Introduction to SED-ML - the Simulation Experiment Description Markup Language

BrainScaleS CodeJam/NeuroML workshop, Edinburgh

13th March 2012

Dagmar Waltemath
Motivation

[a probably copy-righted Whisky image] & [a probably copy-righted honey image] =

Quality of voice

< 1/day
< 3 cups/day
-> 1/day
-> 3 cups/day

whisky
honey
combined

www.sbi.uni-rostock.de
Motivation

Simulation tool

Biological publication repository

models
Simulation results

Simulation tool

Biological publication repository

models
“[..] in Biomodels database the model BIOMD0000000139 and BIOMD0000000140 are two different models and they are supposed to show different results. Unfortunately simulating them in Copasi gives same result for both the models. [..] “
(arvin mer on sbml-discuss)

Fig.: running model files (COPASI simulation tool)
Motivation

Simulation tool

Biological publication repository

Simulation results

(SBW Workbench)
Levels: major revisions containing substantial changes

Versions: minor revisions containing corrections and refinements

Editorial board: coordinates SED-ML development (elected by sed-ml-discuss members)

SED-ML Level 1 Version 1:
- multiple models
- multiple simulation setups
- time course simulations
Major building blocks

Fig.: SED-ML structure (Waltemath et al., 2011)
Model Class: defining the models used in the experiment set-up

- Ideally: link that unambiguously defines a model in an open repository
- Optionally: model preprocessing
e.g., updated or additional model parameter, substituted mathematical function...

```xml
<model id="modell" name="spiking neuron"
  language="urn:sedml:language:sbml.level-2.version-3"
  source="urn:miriam:biomodels.db:BIOMD0000000127">
  <listOfChanges>
    <changeAttribute
      target="/sbml/model/listOfParameters/parameter[@name='c']/@value"
      newValue="-55">
      </changeAttribute>
  </listOfChanges>
</model>
```
Model Class: defining the models used in the experiment set-up

- Ideally: link that unambiguously defines a model in an open repository
- Optionally: model preprocessing
e.g., updated or additional model parameter, substituted mathematical function...

```xml
<model id="model1" name="spiking neuron"
      language="urn:sedml:language:sbml.level-2.version-3"
      source="urn:miriam:biomodels.db:BIOMD000000127">
  <listOfChanges>
    <changeAttribute
      target="/sbml/model/listOfParameters/parameter[@name='c']/@value"
      newValue="-55"/>
  </listOfChanges>
</model>
```
Model Class: defining the models used in the experiment set-up

- Ideally: link that unambiguously defines a model in an open repository
- Optionally: model preprocessing
  e.g., updated or additional model parameter, substituted mathematical function...

```xml
<model id="model1" name="spiking neuron"
  language="urn:sedml:language:sbml.level-2.version-3"
  source="urn:miriam:biomodels.db:BIOMD000000127">
  <listOfChanges>
    <changeAttribute
      target="/sbml/model/listOfParameters/parameter[@name='c']/@value"
      newValue="-55">
    </changeAttribute>
  </listOfChanges>
</model>
```
Model Class: defining the models used in the experiment set-up

- Ideally: link that unambiguously defines a model in an open repository
- Optionally: model preprocessing
  e.g., updated or additional model parameter, substituted mathematical function...

```xml
<model id="model1" name="spiking neuron" language="urn:sedml:language:sbml.level-2.version-3" source="urn:miriam:biomodels.db:BIOMD0000000127">
  <listOfChanges>
    <changeAttribute
target="/sbml/model/listOfParameters/parameter[@name='c']/@value" newValue="-55">
    </changeAttribute>
  </listOfChanges>
</model>
```
Simulation Class: setting up the simulation

- Defining the type of simulation
- Defining the simulation algorithm to apply and its settings
  • Reference: Kinetic Simulation Algorithm Ontology
    http://www.biomodels.net/kisao/

<uniformTimeCourse id="simulation1"
  initialTime="0"
  outputStartTime="0"
  outputEndTime="1000"
  numberOfPoints="1000">
  <algorithm kisaoID="KISAO:0000088" />
</uniformTimeCourse>
Simulation Class: setting up the simulation

- Defining the type of simulation
- Defining the simulation algorithm to apply and its settings
  - Reference: Kinetic Simulation Algorithm Ontology
    http://www.biomodels.net/kisao/

```xml
<uniformTimeCourse id="simulation1"
  initialTime="0"
  outputStartTime="0"
  outputEndTime="1000"
  numberOfPoints="1000">
  <algorithm kisaoID="KISAO:0000088" />
</uniformTimeCourse>
```

Fig.: COPASI simulation tool
Simulation Class: setting up the simulation

- Defining the type of simulation
- Defining the simulation algorithm to apply and its settings
  
  • Reference: Kinetic Simulation Algorithm Ontology
  http://www.biomes.org/kisao/

<uniformTimeCourse id="simulation1"
  initialTime="0"
  outputStartTime="0"
  outputEndTime="1000"
  numberOfPoints="1000">
  <algorithm kisaoID="KISAO:0000088" />
</uniformTimeCourse>
TaskClass: grouping models and simulations

- Assigning 1 simulation to 1 model at a time
- Multiple tasks

```xml
<listOfTasks>
  <task id="task1" name="spiking with initial parameters" modelReference="model1" simulationReference="simulation1" /> 
  <task id="task2" name="spiking with updated parameters" modelReference="model2" simulationReference="simulation1" /> 
</listOfTasks>
```
TaskClass: grouping models and simulations

- Assigning 1 simulation to 1 model at a time
- Multiple tasks

```xml
<listOfTasks>
  <task id="task1" name="spiking with initial parameters" modelReference="model1" simulationReference="simulation1" />
  <task id="task2" name="spiking with updated parameters" modelReference="model2" simulationReference="simulation1" />
</listOfTasks>
```
- Defining all entities needed in the outputs
- Processing of result data after simulation
- Only on explicitly defined model entities
- ...and on predefined implicit variables, e.g., time

```xml
<dataGenerator id="v" name="voltage">
  <listOfVariables>
    <variable id="v1" taskReference="task1" target="/sbml/model/listOfParameters/parameter[@id='v']"/>
  </listOfVariables>
  <math:math>
    <math:ci>v1</math:ci>
  </math:math>
</dataGenerator>
```
DataGenerator Class: post-processing

- Defining all entities needed in the outputs
- Processing of result data after simulation
- Only on explicitly defined model entities
- ...and on predefined implicit variables, e.g., time

```xml
<dataGenerator id="v" name="voltage">
  <listOfVariables>
    <variable id="v1" taskReference="task1" target="/sbml/model/listOfParameters/parameter[@id='v']"/>
  </listOfVariables>
  <math:math>
    <math:ci>v1</math:ci>
  </math:math>
</dataGenerator>
```

“voltage”
Output Class: defining the simulation output

- Defining an output type
- Defining what is to be stored/plotted for the output

```xml
<plot2D id="plot1_Basic" name="voltage change over time">
  <listOfCurves>
    <curve id="c1" logX="false" logY="false"
      xDataReference="timeDG" yDataReference="v" />
  </listOfCurves>
</plot2D>
```
Output Class: defining the simulation output

- Defining an output type
- Defining what is to be stored/plotted for the output

```xml
<plot2D id="plot1_Basic" name="voltage change over time">
  <listOfCurves>
    <curve id="c1" logX="false" logY="false"
       xDataReference="timeDG" yDataReference="v" />
  </listOfCurves>
</plot2D>
```

Fig.: COPASI simulation tool
Example: SED-ML in Computational Biology

SEDML Web Tools

Simulate

Steady State Scan (Boris 2D)

http://sysbioapps.dyndns.org/SED-ML_Web_Tools

SED-ED (Bioinformatics. 2012 Feb 25)

Fig.: SED-ML import/export in CellDesigner
Example: SED-ML and the CellML repository

An OpenCell 0.8 session file is available. SED-ML can also be used to simulate this model, the simulation description is in Lorenz_1963_sedml.xml, and the simulation experiment can be run using the SED-ML Web Tools. The figures below show the results from OpenCell and from using SED-ML.

Fig.: CellML models with supplement SED-ML files. (CellML model repository)
Example: SED-ML for computational neuroscience

Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task.

Roitman JD, Shadlen MN.
Program in Neurobiology and Behavior, Department of Physiology and Biophysics, and Regional Primate Research Center, University of Washington, Seattle, Washington 98195-7290, USA.

Fig.: Roitman, Shadlen (2002)

http://ars.informatik.uni-rostock.de/
How to contribute to SED-ML

1. Have a look at the current SED-ML L1 V1 Specification document on http://sed-ml.org

2. Try out some of the existing examples

3. Identify what is missing for you to encode your simulation experimental setups - What can you not express?

4. Submit a feature request & post it on the list
   feature request tracker: http://sourceforge.net/projects/sed-ml
   mailing list: sed-ml-discuss@lists.sourceforge.net

5. ... submit a proposal with example files and prototype
   proposal tracker: http://sourceforge.net/projects/sed-ml
Thank you for your attention!

Frank Bergmann (editor, elected 2011-2014)
Frank T. Bergmann (PhD in Computational and Systems Biology) is a researcher at the California Institute of Technology where his primary interest is in standardization efforts around SBML. He is also the lead developer of the Systems Biology Workbench along with the LibSEDML library.

Dagmar Waltemath (editor, elected 2011-2014)
Dagmar Waltemath (Diploma degree in Computer Science) is guest researcher at the Systems Biology and Bioinformatics group, Rostock. She works on the standardisation of simulation experiment descriptions in Computational Biology (MIASE).

Richard Adams (editor, elected 2011-2013)
Richard Adams (PhD in Cell Biology) is software project manager at the Centre for Systems Biology, Edinburgh. He works on the SBSI systems biology software framework, SED-ML tools and the jibsedml Java library for SED-ML.

David Nickerson (editor, elected 2011-2013)
David Nickerson is a Research Fellow in the Auckland Bioengineering Institute where he leads the Auckland Kidney Physiome project. David is also involved in many aspects of the CellML project as well as various cardiac modeling projects. He also develops several CellML-related software tools.

Andrew Miller (editor, elected 2011-2012)
Andrew Miller is a researcher at the Auckland Bioengineering Institute. His research interests focus around the representation of mathematical models; he is involved in the development of tools for processing CellML models, including SProS, a SED-ML processing service that forms part of the CellML API.

Nicolas Le Novère (editorial advisor)
Nicolas Le Novère is a group leader at the EMBL-European Bioinformatics Institute. His research unfolds along two axis: 1) modelling neuronal signalling, at the molecular, sub-cellular and cellular levels, and 2) developing tools and resources for systems biology, in particular including standards.

http://sed-ml.org