Arrays Package

Chris J. Myers

University of Utah

COMBINE 2013

September 20, 2013

Arrays Package Overview

- Extend SBML variables (species, parameters, etc.) to vectors/matrices.
- Support mathematical operations on vectors and matrices.
- Represent regular structures (i.e., arrays of submodels) more efficiently.
- Package consists of:
  - Extensions to the MathML subset.
  - A dimension class - used to specify size of arrays.
  - An index class - used to access the array elements.

Extensions to MathML Subset

- constructors: **matrix**, **matrixrow**, **vector**
- element referenced 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**

The Dimension Class

- id : SId
- name : String {use="optional"}
- size : Parameter (must be constant)

```
<parameter id="n" value="10"/>
<compartment id="C" size="1.0" ...>
 <arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="n"/>
  <arrays:dimension id="j" size="n"/>
 </arrays:orderedListOfDimensions>
</compartment>
```

- NOTE: we currently limit to only two dimensions with fixed sizes.

Objects That Can Have a List of Dimensions

For SBML L3V1 Core:

- Parameters
- Compartments
- Species
- Reactions
- Species references
- Rules
- Initial assignments
- Events
- Constraints

Packages may choose to allow dimensions (ex. subModels in comp).

The Index Class

- id : SId {use="optional"}
- name : String {use="optional"}
- math : MathML

Note that indices are 0-based.
QUESTION: are id and name needed?

The Index Class Example

```
for (i=0; i < n; i++) { y(10 - i) = x(i) }
```

```
<parameter id="n" value="10"/>
<parameter id="x" ...><arrays:orderedListOfDimensions>
 <arrays:dimension id="i" size="n"/>
</arrays:orderedListOfDimensions></parameter>
<parameter id="y" ...><arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="n"/>
</arrays:orderedListOfDimensions></parameter>
<assignmentRule variable="y">
 <arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="n"/>
 </arrays:orderedListOfDimensions>
 <arrays:orderedListOfIndices>
  <arrays:index><math>
   <apply><selector/><ci>y</ci><apply><minus/><cn>10</cn><ci>i</ci>
    </apply></apply></math>
  </arrays:index>
 </arrays:orderedListOfIndices>
 <math><apply><selector/><ci>x</ci><cn>i</cn></apply></math>
</assignmentRule>
```

Initial Assignment of Arrays

```
<parameter id="n" value="10"/>
<parameter id="m" value="5"/>
<parameter id="x"...><arrays:orderedListOfDimensions>
 <arrays:dimension id="i" size="n"/></arrays:orderedListOfDimensions>
</parameter>
<initialAssignment variable="x"><arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="m"/></arrays:orderedListOfDimensions>
 <arrays:orderedListOfIndices><arrays:index>
   <math><apply><selector/><ci>x<ci><ci>i</ci></apply></math>
 </arrays:index></arrays:orderedListOfIndices>
 <math><cn>5.7<cn></math>
</initialAssignment>
<initialAssignment variable="x"><arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="m"/></arrays:orderedListOfDimensions>
 <arrays:orderedListOfIndices><arrays:index>
   <math><apply><selector/><ci>x<ci><apply><plus/><ci>i</ci><ci>m</ci>
     </apply></apply></math>
 </arrays:index></arrays:orderedListOfIndices>
 <math><cn>3.2<cn></math>
</initialAssignment>
```

Initial Assignment of Arrays

```
<initialAssignment variable="x">
 <math><cn>5.7<cn></math>
</initialAssignment>

<initialAssignment variable="x">
 <math><cn>3.2<cn></math>
</initialAssignment>
```

PROBLEM: two initial assignments after removing arrays package.

Initial Assignment of Arrays

```
<initialAssignment variable="x">
 <math><cn>5.7<cn></math>
</initialAssignment>

<initialAssignment variable="x">
 <math><cn>3.2<cn></math>
</initialAssignment>
```

PROBLEM: two initial assignments after removing arrays package.

- Perhaps okay in L3V2.

Initial Assignment of Arrays

```
<initialAssignment variable="x__0">
 <math><cn>5.7<cn></math>
</initialAssignment>
...
<initialAssignment variable="x__4">
 <math><cn>5.7<cn></math>
</initialAssignment>
<initialAssignment variable="x__5">
 <math><cn>3.2<cn></math>
</initialAssignment>
...
<initialAssignment variable="x__9">
 <math><cn>3.2<cn></math>
</initialAssignment>
```

PROBLEM: two initial assignments after removing arrays package.

- Perhaps okay in L3V2.
- Problem is removed by flattening.

Objects That Can Have a List of Indices

For SBML L3V1 Core:

- Rules
- Initial assignments
- Species references
- Events assignments

Packages may choose to allow indices (replacements/deletions in comp).
QUESTION: what about other references in attributes?

Arrays Package Discussion

- Are people satisfied with the current approach or are changes needed?
- Which tool will be the second implementation?
- What support is needed in libsbml/JSBML to allow prototyping to begin?
- How will arrays interact with comp?

Arrays+Comp Example

```
<species id="S" ... >
 <arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="n"/>
 </arrays:orderedListOfDimensions>
 <comp:listOfReplacedElements>
  <comp:replacedElement comp:idRef="R" comp:submodelRef="M"/>
  <!-- express which index of S is replacing into which index of M -->
  <comp:replacedElement comp:idRef="R" comp:submodelRef="M"/>
  <!-- express which index of S is replacing into which index of M -->
 </comp:listOfReplacedElements>
</species>
...
<submodel id="M" ... >
 <arrays:orderedListOfDimensions>
  <arrays:dimension id="i" size="m"/>
 </arrays:orderedListOfDimensions>
</submodel>
```