speciesType multi:id="st_tubulin" multi:name="tubulin" multi:bindingSite="false">
    <multi:speciesType multi:id="st_tbs" multi:name="tbs" multi:bindingSite="true" />

</multi:listOfSpeciesTypes>

<multi:listOfSelectors>
```



```
<!-- -------------------------------------------------------------------------- -->
<!-- Description of a free dimer of tubulin                                      -->
<!-- -------------------------------------------------------------------------- -->

    <multi:selector multi:id="sel_tubulin_free">
      <multi:listOfSpeciesTypeStates>
        <multi:speciesTypeState multi:id="sts_tbs1" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tbs2" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tubulin" multi:speciesType="st_tubulin"
                                multi:minOccur="1" multi:maxOccur="1" />
          <multi:listOfContainedSpeciesTypes>
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs1" />
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs2" />
          </multi:listOfContainedSpeciesTypes>
        </multi:speciesTypeStates>
      </multi:listOfSpeciesTypeStates>
      <multi:listOfUnboundBindingSites>
        <multi:bindingSiteReference multi:speciesTypeState="sts_tbs1" />
        <multi:bindingSiteReference multi:speciesTypeState="sts_tbs2" />
      </multi:listOfUnboundBindingSites>
    </multi:selector>
```



```
<!-- -------------------------------------------------------------------------- -->
<!-- To signify a microtubule, we describe a tubulin with one binding site free, the -->
<!-- other being bound                                                          -->
<!-- -------------------------------------------------------------------------- -->
    <multi:selector multi:id="sel_microtubule">
      <multi:listOfSpeciesTypeStates>
        <multi:speciesTypeState multi:id="sts_tbs1" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tbs2" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tubulin" multi:speciesType="st_tubulin"
```

```xml
                                    multi:minOccur="1" multi:maxOccur="1" />
        <multi:listOfContainedSpeciesTypes>
          <multi:containedSpeciesType multi:speciesTypeState="sts_tbs1" />
          <multi:containedSpeciesType multi:speciesTypeState="sts_tbs2" />
        </multi:listOfContainedSpeciesTypes>
      </multi:speciesTypeStates>
    </multi:listOfSpeciesTypeStates>
    <multi:listOfUnboundBindingSites>
      <multi:bindingSiteReference multi:speciesTypeState="sts_tbs1" />
    </multi:listOfUnboundBindingSites>
    <multi:listOfBonds>
      <multi:bond multi:occurrence="required" >
        <multi:bindingSiteReference multi:speciesTypeState="sts_tbs2" />
      </multi:bond>
    </multi:listOfBonds>
  </multi:selector>
```
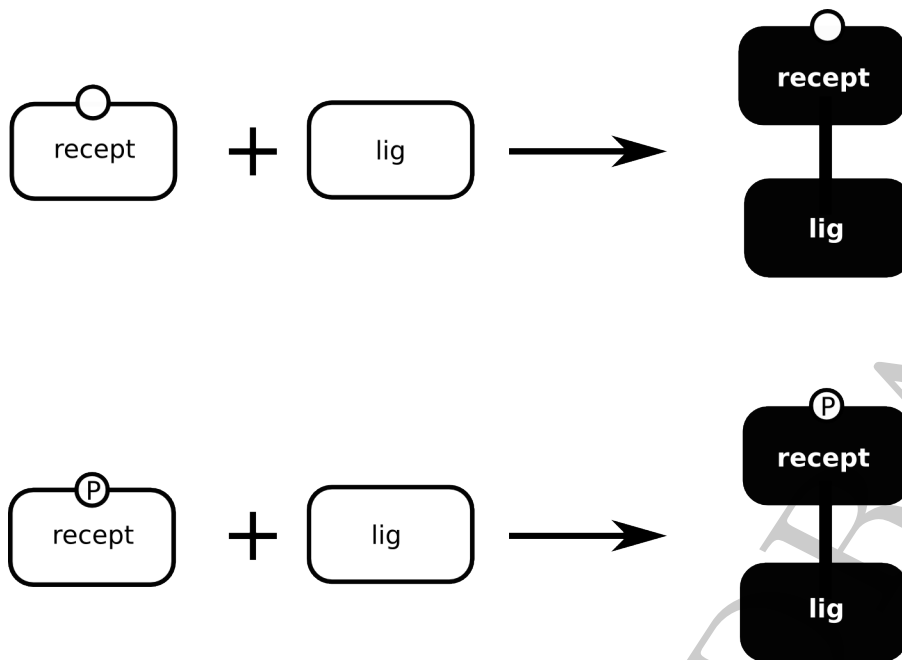


```xml
<!-- ------------------------------------------------------------------------ -->
<!-- This is a microtubule with at least three monomers                       -->
<!-- ------------------------------------------------------------------------ -->
    <multi:selector multi:id="sel_microtubuleExtended">
      <multi:listOfSpeciesTypeStates>
        <multi:speciesTypeState multi:id="sts_tbs1" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tbs2" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tbs3" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tbs4" multi:speciesType="st_tbs"
                                multi:minOccur="1" multi:maxOccur="1" />
        <multi:speciesTypeState multi:id="sts_tubulin1" multi:speciesType="st_tubulin"
                                multi:minOccur="1" multi:maxOccur="1">
          <multi:listOfContainedSpeciesTypes>
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs1" />
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs2" />
          </multi:listOfContainedSpeciesTypes>
        </multi:speciesTypeStates>
        <multi:speciesTypeStates multi:id="sts_tubulin2" multi:speciesType="st_tubulin"
                                 multi:minOccur="1" multi:maxOccur="1">
          <multi:listOfContainedSpeciesTypes>
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs3" />
            <multi:containedSpeciesType multi:speciesTypeState="sts_tbs4" />
          </multi:listOfContainedSpeciesTypes>
        </multi:speciesTypeStates>
      </multi:listOfSpeciesTypeStates>
      <multi:listOfBonds>
        <multi:bond multi:occurrence="required" >
          <multi:bindingSiteReference multi:speciesTypeState="sts_tbs2" />
          <multi:bindingSiteReference multi:speciesTypeState="sts_tbs3" />
        </multi:bond>
      </multi:listOfBonds>
      <multi:listOfUnboundBindingSites>
        <multi:bindingSiteReference multi:speciesTypeState="sts_tbs1" />
      </multi:listOfUnboundBindingSites>
    </multi:selector>
```

```
        </multi:listOfSelectors>

        <listOfSpecies>
          <species id="tubulin" compartment="cell" multi:speciesType="st_tubulin"
                   boundaryCondition="false" hasOnlySubstanceUnit="false"
                   constant="false" initialAmount="500" >
            <multi:listOfSpeciesTypeInstances>
              <multi:SpeciesTypeInstance multi:id="free_tubulin" multi:initialAmount="100">
                <multi:listOfSelectorReferences>
                  <multi:selectorReference multi:selector="sel_tubulin_free" />
                </multi:listOfSelectorReferences>
              </multi:speciesTypeInstance>
              <multi:SpeciesTypeInstance multi:id="microtubule" multi:initialAmount="400">
                <multi:listOfSelectorReferences>
                  <multi:selectorReference multi:selector="sel_microtubule" />
                </multi:listOfSelectorReferences>
              </multi:speciesTypeInstance>
              <multi:SpeciesTypeInstance multi:id="microtubuleExtended" multi:initialAmount="0">
                <multi:listOfSelectorReferences>
                  <multi:selectorReference multi:selector="sel_microtubuleExtended">
                </multi:listOfSelectorReferences>
              </multi:speciesTypeInstance>
            </multi:listOfSpeciesTypeInstances>
          </species>
        </listOfSpecies>

        <listOfReactions>
          <reaction id="tubulin_binding" reversible="false" fast="false">
            <listOfReactants>
              <speciesReference species="tubulin" stoichiometry="1">
                <multi:listOfSpeciesRestriction>
                  <multi:speciesRestriction multi:id="free_tubul"
                                            multi:speciesTypeInstance="free_tubulin"/>
                  <multi:speciesRestriction multi:id="micro"
                                            multi:speciesTypeInstance="microtubule"/>
                </multi:listOfSpeciesRestriction>
              </speciesReference>
            </listOfReactants>
            <listOfProducts>
              <multi:productSpeciesReference species="tubulin" stoichiometry="1">
                <multi:listOfSpeciesRestriction>
                  <multi:speciesRestriction multi:id="micro_extended"
                                            multi:speciesTypeInstance="microtubule_extended"/>
                </multi:listOfSpeciesRestriction>
              </multi:productSpeciesReference>
            </listOfProducts>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" />
            </kineticLaw>
            <multi:listOfReactionRules>
              <multi:reactionRule multi:id="bindingNonPhospho">
                <multi:listOfConditions>
                  <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="free_tubul"/>
                  <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="micro"/>
                </multi:listOfConditions>
                <multi:listOfResults>
                  <multi:speciesTypeRestrictionReference
                                            multi:speciesTypeRestriction="micro_extended"/>
                </multi:listOfResults>
                <kineticLaw>
                  <math xmlns="http://www.w3.org/1998/Math/MathML" >
                    <apply>
                      <times />
                      <ci> cell </ci>
                      <ci> kon </ci>
                      <ci> tubulin </ci>
```
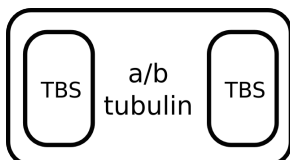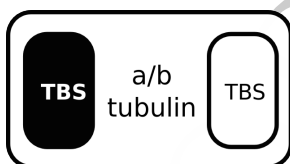
```
            </apply>
          </math>
          <listOfLocalParameters>
            <localParameter id="kon" value="1">
          </listOfLocalParameters>
        </kineticLaw>
      </multi:reactionRule>
    </multi:listOfReactionRules>
  </reaction>
</listOfReactions>

  </model>
</sbml>
```

3.3 EGF receptor

Figure 3.4: *Binding of EGF on EGF receptor stimulates the dimerization of the latter, that then autophosphorylate. This phosphorylation increases the binding of SHC to the receptor. The figure is drawn using the SBGN Entity Relationship language.*

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

56

```
          xmlns:multi="http://www.sbml.org/sbml/level3/version1/multi/version1"
          multi:required="true">

   <model name="EGF receptor">

<!-- ------------------------------------------------------------------------ -->
<!-- Three compartments are defined, to contain the different species taking part -->
<!-- to the EGFR reactions                                                    -->
<!-- ------------------------------------------------------------------------ -->

    <listOfCompartments>
      <compartment id="extracellular" constant="true" spatialDimension="3" size="1" />
      <compartment id="membrane" constant="true" spatialDimension="2" size="1" />
      <compartment id="intracellular" constant="true" spatialDimension="3" size="1" />
    </listOfCompartments>

<!-- ------------------------------------------------------------------------ -->
<!-- We need to define seven species types to be able to describe all the reactions. -->
<!-- ------------------------------------------------------------------------ -->

    <multi:listOfSpeciesTypes>
      <multi:speciesType multi:id="st_EGF" multi:name="EGF" multi:bindingSite="yes">
      <multi:speciesType multi:id="st_TGF" multi:name="TGF" multi:bindingSite="yes">
      <multi:speciesType multi:id="st_EGFBS" multi:name="EGF binding site" multi:bindingSite="true" />
      <multi:speciesType multi:id="st_dimer" multi:name="dimerisation domain" multi:bindingSite="true"
      <multi:speciesType multi:id="st_signal" multi:name="signalling domain" multi:bindingSite="true">
        <multi:listOfStateFeatures>
          <multi:stateFeature multi:id="Y" multi:name="tyrosine">
            <multi:listOfPossibleValues>
              <multi:possibleValue multi:id="P" name="phosphorylated" />
              <multi:possibleValue multi:id="NP" name="non-phosphorylated" />
            </multi:listOfPossibleValues>
          </multi:stateFeature>
        </multi:listOfStateFeatures>
      </multi:speciesType>
      <multi:speciesType multi:id="st_EGFR" multi:name="EGFR" multi:bindingSite="no">
      <multi:speciesType multi:id="st_SHC" multi:name="SHC" multi:bindingSite="yes">
    </multi:listOfSpeciesTypes>

<!-- ------------------------------------------------------------------------ -->
<!-- Selectors are used to describe topologies and connectivities used in the model. -->
<!-- ------------------------------------------------------------------------ -->

    <multi:listOfSelectors>
```

EGF

```
<!-- ------------------------------------------------------------------------ -->
<!-- Description of free EGF, not bound                                       -->
<!-- ------------------------------------------------------------------------ -->

      <multi:selector multi:id="sel_freeEGF">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_EGF" multi:speciesType="st_EGF"
                                   multi:minOccur="1" multi:maxOccur="1" />
        </multi:listOfSpeciesTypeStates>
        <multi:listOfUnboundBindingSites>
          <multi:bindingSiteReference multi:speciesTypeState="sts_EGF" />
        </multi:listOfUnboundBindingSites>
      </multi:selector>
```
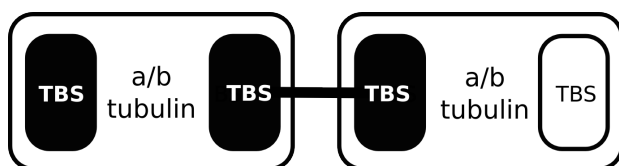
EGF

```
<!-- ------------------------------------------------------------------------- -->
<!-- Description of EGF bound (to EGFR but this is not precised).               -->
<!-- ------------------------------------------------------------------------- -->

     <multi:selector multi:id="sel_boundEGF">
       <multi:listOfSpeciesTypeStates>
         <multi:speciesTypeState multi:id="sts_EGF" multi:speciesType="st_EGF"
                                 multi:minOccur="1" multi:maxOccur="1" />
       </multi:listOfSpeciesTypeStates>
       <multi:listOfBonds>
         <multi:bond occurence="required">
           <multi:bindingSiteReference multi:speciesTypeState="sts_EGF" />
         </multi:bond>
       </multi:listOfBonds>
     </multi:selector>
```

TGFα

```
<!-- ------------------------------------------------------------------------- -->
<!-- Description of free TGF, not bound                                        -->
<!-- ------------------------------------------------------------------------- -->

     <multi:selector multi:id="sel_freeTGF">
       <multi:listOfSpeciesTypeStates>
         <multi:speciesTypeState multi:id="sts_TGF" multi:speciesType="st_TGF"
                                 multi:minOccur="1" multi:maxOccur="1" />
       </multi:listOfSpeciesTypeStates>
       <multi:listOfUnboundBindingSites>
         <multi:bindingSiteReference multi:speciesTypeState="sts_TGF" />
       </multi:listOfUnboundBindingSites>
     </multi:selector>
```

TGFα

```
<!-- ------------------------------------------------------------------------- -->
<!-- Description of TGF bound (to EGFR but this is not precised)               -->
<!-- ------------------------------------------------------------------------- -->

     <multi:selector multi:id="sel_boundTGF">
       <multi:listOfSpeciesTypeStates>
         <multi:speciesTypeState multi:id="sts_TGF" multi:speciesType="st_TGF"
                                 multi:minOccur="1" multi:maxOccur="1" />
       </multi:listOfSpeciesTypeStates>
       <multi:listOfBonds>
         <multi:bond occurence="required">
           <multi:bindingSiteReference multi:speciesTypeState="sts_TGF" />
         </multi:bond>
       </multi:listOfBonds>
     </multi:selector>
```

```
<!-- ------------------------------------------------------------------------ -->
<!-- Description of free EGF binding site, not bound to EGF or TGF.          -->
<!-- ------------------------------------------------------------------------ -->

        <multi:selector multi:id="sel_freeEGFBS">
          <multi:listOfSpeciesTypeStates>
            <multi:speciesTypeState multi:id="sts_EGFBS" multi:speciesType="st_EGFBS"
                                    multi:minOccur="1" multi:maxOccur="1" />
            <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                    multi:minOccur="1" multi:maxOccur="1" >
              <multi:listOfContainedSpeciesTypes>
                <multi:containedSpeciesType multi:speciesTypeState="sts_EGFBS" />
              </multi:listOfContainedSpeciesType>
            </multi:speciesTypeState>
          </multi:listOfSpeciesTypeStates>
          <multi:listOfUnboundBindingSites>
            <multi:bindingSiteReference multi:speciesTypeState="sts_EGFBS" />
          </multi:listOfUnboundBindingSites>
        </multi:selector>
```



```
<!-- ------------------------------------------------------------------------ -->
<!-- Description of EGF binding site bound (either to EGF or TGF but this is not -->
<!-- precised. Another approach would have been to create two selectors, describing -->
<!-- EGFR bound to EGF and EGFR bound to TGF, and combine them at the level of the -->
<!-- species.                                                                -->
<!-- ------------------------------------------------------------------------ -->

        <multi:selector multi:id="sel_boundEGFBS">
          <multi:listOfSpeciesTypeStates>
            <multi:speciesTypeState multi:id="sts_EGFBS" multi:speciesType="st_EGFBS"
                                    multi:minOccur="1" multi:maxOccur="1" />
            <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                    multi:minOccur="1" multi:maxOccur="1" >
              <multi:listOfContainedSpeciesTypes>
                <multi:containedSpeciesType multi:speciesTypeState="sts_EGFBS" />
              </multi:listOfContainedSpeciesType>
            </multi:speciesTypeState>
          </multi:listOfSpeciesTypeStates>
          <multi:listOfBonds>
            <multi:bond occurence="required">
              <multi:bindingSiteReference multi:speciesTypeState="sts_EGFBS" />
            </multi:bond>
```
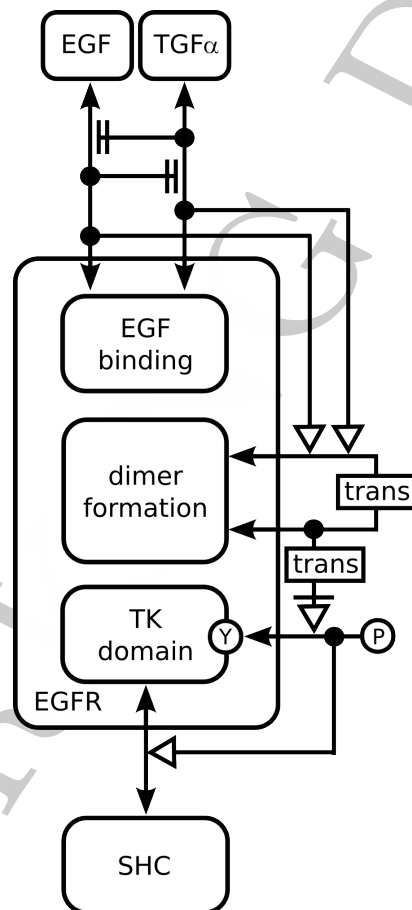
```
          </multi:listOfBonds>
      </multi:selector>
```

EGF binding

dimer formation

EGFR

```
<!-- ----------------------------------------------------------------------- -->
<!-- Description of free EGFR dimerisation domain, not bound to another one, but in   -->
<!-- receptors that are not bound to EGF or TGF.                                        -->
<!-- ----------------------------------------------------------------------- -->

      <multi:selector multi:id="sel_freeDimerNoligand">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_EGFBS" multi:speciesType="st_EGFBS"
                                  multi:minOccur="1" multi:maxOccur="1" />
          <multi:speciesTypeState multi:id="sts_dimer" multi:speciesType="st_dimer"
                                  multi:minOccur="1" multi:maxOccur="1" />
        </multi:speciesTypeState>
          <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                  multi:minOccur="1" multi:maxOccur="1" >
            <multi:listOfContainedSpeciesTypes>
              <multi:containedSpeciesType multi:speciesTypeState="sts_EGFBS" />
              <multi:containedSpeciesType multi:speciesTypeState="sts_dimer" />
            </multi:listOfContainedSpeciesTypes>
          </multi:speciesTypeState>
        </multi:listOfSpeciesTypeStates>
        <multi:listOfUnboundBindingSites>
          <multi:bindingSiteReference multi:speciesTypeState="sts_dimer" />
          <multi:bindingSiteReference multi:speciesTypeState="sts_EGFBS" />
        </multi:listOfUnboundBindingSites>
      </multi:selector>
```

EGF binding

dimer formation

EGFR

```
<!-- ------------------------------------------------------------------------- -->
<!-- Description of free EGFR dimerisation domain, not bound to another one, but in  -->
<!-- receptors that are bound to EGF or TGF.                                    -->
<!-- ------------------------------------------------------------------------- -->

      <multi:selector multi:id="sel_freeDimerLigand">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_EGFBS" multi:speciesType="st_EGFBS"
                                  multi:minOccur="1" multi:maxOccur="1" />
          <multi:speciesTypeState multi:id="sts_dimer" multi:speciesType="st_dimer"
                                  multi:minOccur="1" multi:maxOccur="1" />
          </multi:speciesTypeState>
          <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                  multi:minOccur="1" multi:maxOccur="1" >
            <multi:listOfContainedSpeciesTypes>
              <multi:containedSpeciesType multi:speciesTypeState="sts_EGFBS" />
              <multi:containedSpeciesType multi:speciesTypeState="sts_dimer" />
            </multi:listOfContainedSpeciesTypes>
          </multi:speciesTypeState>
        </multi:listOfSpeciesTypeStates>
        <multi:listOfBonds>
          <multi:bond occurence="required">
            <multi:bindingSiteReference multi:speciesTypeState="sts_EGFBS" />
          </multi:bond>
        </multi:listOfBonds>
        <multi:listOfUnboundBindingSites>
          <multi:bindingSiteReference multi:speciesTypeState="sts_dimer" />
        </multi:listOfUnboundBindingSites>
      </multi:selector>
```
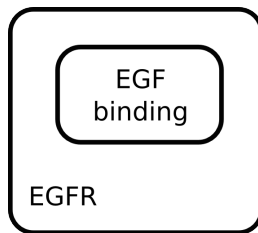


```
<!-- ------------------------------------------------------------------------- -->
<!-- Description of EGFR dimerisation domain bound (to another one but this is not  -->
<!-- described here)                                                           -->
<!-- ------------------------------------------------------------------------- -->

      <multi:selector multi:id="sel_boundDimer">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_dimer" multi:speciesType="st_dimer"
                                  multi:minOccur="1" multi:maxOccur="1" />
          <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                  multi:minOccur="1" multi:maxOccur="1" >
            <multi:listOfContainedSpeciesTypes>
              <multi:containedSpeciesType multi:speciesTypeState="sts_dimer" />
            </multi:listOfContainedSpeciesTypes>
          </multi:speciesTypeState>
        </multi:listOfSpeciesTypeStates>
        <multi:listOfBonds>
          <multi:bond occurence="required">
            <multi:bindingSiteReference multi:speciesTypeState="sts_dimer" />
          </multi:bond>
```
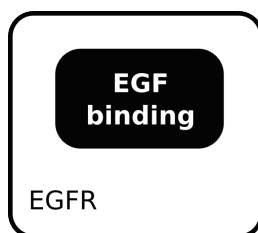
```
          </multi:listOfBonds>
        </multi:selector>
```



```
<!-- ------------------------------------------------------------------------------ -->
<!-- Description of EGFR dimerised with one signalling domain non-phosphorylated     -->
<!-- ------------------------------------------------------------------------------ -->

        <multi:selector multi:id="sel_EGFRDimerNonPhos">
          <multi:listOfSpeciesTypeStates>
            <multi:speciesTypeState multi:id="sts_dimer" multi:speciesType="st_dimer"
                                    multi:minOccur="1" multi:maxOccur="1" />
            <multi:speciesTypeState multi:id="sts_signal" multi:speciesType="st_signal"
                                    multi:minOccur="1" multi:maxOccur="1">
              <multi:listOfStateFeatureInstances>
                <multi:stateFeatureInstance multi:stateFeature="Y">
                  <multi:listOfStateFeatureValues>
                    <multi:stateFeatureValue multi:possibleValue="NP" />
                  </multi:listOfStateFeatureValues>
                </multi:stateFeatureInstance>
              </multi:listOfStateFeatureInstances>
            </multi:speciesTypeState>
            <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                    multi:minOccur="1" multi:maxOccur="1" >
              <multi:listOfContainedSpeciesTypes>
                <multi:containedSpeciesType multi:speciesTypeState="sts_dimer" />
                <multi:containedSpeciesType multi:speciesTypeState="sts_signal" />
              </multi:listOfContainedSpeciesTypes>
            </multi:speciesTypeState>
          </multi:listOfSpeciesTypeStates>
          <multi:listOfBonds>
            <multi:bond occurence="required">
              <multi:bindingSiteReference multi:speciesTypeState="sts_dimer" />
            </multi:bond>
          </multi:listOfBonds>
        </multi:selector>
```
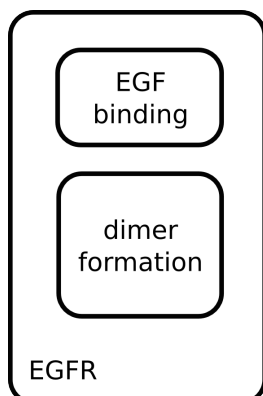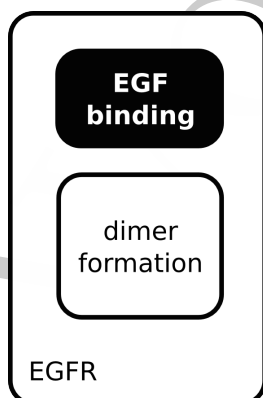
```xml
<!-- ------------------------------------------------------------------------ -->
<!-- Description of EGFR dimerised with one signalling domain phosphorylated   -->
<!-- ------------------------------------------------------------------------ -->

        <multi:selector multi:id="sel_EGFRDimerPhos">
          <multi:listOfSpeciesTypeStates>
            <multi:speciesTypeState multi:id="sts_dimer" multi:speciesType="st_dimer"
                                    multi:minOccur="1" multi:maxOccur="1" />
            <multi:speciesTypeState multi:id="sts_signal" multi:speciesType="st_signal"
                                    multi:minOccur="1" multi:maxOccur="1">
              <multi:listOfStateFeatureInstances>
                <multi:stateFeatureInstance multi:stateFeature="Y">
                  <multi:listOfStateFeatureValues>
                    <multi:stateFeatureValue multi:possibleValue="P" />
                  </multi:listOfStateFeatureValues>
                </multi:stateFeatureInstance>
              </multi:listOfStateFeatureInstances>
            </multi:speciesTypeState>
            <multi:speciesTypeState multi:id="sts_EGFR" multi:speciesType="st_EGFR"
                                    multi:minOccur="1" multi:maxOccur="1" >
              <multi:listOfContainedSpeciesTypes>
                <multi:containedSpeciesType multi:speciesTypeState="sts_dimer" />
                <multi:containedSpeciesType multi:speciesTypeState="sts_signal" />
              </multi:listOfContainedSpeciesTypes>
            </multi:speciesTypeState>
          </multi:listOfSpeciesTypeStates>
          <multi:listOfBonds>
            <multi:bond occurence="required">
              <multi:bindingSiteReference multi:speciesTypeState="sts_dimer" />
            </multi:bond>
          </multi:listOfBonds>
        </multi:selector>
```

SHC

```xml
<!-- ------------------------------------------------------------------------ -->
<!-- Description of free SHC, not bound to EGFR.                               -->
<!-- ------------------------------------------------------------------------ -->

        <multi:selector multi:id="sel_freeSHC">
          <multi:listOfSpeciesTypeStates>
            <multi:speciesTypeState multi:id="sts_SHC" multi:speciesType="st_SHC"
                                    multi:minOccur="1" multi:maxOccur="1" />
          </multi:listOfSpeciesTypeStates>
          <multi:listOfUnboundBindingSites>
            <multi:bindingSiteReference multi:speciesTypeState="sts_SHC" />
          </multi:listOfUnboundBindingSites>
        </multi:selector>
```
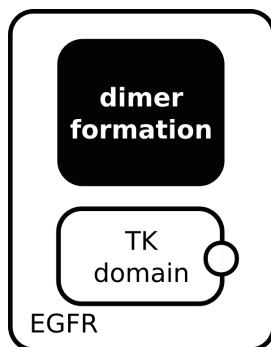
```
<!-- ------------------------------------------------------------------------ -->
<!-- Description of SHC bound to EGFR signalling domain                       -->
<!-- ------------------------------------------------------------------------ -->

       <multi:selector multi:id="sel_SHC_Signal">
         <multi:listOfSpeciesTypeStates>
           <multi:speciesTypeState multi:id="sts_SHC" multi:speciesType="st_SHC"
                                    multi:minOccur="1" multi:maxOccur="1" />
           <multi:speciesTypeState multi:id="sts_signal" multi:speciesType="st_signal"
                                    multi:minOccur="1" multi:maxOccur="1" />
         </multi:speciesTypeState>
         </multi:listOfSpeciesTypeStates>
         <multi:listOfBonds>
           <multi:bond occurence="required">
             <multi:bindingSiteReference multi:speciesTypeState="sts_SHC" />
             <multi:bindingSiteReference multi:speciesTypeState="sts_signal" />
           </multi:bond>
         </multi:listOfBonds>
       </multi:selector>
```



```
<!-- ------------------------------------------------------------------------ -->
<!-- EGFR signalling domain phosphorylated and not bound to SHC               -->
<!-- ------------------------------------------------------------------------ -->

       <multi:selector multi:id="sel_freeEGFRPhos">
         <multi:listOfSpeciesTypeStates>
           <multi:speciesTypeState multi:id="sts_signal" multi:speciesType="st_signal"
                                    multi:minOccur="1" multi:maxOccur="1">
             <multi:listOfStateFeatureInstances>
               <multi:stateFeatureInstance multi:stateFeature="Y">
                 <multi:listOfStateFeatureValues>
                   <multi:stateFeatureValue multi:possibleValue="P" />
                 </multi:listOfStateFeatureValues>
               </multi:stateFeatureInstance>
             </multi:listOfStateFeatureInstances>
           </multi:speciesTypeState>
         </multi:listOfSpeciesTypeStates>
         <multi:listOfUnboundBindingSites>
           <multi:bindingSiteReference multi:speciesTypeState="sts_signal" />
         </multi:listOfUnboundBindingSites>
       </multi:selector>
```
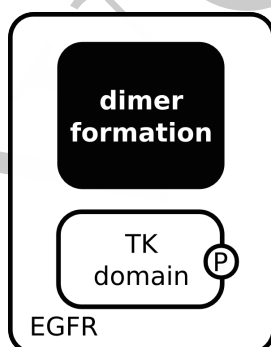
TK
domain

```xml
<!-- ---------------------------------------------------------------------- -->
<!-- EGFR signalling domain non-phosphorylated and not bound to SHC         -->
<!-- ---------------------------------------------------------------------- -->

      <multi:selector multi:id="sel_freeEGFRNonPhos">
        <multi:listOfSpeciesTypeStates>
          <multi:speciesTypeState multi:id="sts_signal" multi:speciesType="st_signal"
                                  multi:minOccur="1" multi:maxOccur="1">
            <multi:listOfStateFeatureInstances>
              <multi:stateFeatureInstance multi:stateFeature="Y">
                <multi:listOfStateFeatureValues>
                  <multi:stateFeatureValue multi:possibleValue="NP" />
                </multi:listOfStateFeatureValues>
              </multi:stateFeatureInstance>
            </multi:listOfStateFeatureInstances>
          </multi:speciesTypeState>
        </multi:listOfSpeciesTypeStates>
        <multi:listOfUnboundBindingSites>
          <multi:bindingSiteReference multi:speciesTypeState="sts_signal" />
        </multi:listOfUnboundBindingSites>
      </multi:selector>

   </multi:listOfSelectors>

<!-- ---------------------------------------------------------------------- -->
<!-- Six species describe the entities actually present in the simulations.  -->
<!-- ---------------------------------------------------------------------- -->

    <listOfSpecies>

<!-- ---------------------------------------------------------------------- -->
<!-- EGF, can be free or bound                                              -->
<!-- ---------------------------------------------------------------------- -->

      <species id="EGF" compartment="extracellular" multi:speciesType="st_EGF"
               boundaryCondition="false" hasOnlySubstanceUnit="false"
               constant="false" initialAmount="500" >
        <multi:listOfSpeciesTypeInstances>
          <multi:SpeciesTypeInstance multi:id="freeEGF" multi:initialAmount="500">
            <multi:listOfSelectorReferences>
              <multi:selectorReference multi:selector="sel_freeEGF" multi:negation="false"/>
            </multi:listOfSelectorReferences>
          </multi:speciesTypeInstance>
          <multi:SpeciesTypeInstance multi:id="boundEGF">
            <multi:listOfSelectorReferences>
              <multi:selectorReference multi:selector="sel_boundEGF" multi:negation="false"/>
            </multi:listOfSelectorReferences>
          </multi:speciesTypeInstance>
        </multi:listOfSpeciesTypeInstances>
      </species>

<!-- ---------------------------------------------------------------------- -->
<!-- TGF, can be free or bound                                              -->
<!-- ---------------------------------------------------------------------- -->

      <species id="TGF" compartment="extracellular" multi:speciesType="st_TGF"
               boundaryCondition="false" hasOnlySubstanceUnit="false"
               constant="false" initialAmount="500" >
        <multi:listOfSpeciesTypeInstances>
          <multi:SpeciesTypeInstance multi:id="freeTGF" multi:initialAmount="500">
            <multi:listOfSelectorReferences>
              <multi:selectorReference multi:selector="sel_freeTGF" multi:negation="false"/>
```
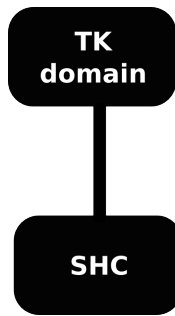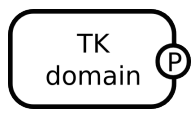
```
                </multi:listOfSelectorReferences>
              </multi:speciesTypeInstance>
              <multi:SpeciesTypeInstance multi:id="boundTGF">
                <multi:listOfSelectorReferences>
                  <multi:selectorReference multi:selector="sel_boundTGF" multi:negation="false"/>
                </multi:listOfSelectorReferences>
            </multi:listOfSpeciesTypeInstances>
          </species>

<!-- ------------------------------------------------------------------------------- -->
<!-- EGF binding site, can be free or bound                                          -->
<!-- ------------------------------------------------------------------------------- -->

        <species id="EGFBS" compartment="extracellular" multi:speciesType="st_EGFBS"
                 boundaryCondition="false" hasOnlySubstanceUnit="false"
                 constant="false" initialAmount="50" >
          <multi:listOfSpeciesTypeInstances>
            <multi:SpeciesTypeInstance multi:id="freeEGFBS" multi:initialAmount="50">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_freeEGFBS" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
            <multi:SpeciesTypeInstance multi:id="boundEGFBS">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_boundEGFBS" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:listOfSpeciesTypeInstances>
          </species>

<!-- ------------------------------------------------------------------------------- -->
<!-- EGF dimerisation domain, can be free or bound                                   -->
<!-- ------------------------------------------------------------------------------- -->

        <species id="dimer" compartment="membrane" multi:speciesType="st_dimer"
                 boundaryCondition="false" hasOnlySubstanceUnit="false"
                 constant="false" initialAmount="50" >
          <multi:listOfSpeciesTypeInstances>
            <multi:SpeciesTypeInstance multi:id="freeDimerNoligand" multi:initialAmount="50">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_freeDimerNoligand" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
            <multi:SpeciesTypeInstance multi:id="freeDimerLigand">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_freeDimerNoligand" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            <multi:SpeciesTypeInstance multi:id="boundDimer">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_boundDimer" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:listOfSpeciesTypeInstances>
          </species>

<!-- ------------------------------------------------------------------------------- -->
<!-- EGF signalling domain, can be phosphorylated or not, bound to SHC or not        -->
<!-- ------------------------------------------------------------------------------- -->

        <species id="signal" compartment="intracellular" multi:speciesType="st_signal"
                 boundaryCondition="false" hasOnlySubstanceUnit="false"
                 constant="false" initialAmount="50" >
          <multi:listOfSpeciesTypeInstances>
            <multi:SpeciesTypeInstance multi:id="DimerNonPhosSignal" multi:initialAmount="50">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_EGFRDimerNonPhos" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
```

```xml
            <multi:SpeciesTypeInstance multi:id="DimerPhosSignal">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_EGFRDimerPhos" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
            <multi:SpeciesTypeInstance multi:id="freeNonPhosSignal">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_freeEGFRNonPhos" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
            <multi:SpeciesTypeInstance multi:id="freePhosSignal">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_freeEGFRPhos" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
             <multi:SpeciesTypeInstance multi:id="boundSignal">
              <multi:listOfSelectorReferences>
                <multi:selectorReference multi:selector="sel_SHC_Signal" multi:negation="false"/>
              </multi:listOfSelectorReferences>
            </multi:speciesTypeInstance>
         </multi:listOfSpeciesTypeInstances>
       </species>

<!-- ------------------------------------------------------------------------------ -->
<!-- SHC, can be free or bound                                                      -->
<!-- ------------------------------------------------------------------------------ -->

    <species id="SHC" compartment="extracellular" multi:speciesType="st_TGF"
             boundaryCondition="false" hasOnlySubstanceUnit="false"
             constant="false" initialAmount="500" >
       <multi:listOfSpeciesTypeInstances>
         <multi:SpeciesTypeInstance multi:id="freeSHC" multi:initialAmount="500">
           <multi:listOfSelectorReferences>
             <multi:selectorReference multi:selector="sel_freeSHC" multi:negation="false"/>
           </multi:listOfSelectorReferences>
         </multi:speciesTypeInstance>
         <multi:SpeciesTypeInstance multi:id="boundSHC">
           <multi:listOfSelectorReferences>
             <multi:selectorReference multi:selector="sel_SHC_Signal" multi:negation="false"/>
           </multi:listOfSelectorReferences>
         </multi:speciesTypeInstance>
       </multi:listOfSpeciesTypeInstances>
     </species>


    </listOfSpecies>

<!-- ------------------------------------------------------------------------------ -->
<!-- 5 reactions are sufficient to describe the system                              -->
<!-- ------------------------------------------------------------------------------ -->

    <listOfReactions>
```
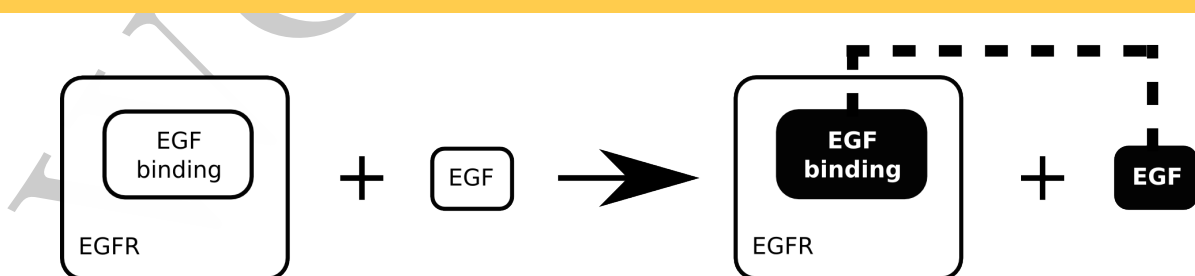
```xml
<!-- ------------------------------------------------------------------------- -->
<!-- Binding of EGF to EGF receptor.                                           -->
<!-- ------------------------------------------------------------------------- -->

        <reaction id="EGFbinding" reversible="false" fast="false">
         <listOfReactants>
           <speciesReference id="react_EGF" species="EGF" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_freeEGF"
                                    multi:speciesTypeInstance="freeEGF" />
             </multi:listOfSpeciesRestriction>
           </speciesReference>
           <speciesReference id="react_EGFBS" species="EGFBS" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_freeEGFBS"
                                    multi:speciesTypeInstance="freeEGFBS" />
             </multi:listOfSpeciesRestriction>
           </speciesReference>
         </listOfReactants>

<!-- ------------------------------------------------------------------------- -->
<!-- In this binding reaction, and the following two, we represent the products -->
<!-- as separate components. See the binding of SHC on EGFR to see how we can also -->
<!-- represent complexes                                                       -->
<!-- ------------------------------------------------------------------------- -->

         <listOfProducts>
           <multi:productSpeciesReference species="EGF" stoichiometry="1"
                               correspondingReactant="react_EGF">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_boundEGF"
                                    multi:speciesTypeInstance="boundEGF" />
             </multi:listOfSpeciesRestriction>
           </multi:productSpeciesReference>

<!-- ------------------------------------------------------------------------- -->
<!-- Since only free EGF is involved in this reaction, and not TGF, it it sufficient -->
<!-- to describe EGFR bound, without precising what it is bound to.             -->
<!-- ------------------------------------------------------------------------- -->

           <multi:productSpeciesReference species="EGFBS" stoichiometry="1"
                               correspondingReactant="react_EGFBS">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_boundEGFBS"
                                    multi:speciesTypeInstance="boundEGFBS" />
             </multi:listOfSpeciesRestriction>
           </multi:productSpeciesReference>
         </listOfProducts>

         <multi:reactionRule multi:id="EGFbindingFreeSite">
           <multi:listOfConditions>
             <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeEGF"/>
             <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeEGFBS"/>
           </multi:listOfConditions>
           <multi:listOfResults>
             <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundEGF"/>
             <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundEGFBS"/>
           </multi:listOfResults>
           <kineticLaw>
             <math xmlns="http://www.w3.org/1998/Math/MathML" >
               <apply>
                 <times />
                 <ci> konegf </ci>
                 <ci> sr_freeEGF </ci>
                 <ci> sr_freeEGFBS </ci>
                 <ci> extracellular </ci>
```

```
          </apply>
        </math>
        <listOfLocalParameters>
          <localParameter id="konegf" value="10">
        </listOfLocalParameters>
      </kineticLaw>
    </multi:reactionRule>
  </reaction>
```



```
<!-- ------------------------------------------------------------------------ -->
<!-- Binding of TGF to EGF receptor.                                          -->
<!-- ------------------------------------------------------------------------ -->

    <reaction id="TGFbinding" reversible="false" fast="false">
      <listOfReactants>
        <speciesReference id="react_TGF" species="TGF" stoichiometry="1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_freeTGF"
                                      multi:speciesTypeInstance="freeTGF" />
          </multi:listOfSpeciesRestriction>
        </speciesReference>
        <speciesReference id="react_EGFBS" species="EGFBS" stoichiometry="1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_freeEGFBS"
                                      multi:speciesTypeInstance="freeEGFBS" />
          </multi:listOfSpeciesRestriction>
        </speciesReference>
      </listOfReactants>
      <listOfProducts>
        <multi:productSpeciesReference species="TGF" stoichiometry="1"
                              correspondingReactant="react_TGF">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_boundTGF"
                                      multi:speciesTypeInstance="boundTGF" />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
        <multi:productSpeciesReference species="EGFBS" stoichiometry="1"
                              correspondingReactant="react_EGFBS">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_boundEGFBS"
                                      multi:speciesTypeInstance="boundEGFBS" />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
      </listOfProducts>

      <multi:reactionRule multi:id="TGFbindingFreeSite">
        <multi:listOfConditions>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeTGF"/>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeEGFBS"/>
        </multi:listOfConditions>
        <multi:listOfResults>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundTGF"/>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundEGFBS"/>
```
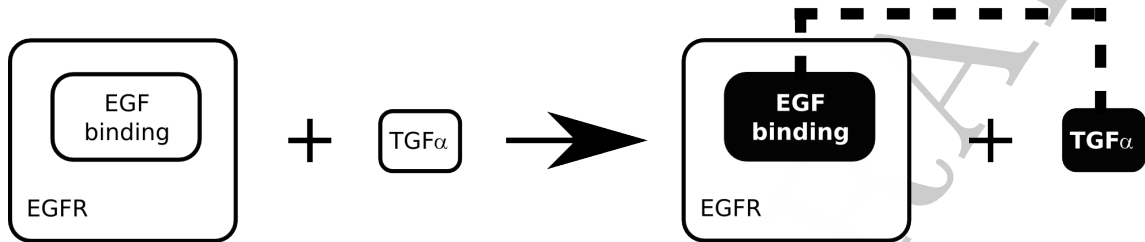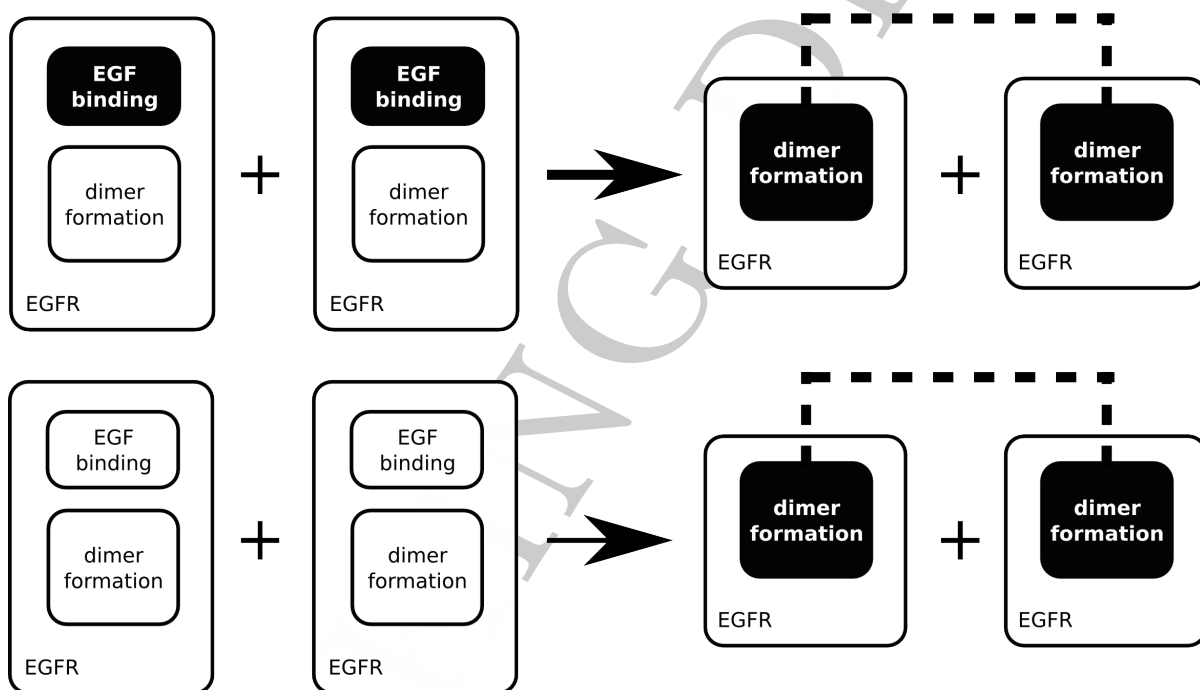
```
          </multi:listOfResults>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> kontgf </ci>
                  <ci> sr_freeTGF </ci>
                  <ci> sr_freeEGFBS </ci>
                  <ci> extracellular </ci>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="kontgf" value="5">
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>
        </reaction>
```



```
<!-- ---------------------------------------------------------------------------- -->
<!-- Dimerisation of receptor is stimulated when bound to a ligand                 -->
<!-- ---------------------------------------------------------------------------- -->

        <reaction id="dimerisation" reversible="false" fast="false">
          <listOfReactants>
            <speciesReference id="react_dimer1" species="dimer" stoichiometry="1">
              <multi:listOfSpeciesRestriction>
                <multi:speciesRestriction multi:id="sr_freeDimerNoligand1"
                                          multi:speciesTypeInstance="freeDimerNoligand" />
                <multi:speciesRestriction multi:id="sr_freeDimerLigand1"
                                          multi:speciesTypeInstance="freeDimerLigand" />
              </multi:listOfSpeciesRestriction>
            </speciesReference>
            <speciesReference id="react_dimer2" species="dimer" stoichiometry="1">
              <multi:listOfSpeciesRestriction>
                <multi:speciesRestriction multi:id="sr_freeDimerNoLigand2"
                                          multi:speciesTypeInstance="freeDimerNoligand" />
```

```
                  <multi:speciesRestriction multi:id="sr_freeDimerLigand2"
                                   multi:speciesTypeInstance="freeDimerLigand" />
            </multi:listOfSpeciesRestriction>
          </speciesReference>
      </listOfReactants>
      <listOfProducts>
        <multi:productSpeciesReference species="dimer" stoichiometry="1"
                                 correspondingReactant="react_dimer1">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_boundDimer1"
                                   multi:speciesTypeInstance="boundDimer />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
        <multi:productSpeciesReference species="dimer" stoichiometry="1"
                                 correspondingReactant="react_dimer2">
          <multi:listOfSpeciesRestriction>
            <multi:speciesRestriction multi:id="sr_boundDimer2"
                                   multi:speciesTypeInstance="boundDimer" />
          </multi:listOfSpeciesRestriction>
        </multi:productSpeciesReference>
      </listOfProducts>

      <multi:reactionRule multi:id="EGFRboundSite">
        <multi:listOfConditions>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeDimerNoligand1'
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeDimerNoligand2'
        </multi:listOfConditions>
        <multi:listOfResults>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundDimer1"/>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundDimer2"/>
        </multi:listOfResults>
          <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML" >
              <apply>
                <times />
                <ci> kdimer </ci>
                <ci> sr_freeDimer1 </ci>
                <ci> sr_freeDimer2 </ci>
                <ci> membrane </ci>
              </apply>
              <ci> kdimer </ci>
            </math>
            <listOfLocalParameters>
              <localParameter id="kdimer" value="1">
            </listOfLocalParameters>
          </kineticLaw>
        </multi:reactionRule>

      <multi:reactionRule multi:id="EGFRboundSite">
        <multi:listOfConditions>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeDimerLigand1"/>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeDimerLigand2"/>
        </multi:listOfConditions>
        <multi:listOfResults>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundDimer1"/>
          <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundDimer2"/>
        </multi:listOfResults>
          <kineticLaw>
            <math xmlns="http://www.w3.org/1998/Math/MathML" >
              <apply>
                <times />
                <ci> kdimer </ci>
                <ci> sr_freeDimer1 </ci>
                <ci> sr_freeDimer2 </ci>
                <ci> membrane </ci>
              </apply>
```
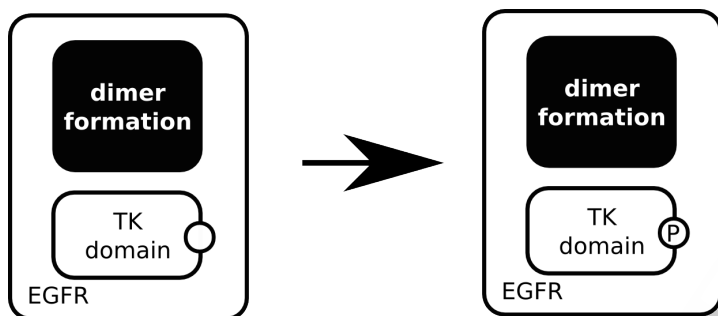
```
        <ci> kdimer </ci>
      </math>
      <listOfLocalParameters>
        <localParameter id="kdimer" value="10">
      </listOfLocalParameters>
    </kineticLaw>
  </multi:reactionRule>

</reaction>
```



```
<!-- ---------------------------------------------------------------------- -->
<!-- Phosphorylation of EGFR if dimerized                                   -->
<!-- ---------------------------------------------------------------------- -->

    <reaction id="phosphorylation" reversible="false" fast="false">
     <listOfReactants>
       <speciesReference id="react_signal" species="signal" stoichiometry="1">
         <multi:listOfSpeciesRestriction>
           <multi:speciesRestriction multi:id="sr_DimerNonPhosSignal"
                                     multi:speciesTypeInstance="DimerNonPhosSignal" />
         </multi:listOfSpeciesRestriction>
       </speciesReference>
     </listOfReactants>
     <listOfProducts>
       <multi:productSpeciesReference species="signal" stoichiometry="1">
         <multi:listOfSpeciesRestriction>
           <multi:speciesRestriction multi:id="sr_DimerPhosSignal"
                                     multi:speciesTypeInstance="DimerPhosSignal />
         </multi:listOfSpeciesRestriction>
       </multi:productSpeciesReference>
     </listOfProducts>

     <multi:reactionRule multi:id="EGFRboundSite">
       <multi:listOfConditions>
         <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_DimerNonPhosSignal'
       </multi:listOfConditions>
       <multi:listOfResults>
         <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_DimerPhosSignal"/>
       </multi:listOfResults>
       <kineticLaw>
         <math xmlns="http://www.w3.org/1998/Math/MathML" >
           <apply>
             <times />
             <ci> kcat </ci>
             <ci> sr_DimerNonPhosSignal </ci>
             <ci> intracellular </ci>
           </apply>
         </math>
         <listOfLocalParameters>
           <localParameter id="kcat" value="1">
```

```
                    </listOfLocalParameters>
                </kineticLaw>
            </multi:reactionRule>

        </reaction>
```



```
<!-- ------------------------------------------------------------------------------- -->
<!-- SHC binds to EGFR faster if the receptor is phosphorylated                      -->
<!-- ------------------------------------------------------------------------------- -->

        <reaction id="SHCbinding" reversible="false" fast="false">
         <listOfReactants>
           <speciesReference id="react_signal" species="signal" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_freePhosSignal"
                                          multi:speciesTypeInstance="freePhosSignal" />
               <multi:speciesRestriction multi:id="sr_freeNonPhosSignal"
                                          multi:speciesTypeInstance="freeNonPhosSignal" />
             </multi:listOfSpeciesRestriction>
           </speciesReference>
           <speciesReference id="react_SHC" species="SHC" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_freeSHC"
                                          multi:speciesTypeInstance="freeSHC" />
             </multi:listOfSpeciesRestriction>
           </speciesReference>
         </listOfReactants>
         <listOfProducts>
           <multi:productSpeciesReference species="signal" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_boundSignal"
                                          multi:speciesTypeInstance="boundSignal" />
             </multi:listOfSpeciesRestriction>
           </multi:productSpeciesReference>
           <multi:productSpeciesReference species="SHC" stoichiometry="1">
             <multi:listOfSpeciesRestriction>
               <multi:speciesRestriction multi:id="sr_boundSHC"
                                          multi:speciesTypeInstance="boundSHC"/>
             </multi:listOfSpeciesRestriction>
           </multi:productSpeciesReference>
```
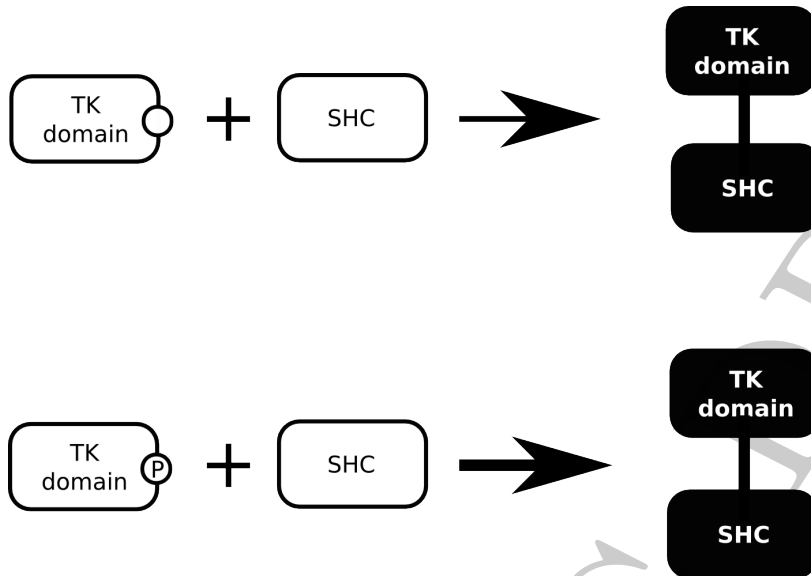
```
        </listOfProducts>

        <multi:reactionRule multi:id="rule_NonPhosSignal">
          <multi:listOfConditions>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freePhosSignal"/>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeSHC"/>
          </multi:listOfConditions>
          <multi:listOfResults>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundSignal"/>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundSHC"/>
          </multi:listOfResults>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> konNP </ci>
                  <ci> sr_freePhosSignal </ci>
                  <ci> sr_freeSHC </ci>
                  <ci> intracellular </ci>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="konNP" value="1">
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>

        <multi:reactionRule multi:id="rule_PhosSignal">
          <multi:listOfConditions>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeNonPhosSignal"/
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_freeSHC"/>
          </multi:listOfConditions>
          <multi:listOfResults>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundSignal"/>
            <multi:speciesTypeRestrictionReference multi:speciesTypeRestriction="sr_boundSHC"/>
          </multi:listOfResults>
            <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML" >
                <apply>
                  <times />
                  <ci> konP </ci>
                  <ci> sr_freeNonPhosSignal </ci>
                  <ci> sr_freeSHC </ci>
                  <ci> intracellular </ci>
                </apply>
              </math>
              <listOfLocalParameters>
                <localParameter id="konP" value="10">
              </listOfLocalParameters>
            </kineticLaw>
          </multi:reactionRule>

      </reaction>

    </listOfReactions>

  </model>
</sbml>
```