Systems Biology Markup Language (SBML) Level 3 Proposal: Array Features

Andrew Finney, Victoria Gor, Ben Bornstein, Eric Mjolsness
afinney@cds.caltech.edu, gor@aig.jpl.nasa.gov, bornstei@aig.jpl.nasa.gov
emj@uci.edu

April 30, 2003

Contents

1 Introduction

This document describes proposed features for inclusion in Systems Biology Markup Language (SBML) Level 3. This document describes features enabling the inclusion of arrays of processes, structures or entities in models. These features would allow a model to be assembled from many copies of identical parts. These features enable the representation of patterns of connection amongst array elements.

This document is not a definition of SBML Level 3 or part of it. This document simply presents various features which could be incorporated into SBML Level 3 as the Systems Biology community wishes. This document is intended for detailed review by that community and to provoke alternative proposals. Throughout this document issues that the authors believe will require further discussion have been highlighted.

For brevity the text of this document is with reference to SBML Level 2 (Finney et al., 2002) and a model composition proposal (Finney, 2003) i.e. features are described in terms of changes to SBML Level 2 combined with the model composition proposal. All the changes proposed in this document are shown in the UML diagram in figure 1. For brevity this diagram shows only new attributes and types for SBML Level 3. For SBML Level 2 types Level 2 attributes are meant to be present in SBML Level 3. All the types and attributes proposed in the model composition are assumed to be present. All types proposed in this document will be derived from the `SBase` type.

2 Integer Parameters

A model would have an optional list of `IntegerParameter` structures. The `IntegerParameter` type is described in section 2.

`IntegerParameter` structures declare integer constants with identifiers in the model's global identifier namespace. They can be used in any MathML expression where they have an implicit type of `integer`. They are distinct from `Parameter` structures: it is not possible to define a rule for an `IntegerParameter`. Integer parameters are constant.

3 Constant Expressions

A constant expression is an expression whose result doesn't vary during simulation. The only symbols occurring in constant expressions are those representing integer variables (declared by `Dimension` structures) or integer parameters.

4 Integer Expressions

In some cases in the proposal a MathML expression is restricted to returning an integer result. An integer expression is composed of integer numbers, symbols representing integer variables (declared by `Dimension` structures) or integer parameters and operators that return integer values given integer arguments. Integer expressions are required in the `upperLimit`, `lowerLimit` and `index` fields in this proposal. In addition the arguments for the MathML `selector` operator should be integer expressions.

Because none of the symbols in integer expressions vary at simulation time an integer expression can be considered constant.

5 Conditional Objects

In this proposal `Species`, `Parameter`, `Link`, `Instance`, `Rule`, `Reaction` and `Compartment` objects all have a new `exists` field. This field contains a MathML expression which returns a boolean result. If this expression is true the containing object is part of the model. If the expression is false the object does not exist. Variables acquire their initial values whenever they come into existence.

As an example the following fragment shows a species that only exists when the parameter `x` is greater than `size`.

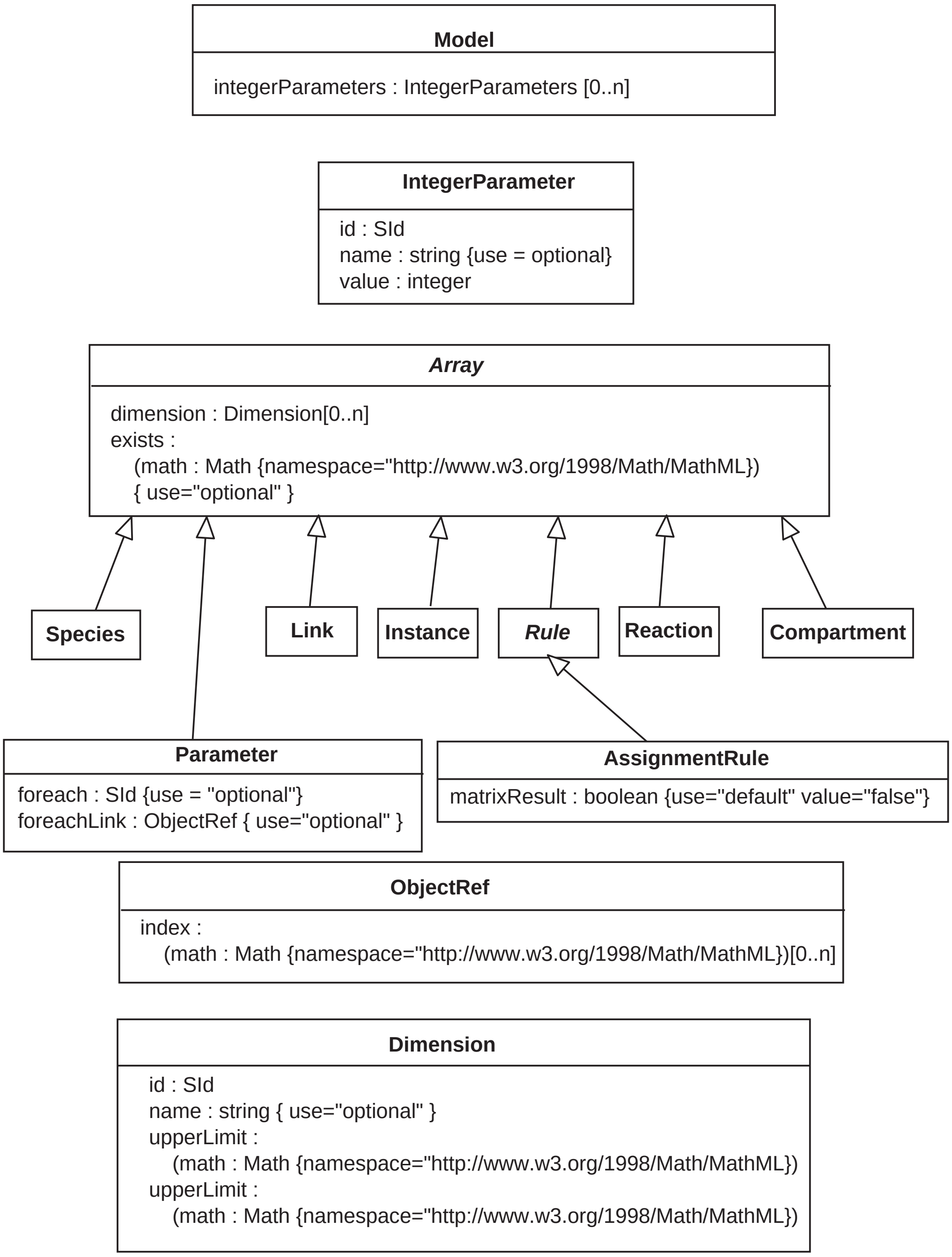
Model
integerParameters : IntegerParameters [0..n]
IntegerParameter
id : SId
name : string {use = optional}
value : integer
Array
dimension : Dimension[0..n]
exists :
(math : Math {namespace="http://www.w3.org/1998/Math/MathML})
{ use="optional" }
Species
Link
Instance
Rule
Reaction
Compartment
Parameter
foreach : SId {use = "optional"}
foreachLink : ObjectRef { use="optional" }
AssignmentRule
matrixResult : boolean {use="default" value="false"}
ObjectRef
index :
(math : Math {namespace="http://www.w3.org/1998/Math/MathML})[0..n]
Dimension
id : SId
name : string { use="optional" }
upperLimit :
(math : Math {namespace="http://www.w3.org/1998/Math/MathML})
upperLimit :
(math : Math {namespace="http://www.w3.org/1998/Math/MathML})

Figure 1: *proposed structures and proposed changes to existing SBML Level 2 and model composition structures*

```
<species id="offspring" initialAmount="0">
    <exists>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
                <gt/>
                <ci>x</ci>
                <ci>size</ci>
            </apply>
        </math>
    </exists>
```

```
</species>
```

A model which contains non-constant expressions in `exists` fields is *dynamic* i.e the model structure can change during simulation. A model which contains no `exists` fields or contains constant expressions in `exists` fields is *static* i.e the model structure cannot change during simulation.

This feature is independent of the rest of the features described in this proposal. As will be shown in section 9.1 there is a class of static models which use `exists` attributes as a mechanism for describing sparse arrays.

6 Arrays

The core of this proposal is the idea that almost all the structures in SBML can be defined as arrays as well as single named objects. We propose that the SBML types `Instance`, `Link`, `Species`, `Compartment`, `Reaction`, `Parameter` and `Rule` can be defined as arrays of objects.

6.1 Array Declaration

The presence of a `dimension` field (`listOfDimensions` sub-element) indicates that the given structure is an array rather than a single object. Each `Dimension` structure in this field defines the array cells in one dimension of the array.

A dimension structure declares an integer symbol that takes integer values over a defined range. Array cells exists on the given dimension for each value of the integer symbol. The range of values is defined by the inclusive range between the results of the `upperLimit` and `lowerLimit` MathML expressions. These expressions must return an integer result. The integer symbol iteratively takes on the value in the range and has scope within the other MathML expression contained by the SBML component element except for `Dimension` structures preceding the given `Dimension` structure in `listOfDimensions`. The integer symbol will overload symbols declared elsewhere in the model.

All the objects in the array have the properties described by the structures's attributes and substructures. Structures that are declared as arrays share the same namespace as those structures that represent single objects.

The following SBML model shows a `Compartment` structure representing a 1 dimensional array consisting of a cells located in the array from a positions 0 to 9 inclusive.

```
<model id="simple">
    <listOfCompartments>
        <compartment id="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </compartment>
    </listOfCompartments>
</model>
```

6.2 Array Element Object Reference

Elements of an array can be referenced from objects using ObjectRef structures introduced by the model composition proposal. The ObjectRef type is extended in this proposal to have an index field (the XML encoding for this field is a listOfIndices element). This field consists of a set of math expressions which index into the array referenced by the object field. There should be as many expressions as there dimensions in the referenced array.

We can use the compartmentLink field on a Species structure to create an array of species distributed across an array of compartments:

```
<model id="ref">
    <listOfCompartments>
        <compartment id="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </compartment>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <compartmentLink object="cell">
                <listOfIndices>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <ci> x </ci>
                    </math>
                </listOfIndices>
            </compartmentLink>
        </species>
    </listOfSpecies>
</model>
```

In this example each species array element is placed in a corresponding compartment array element.

6.3 Array Element Reference In MathML

This section describes mechanisms within a a math expression, for accessing specific elements of an array.

In MathML expressions components declared as arrays can be referenced using ci elements. However this will represent an matrix or vector of the values of the component's elements as opposed to a single scalar value. In MathML expressions the selector operator can be applied to such an array symbol (vector or matrix in MathML) and returns the appropriate value of the indexed array element. The ‘array’ operand of a selector operator must be the name of a structure which has been declared as an array. The index arguments must integer expressions.

6.3.1 Example of using element references in both numeric and object reference fields

The following example fragment shows how the array operator can be used in a numeric expression.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
        <times/>
        <apply>
            <selector>
            <ci> s1 </ci>
            <ci> x </ci>
        </apply>
        <cn> 0.1 </cn>
    </apply>
</math>
```

s1 is an array and x is an integer symbol or parameter. The whole model follows. This model contains 2 arrays of 10 species and defines a set of 10 reactions between corresponding pairs of species. The rate equation for this set of reactions uses the selector element to reference the reactant species.

```
<model id="ref">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s1" compartment="cell"/>
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
        <species id="s2" compartment="cell"/>
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfReactions>
        <reaction id="r">
            <listOfDimensions>
                <dimension id="x">
```

```
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <listOfReactants>
                <speciesReference>
                    <speciesLink object="s1">
                        <listOfIndices>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> x </ci>
                            </math>
                        </listOfIndices>
                    </speciesLink>
                </speciesReference>
            </listOfReactants>
            <listOfProducts>
                <speciesReference>
                    <speciesLink object="s2">
                        <listOfIndices>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> x </ci>
                            </math>
                        </listOfIndices>
                    </speciesLink>
                </speciesReference>
             </listofProducts>
            <kineticLaw>
                <note>
                    <p xmlns="http://www.w3.org/1999/xhtml">s1[x] * 0.1</p>
                </note>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <times/>
                        <apply>
                            <selector>
                            <ci> s1 </ci>
                            <ci> x </ci>
                        </apply>
                        <cn> 0.1 </cn>
                    </apply>
                </math>
            <kineticLaw>
        </reaction>
    </listOfReactions>
</model>
```

6.3.2 Use of csymbol for accessing high dimensional arrays

Unfortunately MathML is not designed to represent matrices with more than 2 dimensions. So that it is possible to access elements of arrays with more than 3 dimensions we define a new URI for use with the MathML `csymbol` element which represents a selector operator that takes an arbitrary number of arguments. The URI is `http://www.sbml.org/symbols/arrayselector`. This operator can be used in place of the `selector` element.

The following example fragment shows how the array operator can be used in a numeric expression.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply>
        <times/>
        <apply>
```

```
                <csymbol
                    encoding="SBML"
                    definitionURL="http://www.sbml.org/symbols/arrayselector">
                    ?
                </csymbol>
                <ci> s1 </ci>
                <ci> x </ci>
                <ci> y </ci>
                <ci> z </ci>
            </apply>
            <cn> 0.1 </cn>
        </apply>
    </math>
```

6.3.3 Issue

Do we want to support these two operators simultaneously?

7 Simplified Array Structures for Related Objects

It is possible to incorporate a simplified mechanism for creating species, compartment, rule and parameter arrays and for referencing the elements of those arrays.

7.1 Implied Species Arrays

In this proposal when the `compartment` field of a `Species` structure contains the identifier of an array of compartments the `compartmentLink` field has to be omitted. A structure configured in this fashion represents an array of species with the same specification as the given compartment array.

Alternatively the `index` field enclosed in the `compartmentLink` field may be missing. Again a structure configured in this fashion represents an array of species with the same specification as the given compartment array.

In these cases each element of the species array is located in a corresponding compartment element. The species array is not explicitly declared i.e. the `dimension` field (`listOfDimensions` element) has alternative semantics. For the moment in our examples it is omitted as well.

For example given

```
<compartment id="cell">
    <listOfDimensions>
         <dimension id="x">
             <lowerLimit>
                 <math xmlns="http://www.w3.org/1998/Math/MathML">
                     <cn> 0 </cn>
                 </math>
             </lowerLimit>
             <upperLimit>
                 <math xmlns="http://www.w3.org/1998/Math/MathML">
                     <cn> 9 </cn>
                 </math>
             </upperLimit>
         </dimension>
    </listOfDimensions>
</compartment>
```

the structure

```
<species id="s" initialAmount="0">
    <listOfDimensions>
        <dimension id="x">
            <lowerLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
```

```
                    <cn> 0 </cn>
                </math>
            <lowerLimit>
            <upperLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 9 </cn>
                </math>
            <upperLimit>
        </dimension>
    </listOfDimensions>
    <compartmentLink object="cell">
        <listOfIndices>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <ci> x </ci>
            </math>
        </listOfIndices>
    </compartmentLink>
</species>
```

can be replaced with the equivalent structures:

```
<species id="s" compartment="cell" initialAmount="0"/>
```

or

```
<species id="s" initialAmount="0">
    <compartmentLink object="cell"/>
</species>
```

If the species `dimension` field (`listOfDimensions` sub-element) is present but the `compartmentLink` field is not present or the enclosed `index` field is not present the resulting species array created has the combined dimensions of the compartment array and the dimensions enclosed in the `dimension` field. The compartment dimensions are implied in the sequence before the species dimensions. For example the following structure defines a 2 dimension array of species where the first dimension is mapped across a row of compartments:

```
<compartment id="cell"/>
    <listOfDimensions>
        <dimension id="x">
            <lowerLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 0 </cn>
                </math>
            </lowerLimit>
            <upperLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 9 </cn>
                </math>
            </upperLimit>
        </dimension>
    </listOfDimensions>
</compartment>
...
<species id="ss" compartment="cell" initialAmount="0"/>
    <listOfDimensions>
        <dimension id="y">
            <lowerLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 0 </cn>
                </math>
            </lowerLimit>
            <upperLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 5 </cn>
                </math>
            </upperLimit>
        </dimension>
    </listOfDimensions>
</species>
```

In these cases the symbols declared in the referenced compartment's `dimension` field (`listOfDimensions` sub-element) are in scope in all the MathML expressions contained within the species structure. For example a species structure could omit the `compartmentLink` field yet still use symbols from the compartment's `dimension` field in the species' `exists` field (see section 9.1 for more details on this field).

7.2 Implied Compartment Arrays

In SBML Level 2 it is possible to specify nested compartments. For example the fragment:

```
<listOfCompartments>
    <compartment id="a">
    <compartment id="b" outside="a">
</listOfCompartments>
```

Defines a compartment `b` which is enclosed by `a`.

The scheme used for species in the previous section can be applied in this case. For example

```
<listOfCompartments>
    <compartment id="a">
        <listOfDimensions>
            <dimension id="x">
                <lowerLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 0 </cn>
                    </math>
                </lowerLimit>
                <upperLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 5 </cn>
                    </math>
                </upperLimit>
            </dimension>
        </listOfDimensions>
    </compartment>
    <compartment id="b" outside="a"/>
</listOfCompartments>
```

is equivalent to

```
<listOfCompartments>
    <compartment id="a"/>
        <listOfDimensions>
            <dimension id="x">
                <lowerLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 0 </cn>
                    </math>
                </lowerLimit>
                <upperLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 5 </cn>
                    </math>
                </upperLimit>
            </dimension>
        </listOfDimensions>
    </compartment>
    <compartment id="b">
        <listOfDimensions>
            <dimension id="x">
                <lowerLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 0 </cn>
                    </math>
                </lowerLimit>
                <upperLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 5 </cn>
                    </math>
                </upperLimit>
            </dimension>
        </listOfDimensions>
        <outsideLink object="a">
            <listOfIndices>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> x </ci>
                </math>
            </listOfIndices>
        </outsideLink>
    </compartment>
</listOfCompartments>
```

Similar scoping rules for symbols apply here as they do for species structures.

7.3 Implied Parameter Arrays

We can apply the above concept to parameters if we introduce a new `SId` field, `foreach` and corresponding `ObjectLink` field, `foreachLink`, to the parameter structure. This field allows us to reference any other symbol from this structure and thus attach a parameter to each element of the referenced array.

For example given:

```
<compartment id="cell">
    <listOfDimensions>
        <dimension id="x">
            <lowerLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 0 </cn>
                </math>
            </lowerLimit>
            <upperLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 5 </cn>
                </math>
            </upperLimit>
        </dimension>
    </listOfDimensions>
</compartment>
...
```

then

```
<parameter id="p" foreach="cell" value="0"/>
```

is equivalent to

```
<parameter id="p" value="0">
    <listOfDimensions>
        <dimension id="x">
            <lowerLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 0 </cn>
                </math>
            </lowerLimit>
            <upperLimit>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <cn> 5 </cn>
                </math>
            </upperLimit>
        </dimension>
    </listOfDimensions>
</parameter>
```

Similar scoping rules for symbols apply here as they do for species structures.

8 Arrays of Assignment Rules

Declaring arrays of assignment rules is slightly different from other component types. Rule structures don't contain an `id` field that declares a new symbol instead they reference another structure via the `variable` field. The expressions contained in the `dimension` field (`listOfDimensions` sub-element) operate in the same way as described above except that the assignment rule array cells constrain corresponding cells with same indices in the array referenced by the `variable` field. For example the following model has a rule applied to an array of species.

```
<model id="rules">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s" initialAmount="0.1" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule variable="s" type="rate">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0.1 </cn>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

As rules do not declare symbols it is possible for more than one rule to be applied to the same array. For example consider the following example in which two different rules are applied to different halves of the same array:

```
<model id="rules">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s" initialAmount="0.1" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule variable="s" type="rate">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 4 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0.1 </cn>
            </math>
        </assignmentRule>
        <assignmentRule variable="s" type="rate">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 5 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0.2 </cn>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

8.1 Implied Arrays of Rules

If the the dimension field (listOfDimensions sub-element) is omitted and if the variable reference by the assignment rule is an array then the assignment rule implicitly consists of an array with the same dimensions as the reference variable object. For example the example

```
<model id="rules">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s" initialAmount="0.1" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule variable="s" type="rate">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0.1 </cn>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

is equivalent to:

```
<model id="rules">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s" initialAmount="0.1" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule variable="s" type="rate">
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0.1 </cn>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

8.2 Rules which supply a matrix result

Assignment Rules can be defined which apply to the whole of an array in which the formula expression returns a whole array value rather than a single value to be inserted into each array element. This type of rule is indicated by the `matrixResult` attribute having a `true` value (default is `false`). The resulting array should have the same dimensions as the assignment variable for which it is declared. The result of the expression is then mapped onto the array referenced in the rules `variable` field. This type of rule must not have any dimension

Whole matrix operators are described in more detail in section 10. For now assume that we can, for instance, multiply arrays then consider the example model:

```
<model id="rules">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s1" initialAmount="0.1" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    <lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    <upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
        <species id="s2" initialAmount="0.2" compartment="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    <lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    <upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule matrixResult="true" variable="s1" type="rate"/>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                    <plus/>
                    <ci> s1 </ci>
                    <ci> s2 </ci>
                </apply>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

The rule for `s1` defines that the rate of change of the values of the array of `s1` is the product of the concentration vectors `s1` and `s2`.

8.3 Initial Assignment Rules

One the limitations of the previously described structures is that there is no way to specify that initial conditions vary across elements of an array. We describe a new feature in this section to overcome this limitation.

In this proposal we introduce a new type of `AssignmentRule` *initialScalar* in addition to the existing *scalar* and *rate* types. (These are values of the `type` field.) *initialScalar* and *scalar* rules are evaluated at the beginning of a simulation to establish initial values for variables. However *scalar* rules are evaluated throughout the simulation where as *initialScalar* rules are not. The combined *scalar* and *initialScalar* rule set must not contain algebraic loops. The behavior of variables assigned values by an *initialScalar* rule can be determined either by a reaction or a *rate* rule (this is not true of *scalar* rules).

This feature could be considered separately from the rest of the arrays proposal.

For example a model which wished to vary the initial concentration of a species across a 1 dimensional array of species would be encoded as follows:

```
<model id="varying">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartment>
    <listOfSpecies>
        <species id="s">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    <lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    <upperLimit>
                </dimension>
            </listOfDimensions>
        </species>
    </listOfSpecies>
    <listOfRules>
        <assignmentRule type="initialScalar" variable="s">
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <apply>
                    <times/>
                    <ci> x </ci>
                    <cn> 0.1 </cn>
                </apply>
            </math>
        </assignmentRule>
    </listOfRules>
</model>
```

9 Sparse Arrays and Connections

9.1 Sparse Arrays

In practice arrays are not very useful for modelling unless its possible to describe connection schemes between elements of the arrays. For example if one creates a model of a tissue of cells as an array of compartments then the model doesn't become interesting until the interactions between the cells are incorporated. This section begins the process of proposing structures which allow interconnection schemes to be defined.

We can use the `exists` field (see section 5 as a mechanism for creating sparse arrays. Elements of an array are simply absent if the `exists` expression is false for a given set of dimension integer variables.

9.2 Sparse Array Example

The following example shows how the `exists` field is used to create a structure for a triangular array where the maximum `y` index is equal to the `x` index.

```
<model id="starfish">
    <listOfCompartments>
        <compartment id="cell">
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="y">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <exists>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <leq/>
                        <ci> x </ci>
                        <ci> y </ci>
                    </apply>
                </math>
            </exists>
        </compartment>
    </listOfCompartments>
</model>
```

This can be defined in an alternative fashion by using the `x` symbol to define the bounds of the `y` dimension:

```
<model id="starfish">
    <listOfCompartments>
        <compartment id="cell">
            <listOfDimensions>
                <dimension id="x">
```

```
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="y">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <ci> x </ci>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </compartment>
    </listOfCompartments>
</model>
```

The index operand to an `selector` operator must refer to an element which exists in an array.

9.3 Using Sparse Arrays to Represent Connection schemes

We can use a proposed sparse array feature to represent the connections between elements of another array. If we have a n dimensional array of components we can represent connections between those components as a $2n$ dimensional sparse array. In the sparse array elements only occur where connections are to be modelled. Thus components that exist as connections are thus defined as sparse $2n$ dimensional arrays.

This is shown in the following example, in which `grid` is a 2 dimensional array of compartments, `s` is a 2 dimensional array of species distributed over the `grid`compartments and, `connections` is a sparse 4 dimensional array of reactions between elements of `s`. `connections` contains array elements for all pairs of co-ordinates where the co-ordinates are exactly one array element away from each other.

```
<model id="tissue">
    <listOfCompartments>
        <compartment id="grid"/>
            <listOfDimensions>
                <dimension id="x">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="y">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
        </compartment>
    </listOfCompartments>
    <listOfSpecies>
        <species id="s" initialAmount="0.1" compartment="grid">
    </listOfSpecies>
    <listOfReactions>
        <reaction id="connections">
            <listOfDimensions>
                <dimension id="x1">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="y1">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    <lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <ci> 9 </ci>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="x2">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    <lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 9 </cn>
                        </math>
                    </upperLimit>
                </dimension>
                <dimension id="y2">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <ci> 9 </ci>
                        </math>
                    </upperLimit>
                </dimension>
            </listOfDimensions>
            <exists>
                <note>
                    <p xmlns="http://www.w3.org/1999/xhtml">
                        abs(x2 - x1) == 1 || abs(y2 - y1) == 1
                    </p>
                </note>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <or/>
                        <apply>
                            <eq/>
                            <apply/>
                                <abs/>
                                <apply>
                                    <minus/>
                                    <ci> x2 </ci>
                                    <ci> x1 </ci>
                                </apply>
                            </apply>
                            <cn> 1 </cn>
                        </apply>
                        <apply>
                            <eq/>
                            <apply/>
                                <abs/>
                                <apply>
                                    <minus/>
                                    <ci> y2 </ci>
                                    <ci> y1 </ci>
                                </apply>
                            </apply>
                            <cn> 1 </cn>
                        </apply>
                    </apply>
                </math>
            </exists>
            <listOfReactants>
                <speciesReference>
                    <speciesLink object="s">
                        <listOfIndices>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> x1 </ci>
                            </math>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> y1 </ci>
                            </math>
                        </listOfIndices>
                    </speciesLink>
                </speciesReference>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "s">
                    <speciesLink object="s">
                        <listOfIndices>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> x2 </ci>
                            </math>
                            <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> y2 </ci>
                            </math>
                        </listOfIndices>
                    </speciesLink>
                </speciesReference>
            </listOfProducts>
            <kineticLaw>
                <note>
                    <p xmlns="http://www.w3.org/1999/xhtml">s[x1][y1] * 0.1"</p>
                </note>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <times/>
                        <apply>
                            <selector/>
                            <ci> s </ci>
                            <ci> x1 </ci>
                            <ci> y1 </ci>
```

```
                        </apply>
                        <cn> 0.1 </cn>
                    </apply>
                </math>
            </kineticLaw
        </reaction>
    </listOfReactions>
</model>
```

The `connections` specification is bidirectional: for every pair of adjacent `grid` co-ordinates there are a pair of elements. `connections` can be simplified and made unidirectional by changing the `exists` expression to:

```
x2 - x1 == 1 || y2 - y1 == 1
```

In this case the connections only run from bottom to top and left to right.

10 Array Math

Under this proposal the set of MathML elements that can be incorporated into SBML is extended to include the following:

- *constructors* `matrix`, `matrixrow`, `vector`
- *element reference operator* `selector`
- *qualifier components* `bvar`, `lowlimit`, `uplimit`, `interval`, `condition`
- *linear algebra operators* `vectorproduct`, `scalarproduct`, `outerproduct`, `transpose`
- *sum product operators* `sum`, `product`
- *quantifier operators* `forall`, `exists`

Please refer to the MathML specification (W3C, 2000) for further details of the operation of these elements.

11 Example: the community effect in developmental gene regulation

This section contains an example model of the community effect in developmental gene regulation (Gurdon, 1988). This model demonstrates how the array proposal can be combined with the model composition proposal.

In the SBML given below the original model is broken into two parts using model composition. The first SBML model encodes the biochemical network inside each cell (this model doesn't use any array features). The second SBML model assembles instances of the first model into a tissue using the proposed array features.

The original model basically contains a high abstracted model of gene expression for a single gene as follows:

$$\begin{array}{ccc}
ubiq & \xrightarrow{v_{mu} ubiq} & waste \\
mRNA & \xrightarrow{k_{dm} mRNA} & waste \\
p & \xrightarrow{k_{dp} p} & waste \\
mat & \xrightarrow{v_i} & ubiq \\
mat & \xrightarrow{\frac{k_t ubiq}{k_{mu}+ubiq}} & mRNA \\
mat & \xrightarrow{\frac{k_s mRNA}{k_{mu}+mRNA}} & p
\end{array}$$

where *ubiq* is a transcription factor, *p* is the gene product, *mat* represents the material used to construct active species and *waste* represents the material produced by the degradation of active species. Both *mat* and *waste* are modelled as boundary conditions.

The second model creates a “tissue” of these cells: a rectangular array of cells. This model contains a positive feedback loop between adjacent cells by enabling the gene product of an adjacent cell to be a transcription factor of the gene in the current cell. Thus the feedback loop is completed by the following reaction in the second model between adjacent cells.

$$mat_a \xrightarrow{\frac{k_r p_b}{k_m r + p_b}} mRNA_a$$

where mat_a and $mRNA_a$ are concentrations in a given cell and p_b is the contribution of gene product from an adjacent cell.

The first model:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3" version="1" level="3">
<model id="cell_model">
    <listOfCompartments>
        <compartment id="cell"/>
    </listOfCompartments>
    <listOfSpecies>
        <species id="mat" compartment="cell_compartment" boundaryCondition="true"
            initialAmount="1.0"/>
        <species id="mRNA" compartment="cell_compartment" initialAmount="0"/>
        <species id="waste" compartment="cell" boundaryCondition="true" initialAmount="1.0"/>
        <species id="ubiq" compartment="cell" initialAmount="0"/>
        <species id="p" compartment="cell" initialAmount="0"/>
    </listOfSpecies>
    <listOfParameters>
        <parameter id="kmu" value="0.1"/>
    </listOfParameters>
    <listOfReactions>
        <reaction id="ubiq2waste">
            <listOfReactants>
                <speciesReference species= "ubiq"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "waste"/>
            </listOfProducts>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <times/>
                        <ci> vmu </ci>
                        <ci> ubiq </ci>
                    </apply>
                </math>
                <listOfParameters>
                    <parameter id="vmu" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
        <reaction id="mRNA2waste">
            <listOfReactants>
                <speciesReference species= "mRNA"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "waste"/>
            </listOfProducts>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <times/>
                        <ci> kdm </ci>
                        <ci> mRNA </ci>
                    </apply>
                </math>
                <listOfParameters>
                    <parameter id="kdm" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
        <reaction id="p2waste">
            <listOfReactants>
                <speciesReference species= "p"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "waste"/>
            </listOfProducts>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <times/>
                        <ci> kdp </ci>
                        <ci> p </ci>
                    </apply>
                </math>
                <listOfParameters>
                    <parameter id="kdp" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
        <reaction id="mat2ubiq">
            <listOfReactants>
                <speciesReference species= "mat"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "ubiq"/>
            </listOfProducts>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> vi </ci>
                </math>
                <listOfParameters>
                    <parameter id="vi" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
        <reaction id="mat2p">
            <listOfReactants>
                <speciesReference species= "mat"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "p"/>
            </listOfProducts>
            <listOfModifiers>
                <modifierSpeciesReference species="mRNA"/>
            </listOfModifiers>
            <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <divide/>
                        <apply>
                            <times/>
                            <ci>ks</ci>
                            <ci>mRNA</ci>
                        </apply>
                        <apply>
                            <plus/>
                            <ci>kmu</ci>
                            <ci>mRNA</ci>
                        </apply>
```

```
                    </apply>
                </math>
                <listOfParameters>
                    <parameter id="ks" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
        <reaction id="local_mat2mRNA">
            <listOfReactants>
                <speciesReference species= "mat"/>
            </listOfReactants>
            <listOfProducts>
                <speciesReference species= "mRNA"/>
            </listOfProducts>
             <listOfModifiers>
                <modifierSpeciesReference species="ubiq"/>
            </listOfModifiers>
           <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <apply>
                        <divide/>
                        <apply>
                            <times/>
                            <ci>kt</ci>
                            <ci>ubiq</ci>
                        </apply>
                        <apply>
                            <plus/>
                            <ci>kmu</ci>
                            <ci>ubiq</ci>
                        </apply>
                    </apply>
                </math>
                <listOfParameters>
                    <parameter id="kt" value="0.1"/>
                </listOfParameters>
            </kineticLaw>
        </reaction>
    </listOfReactions>
    <listOfPorts>
        <port object="mat"/>
        <port object="mRNA"/>
        <port object="p"/>
    </listOfPorts>
</model>
</sbml>
```

The second model:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3" version="1" level="3">
<model id="community_effect">
    <listOfIntegerParameters>
        <integerParameter id="xBound" value = "9"/>
        <integerParameter id="yBound" value = "5"/>
    </listOfIntegerParameters>
    <listOfReactions>
        <reaction id="neighbour_mat2mRNA">
            <listOfDimensions>
                <dimension id="x1">
                    <lowerLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <cn> 0 </cn>
                        </math>
                    </lowerLimit>
                    <upperLimit>
                        <math xmlns="http://www.w3.org/1998/Math/MathML">
                            <ci> xBound </ci>
                        </math>
        </upperLimit>
    </dimension>
    <dimension id="y1">
        <lowerLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0 </cn>
            </math>
        </lowerLimit>
        <upperLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <ci> yBound </ci>
            </math>
        </upperLimit>
    </dimension>
    <dimension id="x2">
        <lowerLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0 </cn>
            </math>
        </lowerLimit>
        <upperLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <ci> xBound </ci>
            </math>
        </upperLimit>
    </dimension>
    <dimension id="y2">
        <lowerLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <cn> 0 </cn>
            </math>
        </lowerLimit>
        <upperLimit>
            <math xmlns="http://www.w3.org/1998/Math/MathML">
                <ci> yBound </ci>
            </math>
        </upperLimit>
    </dimension>
</listOfDimensions>
<exists>
    <note>
        <p xmlns="http://www.w3.org/1999/xhtml">
            (x1 == x2 or y1 == y2) and (x1 != x2 or y1 != y2)
        </p>
    </note>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
            <and/>
            <apply>
                <or/>
                <apply>
                    <eq/>
                    <ci>x1</ci>
                    <ci>x2</ci>
                </apply>
                <apply>
                    <eq/>
                    <ci>y1</ci>
                    <ci>y2</ci>
                </apply>
            </apply>
            <apply>
                <or/>
                <apply>
                    <neq/>
                    <ci>x1</ci>
                    <ci>x2</ci>
                </apply>
                <apply>
```

```
                    <neq/>
                    <ci>y1</ci>
                    <ci>y2</ci>
                </apply>
            </apply>
        </apply>
    </math>
</exists>
<listOfReactants>
    <speciesReference>
        <speciesLink object="cell">
            <objectLink object="mat"/>
            <listOfIndices>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> x1 </ci>
                </math>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> y1 </ci>
                </math>
            </listOfIndices>
        </speciesLink>
    </speciesReference>
</listOfReactants>
<listOfProducts>
    <speciesReference>
        <speciesLink object="cell">
            <objectLink object="mRNA"/>
            <listOfIndices>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> x1 </ci>
                </math>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> y1 </ci>
                </math>
            </listOfIndices>
        </speciesLink>
    </speciesReference>
</listOfProducts>
<listOfModifiers>
    <modifierSpeciesReference>
        <speciesLink object="cell">
            <objectLink object="p"/>
            <listOfIndices>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> x2 </ci>
                </math>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> y2 </ci>
                </math>
            </listOfIndices>
        </speciesLink>
    </modifierSpeciesReference>
</listOfModifiers>
<kineticLaw>
    <note>
        <p xmlns="http://www.w3.org/1999/xhtml">
            (kr * cell[x2][y2].p)/(kmr + cell[x2][y2].p)
        </p>
    </note>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
            <divide/>
            <apply>
                <times/>
                <ci>kr</ci>
                <apply>
                    <csymbol
                        encoding="SBML"
                        definitionURL=
                                    "http://www.sbml.org/symbols/instanceselector">
                                .
                            </csymbol>
                            <apply>
                                <selector/>
                                <ci> cell </ci>
                                <ci> x2 </ci>
                                <ci> y2 </ci>
                            </apply>
                            <ci>p</ci>
                        </apply>
                    </apply>
                    <apply>
                        <add/>
                        <ci>kmr</ci>
                        <apply>
                            <csymbol
                                encoding="SBML"
                                definitionURL=
                                    "http://www.sbml.org/symbols/instanceselector">
                                .
                            </csymbol>
                            <apply>
                                <selector/>
                                <ci> cell </ci>
                                <ci> x2 </ci>
                                <ci> y2 </ci>
                            </apply>
                            <ci>p</ci>
                        </apply>
                    </apply>
                </apply>
            </math>
            <listOfParameters>
                <parameter id="kr" value="0.1"/>
                <parameter id="kmr" value="0.07"/>
            </listOfParameters>
        </kineticLaw>
    </reaction>
</listOfReactions>
<listOfInstances>
    <instance id="cell" xlink:type="simple" xlink:href="cellModel.xml">
        <listOfDimensions>
            <dimension id="x">
                <lowerLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 0 </cn>
                    </math>
                </lowerLimit>
                <upperLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <ci> xBound </ci>
                    </math>
                </upperLimit>
            </dimension>
            <dimension id="y">
                <lowerLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <cn> 0 </cn>
                    </math>
                </lowerLimit>
                <upperLimit>
                    <math xmlns="http://www.w3.org/1998/Math/MathML">
                        <ci> yBound </ci>
                    </math>
                </upperLimit>
            </dimension>
        </listOfDimensions>
    </instance>
    </listOfInstances>
  </model>
  </sbml>
```

12 Acknowledgements

Many thanks to Hamid Bolouri for suggesting the community effect model as an example model.

References

Finney, A. (2003). Systems Biology Markup Language (SBML) Level 3 Proposal: Model composition features.

Finney, A., Hucka, M., and Bolouri, H. (2002). Systems Biology Markup Language (SBML) Level 2: Structures and facilities for model definitions. Available via the World Wide Web at `http://www.sbw-sbml.org/sbml/docs/sbml-development/level-2/`.

Gurdon, J. B. (1988). A community effect in animal development. *Nature*, 336(6201):772–774.

W3C (2000). W3C's math home page. Available via the World Wide Web at `http://www.w3.org/Math/`.