

# Outline of Sets proposal + an Arrays and Model Composition reference implementation

Andrew Finney

# Issues with Array Proposal

- **Many biological systems**
  - are not bounded
    - e.g. tissues grow
  - are not fixed
    - e.g. tissues die
  - are not always ordered
  - orderings are often implemented by local relationships
- **Array based models have to address these issues explicitly**
  - e.g. invent schemes for tissue growth

# Sets

- **Put objects into sets**
  - no ordering
  - no bounds
  - dynamic
    - objects can change membership over time
    - objects have a defined lifetime
      - ‘born’ and ‘killed’ on events
- **Problems**
  - without ordering how do we access objects?
    - e.g. to create a reaction
  - what level of flexibility do we need with respect to set membership?

# Accessing Set Members

- Use “queries”
  - select members that have a given property value
    - e.g. pairs of objects in close proximity
- Member must have properties
  - e.g. position of member in space
  - basic objects, e.g. species, on their own don't have a rich enough set of properties
  - use submodel instances (see Model Composition)
    - e.g. instance `parameters` can contain the position in space
    - set members can only be submodel instances (?)

# Set Membership constraints

- **Why constrain?**
  - limit implementation cost/complexity
- **Essential constraint**
  - all members of a set have the same type
  - heterogeneity through type hierarchy
- **Some Possible constraints:**
  - no constraints
    - object can be in any number of sets
    - object lifetime independent of set membership
  - object member of only one set
    - object lifetime identical to set membership
  - one set for all members of the same type

# Reference Implementation of Arrays and Model Composition

- **Purpose**
  - demonstrate feasibility and usefulness of Array and Model Composition features
  - prototype new SBML concepts
  - provide potential standard reference
  - parser library for future Levels
- **Planned Features**
  - translate provisional Level 3 into Level 2
    - expands arrays
    - flattens submodel hierarchy
  - text based model capture of provisional Level 3
- **Under development**

# Text Format

- Example:

```
compartment x[0..10:z];
species a[0..10:z] in x[z];
variable p[0..10:z];
reaction r[0..9:z] : a[z] -> a[z + 1]
    : p[z] * a[z];
rule [1..10:z] p[z]' = p[z-1];
rule p[0]' = 1 ;
```

# Fragment of Provisional SBML Generated

```
<listOfCompartments>
  <compartment id="x">
    <listOfDimensions>
      <dimension id="z">
        <lowerLimit>
          <ns1:math
xmlns:ns1="http://www.w3.org/1998/Math/MathML">
            <ns1:cn>0</ns1:cn>
          </ns1:math>
        </lowerLimit>
        <upperLimit>
          <ns2:math
xmlns:ns2="http://www.w3.org/1998/Math/MathML">
            <ns2:cn>10</ns2:cn>
          </ns2:math>
        </upperLimit>
      </dimension>
    </listOfDimensions>
  </compartment>
</listOfCompartments>
```

# Flattened SBML Level 2 Form

```
<listOfCompartments>
```

```
  <compartment id="x_0" />
```

```
  <compartment id="x_1" />
```

```
  <compartment id="x_2" />
```

```
  <compartment id="x_3" />
```

```
  <compartment id="x_4" />
```

```
  <compartment id="x_5" />
```

```
  <compartment id="x_6" />
```

```
  <compartment id="x_7" />
```

```
  <compartment id="x_8" />
```

```
  <compartment id="x_9" />
```

```
  <compartment id="x_10" />
```

```
</listOfCompartments>
```

# Implementation and Status

- **Implementation**
  - Java
  - Object Model generated using JAXB
  - Text Format parsing (not XML) parsed using JavaCC
- **Status**
  - Open source
    - on SBML sourceforge site
  - Basic Array Implementation completed
  - More to do
    - validation of inputs
      - provisional level 3
      - text format
    - documentation
    - model composition
    - try real models!