
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 11, 2003

Contents

1	Introduction	2
2	Models	2
3	Integer Parameters	2
4	Constant Expressions	2
5	Integer Expressions	3
6	Arrays	3
6.1	Array Declaration	3
6.2	Array Element Object Reference	4
6.3	Array Element Reference In MathML	5
7	Simplified Array Structures for Related Objects	7
7.1	Implied Species Arrays	7
7.2	Implied Compartment Arrays	9
7.3	Implied Parameter Arrays	10
8	Arrays of Assignment Rules	11
8.1	Implied Arrays of Rules	12
8.2	Rules which supply a matrix result	13
9	Sparse Arrays and Connections	14
9.1	Sparse Arrays	14
9.2	Sparse Array Example	15
9.3	Using Sparse Arrays to Represent Connection schemes	16
10	Array Math	19
11	Example: the community effect in developmental gene regulation	19
12	Acknowledgements	27
	References	27

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) i.e. features are described in terms of changes to SBML Level 2. This document also refers to another Level 3 proposal on model composition (Finney, 2003). In addition for brevity the UML diagrams in this proposal show 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 Models

The proposed structure of the **Model** type is shown in figure 1. A model would have an optional list of **IntegerParameter** structures.

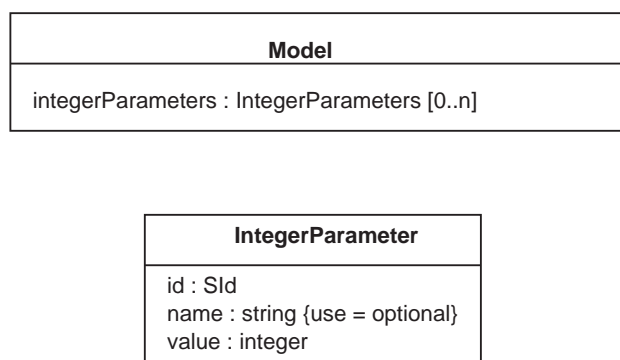


Figure 1: *The definition of the Model and IntegerParameter types*

The **IntegerParameter** type is described in section 3.

3 Integer Parameters

The **IntegerParameter** type is shown in figure 1. **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**.

4 Constant Expressions

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

5 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 **dimension** 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.

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 following SBML types, **Instance**, **Link**, **Species**, **Compartment**, **Reaction**, **Parameter** and **Rule** can be defined as arrays of objects. Figure 2 summarizes the changes made to these structures.

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 exist 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**.

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 10 cells located in the array from 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>
```

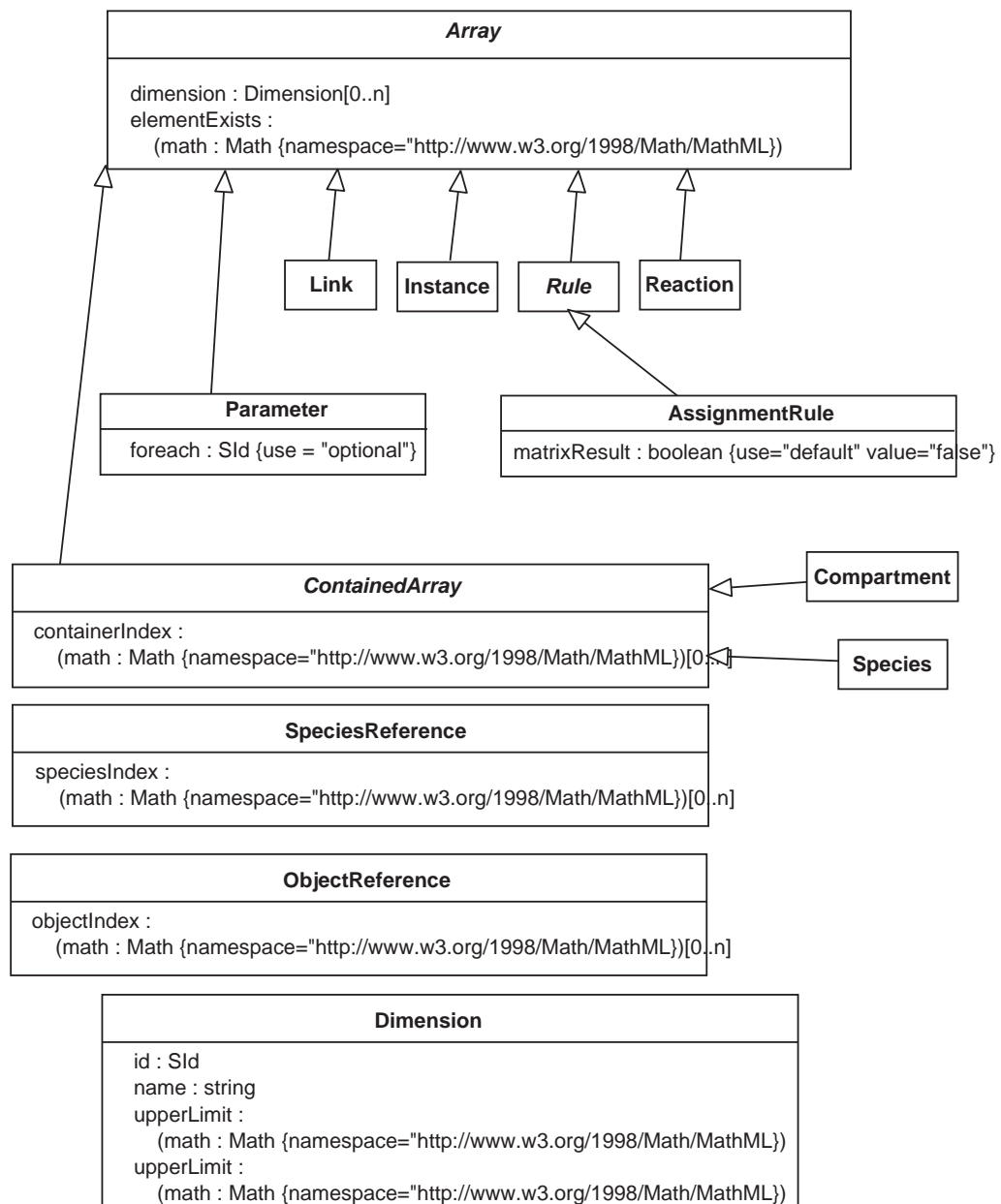


Figure 2: Changes made to existing SBML Level 2 structures with related additional structures

6.2 Array Element Object Reference

As shown in figure 2 it is proposed that structures which refer to other structures have _____Index fields (listOf_____Indices sub-elements) for example the type `SimpleSpeciesReference` has a `speciesIndex` field. These fields provide the index of the relevant referenced object with in an array.

These index fields should contain one integer expression for each dimension of the array being accessed.

We can use the `containerIndex` 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" 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>
    <listOfContainerIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
    </listOfContainerIndices>
  </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

In this section mechanisms within a math expression, for accessing specific elements of an array, are described.

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 species="s1">
          <listOfSpeciesIndices>
            <math xmlns="http://www.w3.org/1998/Math/MathML">

```

```

        <ci> x </ci>
      </math>
    </listOfSpeciesIndices>
  </speciesReference>
</listOfReactants>
<listOfProducts>
  <speciesReference species= "s2"/>
    <listOfSpeciesIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
    </listOfSpeciesIndices>
  </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, <http://www.sbml.org/symbols/selector> for use with the MathML `csymbol` element to define a selector operator that takes an arbitrary number of arguments. This operator can be used in place of a the `selector` element.

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 `containerIndex` field (`listOfContainerIndices` element) can be omitted. A structure configured in this fashion represents an array of species with the same specification as the given compartment array. 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" compartment="cell" 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>
  <listOfContainerIndices>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <ci> x </ci>
    </math>
  </listOfContainerIndices>
</species>

```

can be replaced with the equivalent structure:

```

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

```

If the species **dimension** field (**listOfDimensions** sub-element) is present but the **containerIndex** field (**listOfContainerIndices** sub-element) 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>

```

On a species element which omits the `containerIndex` field (`listOfContainerIndices` sub-element) then 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 `containerIndex` field yet still use symbols from the compartment's `dimension` field in the species' `elementExists` field (see section ?? 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>
    </dimension>
  </listOfDimensions>
  <compartment id="b" outside="a"/>
</listOfCompartments>

```

```

        </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"/>
  <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>
  <listOfContainerIndices>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <ci> x </ci>
    </math>
  </listOfContainerIndices>
</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`, 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 a `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>
    </assignmentRule>
  </listOfRules>
</model>

```

```

        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <cn> 0.1 </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 previous example 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>

```

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:

```

<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>

```

```

    </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>

```

8.2 Rules which supply a matrix result

Assignment Rules can be defined which apply to the whole of an array, or slices of arrays, 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 a 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>

```

```

        </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**.

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.

In this proposal the structures **Species**, **Compartment**, **Reaction** etc. include an additional **elementExists** MathML field. This field should only occur when the structure is declared as an array. **elementExists** contains a constant expression which defines whether an array element actually occurs at a given position in the array. This expression is conditional, which means that a false value implies that the element won't occur and a true value implies that an element will occur.

The default value of **elementExists**, true, ensures that by default the shape of the array specification is determined just by the **dimension** field (**listOfDimensions** sub-element) alone.

Apart from the **elementExists** field, the set of dimension symbols only have values for those elements that exist in the array in which they are declared.

The index operand to an array operator must refer to an element which exists in the array. This means that

the index must be a value to which the `elementExists` expression returns a true value.

9.1.1 Issue

To enable dynamic structures in SBML we could allow the `elementExists` expression to be dynamic. This would allow the model to change dynamically during the simulation. Perhaps other more explicit dynamic restructuring of a model should be considered in SBML. In fact we could generalise the concept of `elementExists` (perhaps renaming the field `exists`) so that it operates even when the given structure is not an array. In this way a model could change dynamically during a simulation.

9.2 Sparse Array Example

The following example shows a proposed structure for triangular arrays 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>
      <elementExists>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <leq/>
            <ci> x </ci>
            <ci> y </ci>
          </apply>
        </math>
      </elementExists>
    </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>
  </listOfCompartments>

```



```

        </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>
    <elementExists>
      <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>
</math>
</elementExists>
<listOfReactants>
    <speciesReference species= "s">
        <listOfSpeciesIndices>
            <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>
        </listOfSpeciesIndices>
    </speciesReference>
</listOfReactants>
<listOfProducts>
    <speciesReference species= "s"/>
        <listOfSpeciesIndices>
            <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>
        </listOfSpeciesIndices>
    </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 `elementExists` 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`

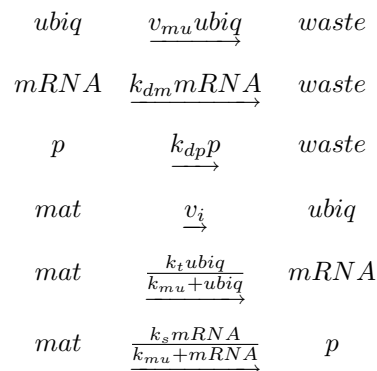
Please refer to the MathML specification (?) for further details of the operation of these elements.

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.

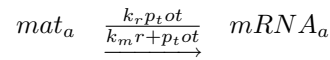
In the SBML given below the original model is broken into two parts the first SBML model encodes the biochemical network inside each cell (this model doesn't use any array features). The 2nd SBML model defines how this model is assembled into a tissue and uses the array features.

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



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.

This model is applied to a rectangular array of compartments. The 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 for



where p_{tot} is the total contribution of gene product from the cells neighbors. These equations are encoded as reaction in the first model. The p and p_{tot} species are the ports of the first model.

The second model simply creates an array of instances of the first model and calculates the value of p_{tot} for each 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"/>
    <species id="pTot" compartment="cell"/>
  </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>
      </kineticLaw>
    </reaction>
  </listOfReactions>
</model>
</sbml>
```

```

        </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>
  <reaction id="neighbour_mat2mRNA">
    <listOfReactants>
      <speciesReference species= "mat"/>
    </listOfReactants>
    <listOfProducts>
      <speciesReference species= "mRNA"/>
    </listOfProducts>
    <listOfModifiers>
      <modifierSpeciesReference species= "p"/>
    </listOfModifiers>
    <kineticLaw>
      <note>
        <p xmlns="http://www.w3.org/1999/xhtml">(kr * pTot)/(kmr + pTot)</p>
      </note>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
          <divide/>
          <apply>
            <times/>
            <ci>kr</ci>
            <ci>pTot</ci>
          </apply>
          <apply>
            <add/>
            <ci>kmr</ci>
            <ci>pTot</ci>
          </apply>
        </apply>
      </math>
      <listOfParameters>
        <parameter id="kr" value="0.1"/>
        <parameter id="kmr" value="0.07"/>
      </listOfParameters>
    </kineticLaw>
  </reaction>

```

```

        </reaction>
    </listOfReactions>
    <listOfPorts>
        <port object="pTot"/>
        <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>
    <listOfCompartments>
      <compartment id="cell_compartment">
        <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>
      </compartment>
    </listOfCompartments>
    <listOfSpecies>
      <species id="p" compartment="cell_compartment" initialAmount="0"/>
      <species id="pTot" compartment="cell_compartment" initialAmount="0"/>
    </listOfSpecies>
    <listOfRules>
      <assignmentRule id="pTot">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <sum/>
            <bvar><ci type="integer">xa</ci></bvar>
            <interval><cn>0</cn><ci>xBound</ci></interval>
          </apply>
          <bvar><ci type="integer">ya</ci></bvar>
          <interval><cn>0</cn><ci>yBound</ci></interval>
          <condition>
            <apply>
              <and/>
              <apply>
                <or/>
                <apply>
                  <eq/>

```

```

        <ci> x </ci>
        <ci> xa </ci>
    </apply>
    <apply>
        <eq/>
        <ci> y </ci>
        <ci> ya </ci>
    </apply>
</apply>
<apply>
    <or/>
    <apply>
        <neq/>
        <ci> x </ci>
        <ci> xa </ci>
    </apply>
    <apply>
        <neq/>
        <ci> y </ci>
        <ci> ya </ci>
    </apply>
</apply>
</condition>
<apply>
    <selector/>
    <ci>p</ci>
    <ci>xa</ci>
    <ci>xb</ci>
</apply>
</apply>
</math>
</assignmentRule>
</listOfRules>
<listOfInstances>
    <instance id="cell_instance" 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>
<listOfLinks>
    <link>
        <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>
    <from object="cell_compartment">
      <listOfObjectIndices>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <ci> x </ci>
        </math>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <ci> y </ci>
        </math>
      </listOfObjectIndices>
    </from>
    <to object="cell_instance">
      <listOfObjectIndices>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <ci> x </ci>
        </math>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <ci> y </ci>
        </math>
      </listOfObjectIndices>
      <subobject object="cell"/>
    </to>
  </link>
</link>
<link>
  <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>

```

```

        </math>
      </upperLimit>
    </dimension>
  </listOfDimensions>
  <from object="p">
    <listOfObjectIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> y </ci>
      </math>
    </listOfObjectIndices>
  </from>
  <to object="cell_instance">
    <listOfObjectIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> y </ci>
      </math>
    </listOfObjectIndices>
    <subobject object="p"/>
  </to>
</link>
<link>
  <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>
  <from object="pTot">
    <listOfObjectIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> y </ci>
      </math>
    </listOfObjectIndices>
  </from>
  <to object="cell_instance">
    <listOfObjectIndices>
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <ci> x </ci>
      </math>
      <math xmlns="http://www.w3.org/1998/Math/MathML">

```

```

        <ci> y </ci>
      </math>
    </listOfObjectIndices>
    <subobject object="pTot"/>
  </to>
</link>
</listOfLinks>
</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.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/>.