



# SBRML Interoperability

Frank T. Bergmann, PhD

California Institute of Technology

SBRML: Pedro Mendes, Joseph O. Dada

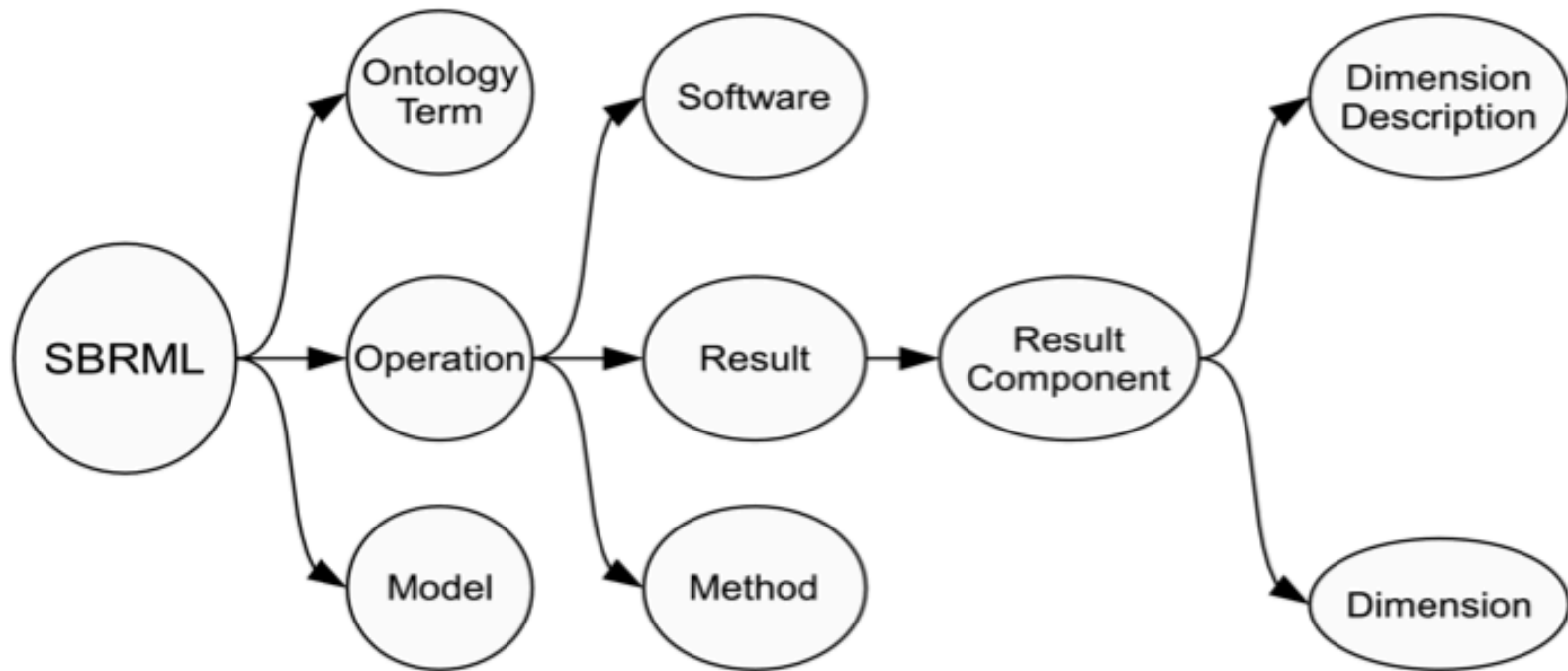
SED-ML: Nicolas Le Novère, Dagmar Waltemath, Frank T. Bergmann

Organizer: Sven Sahle

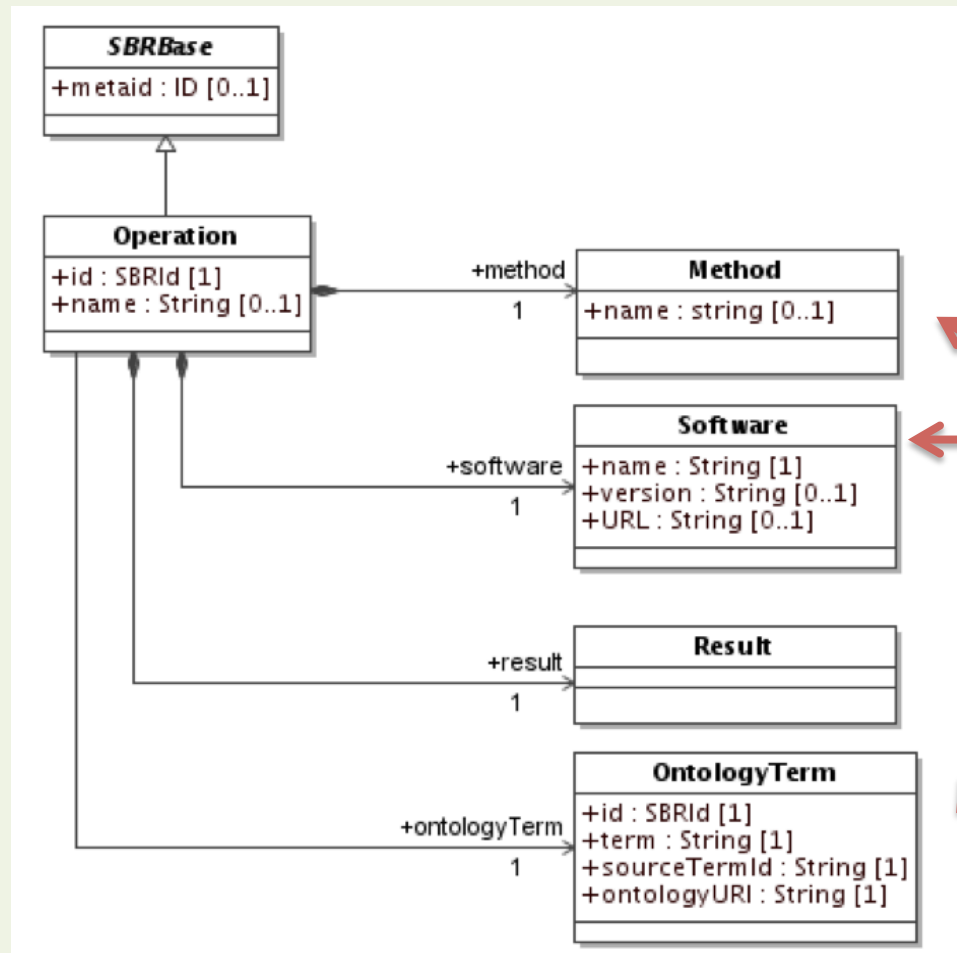
# **THE MEETING**

# **SYSTEMS BIOLOGY RESULTS MARKUP LANGUAGE (SBRML)**

# SBRML - Overview

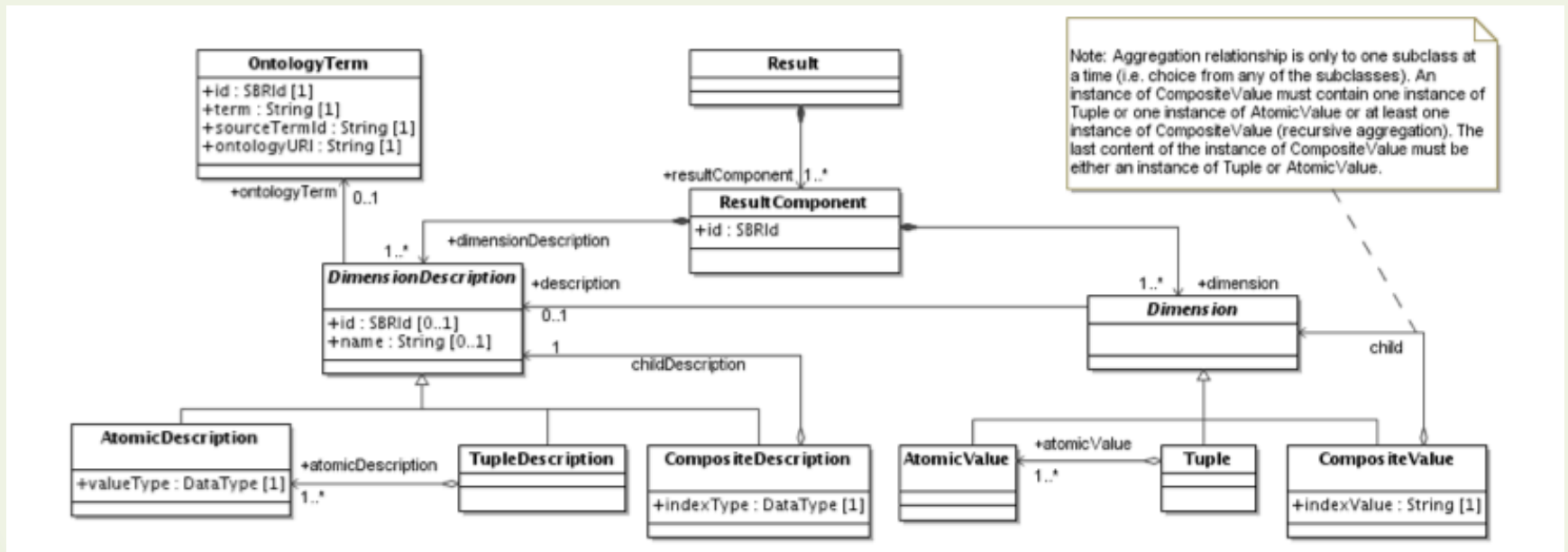


# SBRML - Operation



Overlap!

# SBRML - Results



# SBRML – Proposed Change I

- Move the result component (and downstream elements in the UML diagram) into its own namespace, as this is the primary component that would be referenced by SED-ML descriptions
- This would effectively separate the numerical components in SBRML from the descriptive components
- SBRML remains untouched otherwise, existing software that uses it can be quickly updated

# SBRML – Proposed Change II

- The dimension description in SBRML refers to the one model that gave rise to the data.
- By introducing a new indexType 'xpath' we will be able to reuse this indexing for the model referenced by SED-ML

```
<dimensionDescription>
  ▼<compositeDescription name="Time" ontologyTerm="term3" indexType="double">
    ▼<compositeDescription name="species" indexType="string">
      ▼<tupleDescription>
        <atomicDescription name="Concentration" ontologyTerm="term4" valueType="double"/>
        <atomicDescription name="Particle Numbers" ontologyTerm="term5" valueType="Integer"/>
      </tupleDescription>
    </compositeDescription>
  </compositeDescription>
</dimensionDescription>
```



# **HOW DOES SBRML BENEFIT FROM SED-ML**

# SBRML references SED-ML

- The minimal change for SBRML is to add an optional attribute 'source' to the Method Element, which would allow it to reference the SED-ML simulation description (Or other formats).
- Other possible changes discussed include subclassing the SBRML Model / Method or Operation class to add support for SED-ML elements there.

# **HOW AND TO GET IT INTO SED-ML**

# Use Cases

The newly extracted numerical core (dubbed NUML = NUmerical Markup Language) will be of interest in many places:

- Referencing previous Simulation Result to parameterize simulation.
- Reference external data in post processing (i.e. DataGenerators)
- Referencing (external) data in model pre-processing (i.e.: Model changes)

# Externalize stuff

```
▼<resultComponent id="component1">
  ▼<dimensionDescription>
    ▼<compositeDescription name="Time" ontologyTerm="term3" indexType="double">
      ▼<compositeDescription name="species" indexType="string">
        ▼<tupleDescription>
          <atomicDescription name="Concentration" ontologyTerm="term4" valueType="double"/>
          <atomicDescription name="Particle Numbers" ontologyTerm="term5" valueType="Integer"/>
        </tupleDescription>
      </compositeDescription>
    </dimensionDescription>
  ▼<dimension>
    ▼<compositeValue indexValue="0">
      ▼<compositeValue indexValue="Phser">
        ▼<tuple>
          <atomicValue>0</atomicValue>
          <atomicValue>0</atomicValue>
        </tuple>
      </compositeValue>
    </compositeValue>
    ▼<compositeValue indexValue="1">
      ▼<compositeValue indexValue="Phser">
        ▼<tuple>
          <atomicValue>0.996305</atomicValue>
          <atomicValue>5.99989e+17</atomicValue>
        </tuple>
      </compositeValue>
    </compositeValue>
    ▼<compositeValue indexValue="2">
      ▼<compositeValue indexValue="Phser">
        ▼<tuple>
          <atomicValue>1.98526</atomicValue>
          <atomicValue>1.19555e+18</atomicValue>
        </tuple>
      </compositeValue>
    </compositeValue>
  </dimension>
</resultComponent>
```



Vital For SED-ML

This could actually be in an external file.

# Basic Idea

- Declare a way to index the data wherever it may come from
- Reference it in ChangeMath / DataGenerator and possible future simulation classes

# Add ListOf<Data> to SED-ML

```
<listOfData>  
  <dataFromSimulation id="data1" task="task1">  
    <numl:dimensionDescription>  
      <numl:compositeDescription numl:indexType="double">  
        <numl:compositeDescription numl:indexType="xpath">  
          <numl:atomicDescription numl:valueType="double"/>  
        </numl:compositeDescription>  
      </numl:compositeDescription>  
    </numl:dimensionDescription>  
  </dataFromSimulation>  
</listOfData>
```

# Add ListOf<Data> to SED-ML

```
<listOfData>  
  <dataFromSimulation id="data1" task="task1">  
    <numl:dimensionDescription>  
      <numl:comp >  
        <numl:com >  
          <numl:at >  
        </numl:co >  
      </numl:com >  
    </numl:dime >  
  </dataFromSi >  
</listOfData>
```

Similarly for other data:

“externalData” would just have a source attribute instead of a task.

“data” could inline data using “numl:dimension”



# Using the ListOf<Data>

- After the values are declared, their identifiers could be used in all MathML expressions!
- SED-ML L1V1 already introduces the MathML Lambda element!
- This would allow us to use the aforementioned definitions as:
  - $\text{value} = \text{data1}(\text{time}, \text{variable})$

# Thoughts / Questions ?

- Get involved:
  - [sed-ml-discuss@lists.sourceforge.net](mailto:sed-ml-discuss@lists.sourceforge.net)
- More information:
  - <http://www.comp-sys-bio.org/tiki-index.php?page=SBRML>
  - <http://sed-ml.org/>
  - <http://sbrml.sourceforge.net/>