

# Experience with SBML Packages within iBioSim

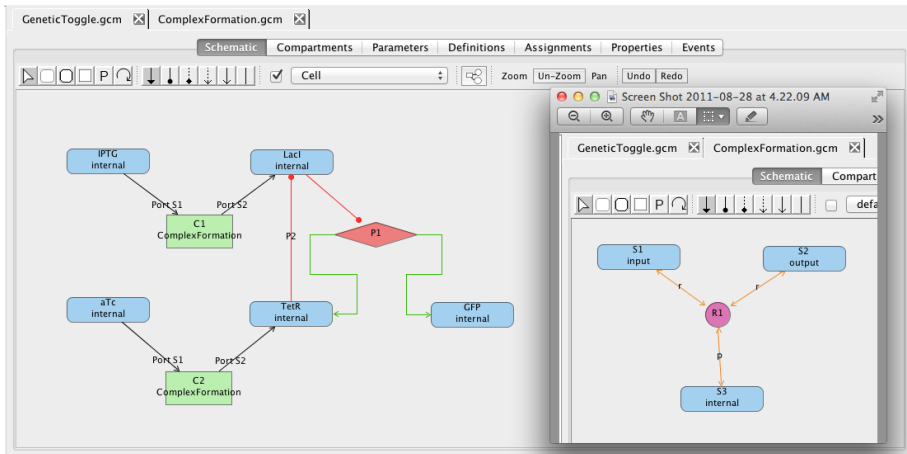
Chris J. Myers

University of Utah

COMBINE 2012

August 15, 2012

# iBioSim 2.0 (COMBINE 2011)



# iBioSim 2.0 (COMBINE 2011)

The screenshot shows the SBOL Browser application window. It has a title bar with standard OS window controls and the text "SBOL Browser". The main area is divided into two panes. The left pane, titled "Collections:", contains a list with "all" selected and "exampleCollection" below it. The right pane, titled "DNA Components:", contains a list of component IDs: BBa\_B0010, BBa\_B0012, BBa\_B0015, BBa\_B0034, BBa\_C0012 (selected), BBa\_C0040, BBa\_K592101, BBa\_R0010, and BBa\_R0040. Below these panes is a large text area displaying the details for the selected component, BBa\_C0012. At the bottom right of the window are "Cancel" and "Ok" buttons.

SBOL Browser

Collections:

- all
- exampleCollection

DNA Components:

- BBa\_B0010
- BBa\_B0012
- BBa\_B0015
- BBa\_B0034
- BBa\_C0012
- BBa\_C0040
- BBa\_K592101
- BBa\_R0010
- BBa\_R0040

Display ID: BBa\_C0012  
Name: lacI repressor from E. coli (+LVA)  
Description: Sequence taken from the repressilator of Elowitz and Leibler (2000). The obtained sequence was compared to the wild-type sequence for LacI obtained through a database search. The sequence had been modified from the wild-type in that wild-type GTG start was changed to an ATG start (note, actual ORF in E.coli has several GTG starts it would seem). The LVA tag has been added for quicker degradation. Incompatible with systems containing LacI, lactose, or IPTG.  
Annotations: NA  
Types: SO\_0000316  
DNA Sequence: atggtgaatgtgaaccagtaacggtatacagatgtcgcagagatgcccgggtgtcttctatcagaccgtttccgcgctggtgaaccaggccagccacgctttctgcgaaaacgcgggaaaaagtgaagcggcggatggcggagctgaattacattcccaaccgcgtggcacaacaactggcgggcaaacagctggtgctgattggcgttccacctcca gtcctggccctgcacgcgcctgcgcaaatgtcgcggcgattaaatctcgcgccgatcaactgggtgcccagcgtggtggtgctgatggtagaacgaagcggcgtcgaagcctgtaa agcggcggtgcacaatctctcgcgcaacgcgtcagtgggctgatcattaactatcgcctggatgaccaggatgccattgctggaagctgcctgcactaatgtccggcgttattc ttgatgtctcgtacagacaccatcaacagattattttctccatgaagacggtagcgcgactggcgtggagcatctggtgcattgggtaccagaaatcgcgctgtagcggg cccattaagtctcgtcggcgcgtcgtcgtctggctggctggcgcataaatatctcactcgaatcaaatcagccgatagcggaaacgggaagcgcgactggagtgccatgtccggttt caacaacacatgcaaatgctgaatgagggcatcgttccactgcgatcgtggtgccaacgatcagatggcgtggcgcgaatgcgcgcattaccgagtcgggctgcgcggtt gtcggatatctcggtagtgggatacagcagataccgaagacagctcatgttatccgccgttaaccaccatcaaacaggatttgcctgctggtgggcaaacagcgtggaccgtt

Cancel Ok

# iBioSim 2.0 (COMBINE 2011)

simPop GeneticToggle.gcm

Simulation Options Abstraction Options Schematic Parameters SBML Elements TSD Graph Histogram

C1 GeneticToggle	C2 GeneticToggle	C3 GeneticToggle	C4 GeneticToggle	C5 GeneticToggle	C6 GeneticToggle	C7 GeneticToggle	C8 GeneticToggle	C41 GeneticToggle
C9 GeneticToggle	C10 GeneticToggle	C11 GeneticToggle	C12 GeneticToggle	C13 GeneticToggle	C14 GeneticToggle	C15 GeneticToggle	C16 GeneticToggle	C42 GeneticToggle
C17 GeneticToggle	C18 GeneticToggle	C19 GeneticToggle	C20 GeneticToggle	C21 GeneticToggle	C22 GeneticToggle	C23 GeneticToggle	C24 GeneticToggle	C43 GeneticToggle
C25 GeneticToggle	C26 GeneticToggle	C27 GeneticToggle	C28 GeneticToggle	C29 GeneticToggle	C30 GeneticToggle	C31 GeneticToggle	C32 GeneticToggle	C44 GeneticToggle
C33 GeneticToggle	C34 GeneticToggle	C35 GeneticToggle	C36 GeneticToggle	C37 GeneticToggle	C38 GeneticToggle	C39 GeneticToggle	C40 GeneticToggle	C45 GeneticToggle

Choose Simulation Clear Appearances

- Import/edit/export all of SBML Level 3 Version 1 core.
- Simulates all of SBML L3V1 except csymbol delay.
- Internally uses a custom *genetic circuit model* (GCM) language with:
  - Support for hierarchical models.
  - Schematic layout information.
  - Special constructs to model genetic regulation.
  - *Synthetic biology open language* (SBOL) annotations.
  - Information for spatial grid models.
  - Special functions for random distributions and dynamic modeling.
- Can these features be supported by SBML L3V1 with packages?

- Uses SBML L3V1 with packages throughout.
  - Uses *comp* package to support hierarchical models.
  - Uses *layout* package to support schematic information.
  - Represents genetic regulation with reactions and SBO terms.
  - SBML elements annotated with URIs from SBOL DNA component libraries.
  - Spatial grids modeled with custom array annotations.
  - Special functions for random distributions and dynamic modeling.

- *comp* package
  - Should all references be through ports?
  - Should all replacements be like for like?
  - Should conversion factors be kept?
- *layout* package
  - How to reference items without an SId?
  - How to represent rules, events, constraints, and subModels?
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like?
  - Should conversion factors be kept?
- *layout* package
  - How to reference items without an SId?
  - How to represent rules, events, constraints, and subModels?
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package



- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept?
- *layout* package
  - How to reference items without an SId?
  - How to represent rules, events, constraints, and subModels?
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId?
  - How to represent rules, events, constraints, and subModels?
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?  
**No, only initial and event assignments, delays, and priorities**
  - How should it refer to distributions?
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?  
**No, only initial and event assignments, delays, and priorities**
  - How should it refer to distributions? **Using URIs**
  - How to deal with multivariate distributions?
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?  
**No, only initial and event assignments, delays, and priorities**
  - How should it refer to distributions? **Using URIs**
  - How to deal with multivariate distributions? **Use the arrays package**
- *arrays* package
  - How to support arrays in math?
- *dynamic* package

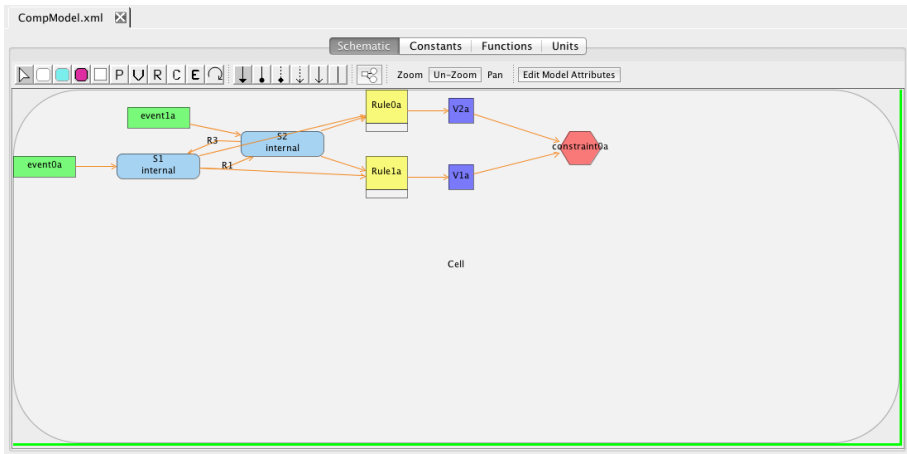
- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?  
**No, only initial and event assignments, delays, and priorities**
  - How should it refer to distributions? **Using URIs**
  - How to deal with multivariate distributions? **Use the arrays package**
- *arrays* package
  - How to support arrays in math? **Extend supported mathML subset**
- *dynamic* package



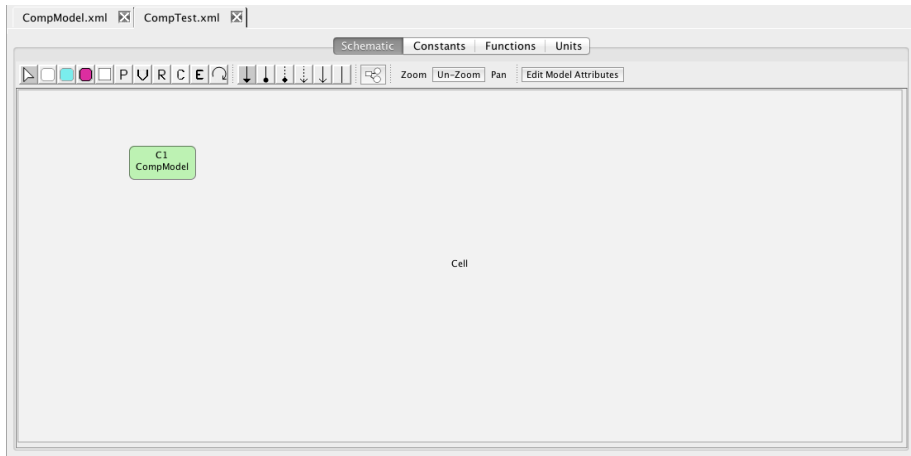
- *comp* package
  - Should all references be through ports? **Best practice**
  - Should all replacements be like for like? **Yes**
  - Should conversion factors be kept? **Yes, but simplified**
- *layout* package
  - How to reference items without an SId? **Add a metaldRef**
  - How to represent rules, events, constraints, and subModels?  
**Add GeneralGlyph and ReferenceGlyph**
- *distrib* package
  - Can distributions be used in all expressions?  
**No, only initial and event assignments, delays, and priorities**
  - How should it refer to distributions? **Using URIs**
  - How to deal with multivariate distributions? **Use the arrays package**
- *arrays* package
  - How to support arrays in math? **Extend supported mathML subset**
- *dynamic* package
  - **Keep separate from arrays package, introduce special events for birth, death, movement, etc., but many issues remain.**

- Completed support for entire comp package including conversion factors and local parameter replacements.
- Utilize GeneralGlyph and ReferenceGlyph for rules, events, constraints, and subModels.
- Introduced proper RDF style SBOL and grid annotations.

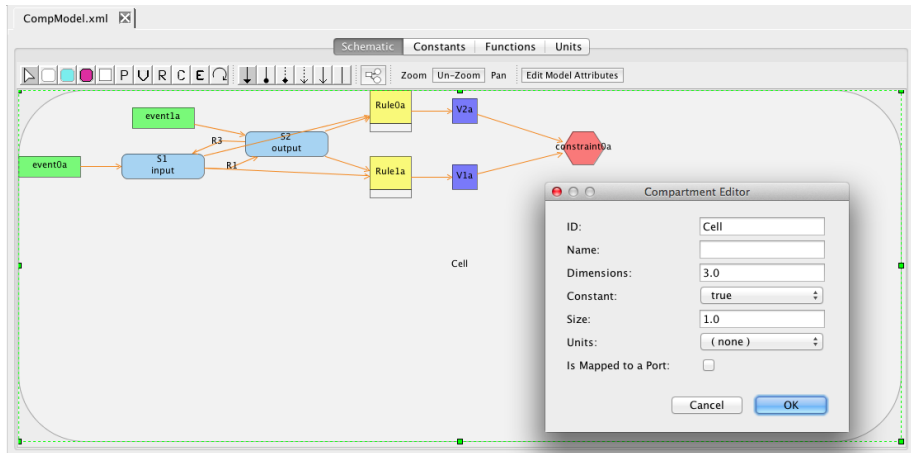
# iBioSim 2.2 (COMBINE 2012)



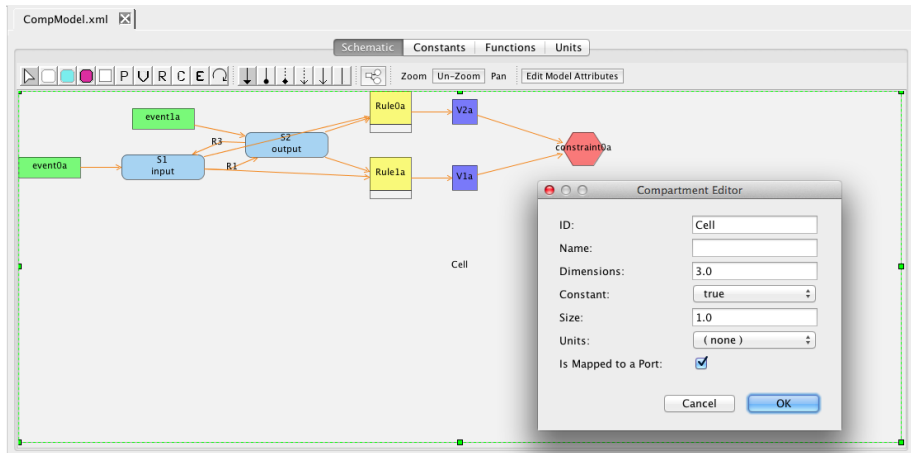
# iBioSim 2.2 (COMBINE 2012)



# iBioSim 2.2 (COMBINE 2012)



# iBioSim 2.2 (COMBINE 2012)

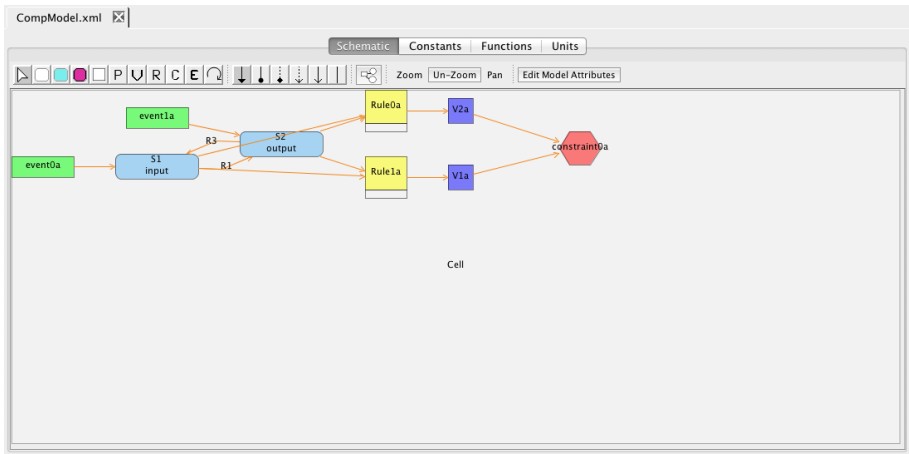


# iBioSim 2.2 (COMBINE 2012)

The screenshot displays the iBioSim 2.2 interface. On the left, a model diagram shows a compartment labeled 'S1' with an 'internal' state. It is connected to two events: 'event0a' and 'event1a'. Reaction R1 is associated with the compartment, and reaction R3 is associated with event1a. On the right, the 'Species Editor' dialog box is open for species 'S1'. The 'Type' dropdown menu is expanded, showing options: 'input' (selected), 'internal', 'output', 'constitutive', and 'degrades'. The 'Compartment' is set to 'Cell'. Other parameters include Boundary Condition (false), Constant (false), Has Only Substance Units (false), Units (none), Conversion Factor (none), and Initial Amount/Concentration ([0.0]). Degradation rate (kd) is set to default (0.0075), Complex formation equilibrium (Kc) to default (0.05/1.0), and Membrane diffusion rate (fd/rv) (kmdiff) to default (1.0/0.01). An 'Associate SBOL' button is present at the bottom of the dialog, along with 'Cancel' and 'Ok' buttons.

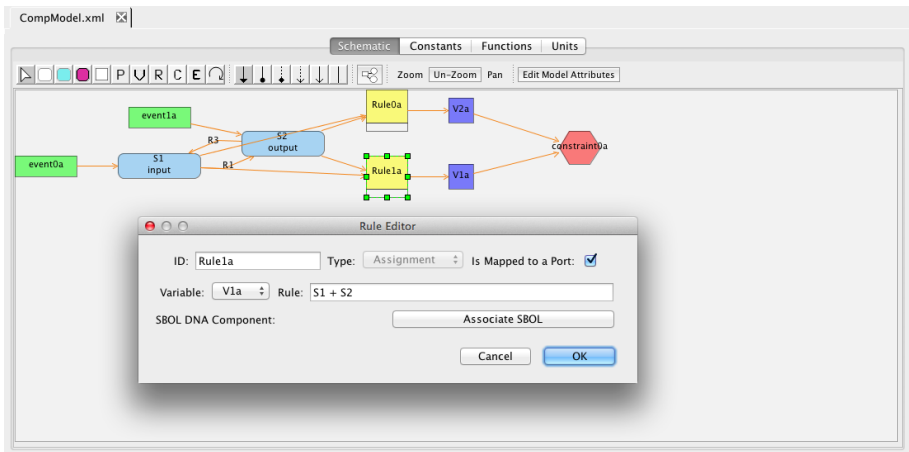
Property	Value
ID	S1
Name	
Type	input
Compartment	Cell
Boundary Condition	false
Constant	false
Has Only Substance Units	false
Units	( none )
Conversion Factor	( none )
Initial Amount/Concentration	[0.0]
Degradation rate (kd)	default 0.0075
Complex formation equilibrium (Kc)	default 0.05/1.0
Membrane diffusion rate (fd/rv) (kmdiff)	default 1.0/0.01

# iBioSim 2.2 (COMBINE 2012)

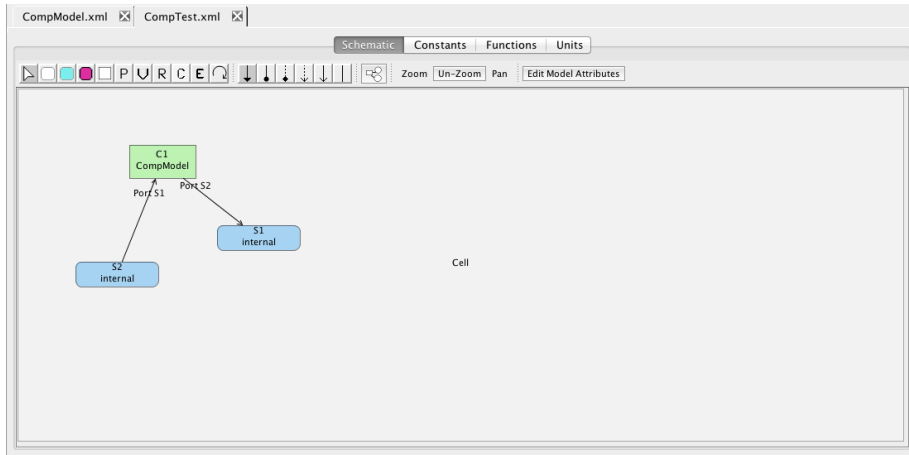




# iBioSim 2.2 (COMBINE 2012)



# iBioSim 2.2 (COMBINE 2012)



# iBioSim 2.2 (COMBINE 2012)

Component Editor

ID: C1

Type	Port	Direction	Replacement	Conversion
compartment	Cell	<--	Cell	(none)
localParameter	R1__kf	<--	--delete--	(none)
localParameter	R3__kf	<--	topKf	(none)
species	S1	<--	S2	conv
species	S2	-->	S1	(none)
reaction	R1	<--	--none--	(none)
reaction	R3	<--	--none--	(none)
functionDefinition	f	<--	--delete--	(none)
unitDefinition	perSecond	<--	--delete--	(none)
assignmentRule	Rule1a	<--	--include--	(none)
constraint	constraint0a	<--	--include--	(none)
event	event0a	<--	--include--	(none)
event	event1a	<--	--include--	(none)
rateRule	Rule0a	<--	--include--	(none)

SBOL DNA Component: Associate SBOL

Time Conversion Factor: (none)

Extent Conversion Factor: (none)

Cancel Ok

# Best Practices for the Layout Package

- Restrict use of `CompartmentGlyph`, `SpeciesGlyph`, `ReactionGlyph`, and `SpeciesReferenceGlyphs` to their respective SBML elements.
- Use `GeneralGlyph` and `ReferenceGlyph` for everything else including rules, events, constraints, and `subModels`.
- Remember glyph SIds are in the same namespace of most other SBML elements.
- Prefer the use of `SIdRefs` to `metaldRefs`.

# Best Practices for the Comp Package

- Should refer to SBML elements in replacements/deletions consistently.
  - Prefer portRefs to SIdRefs, and SIdRefs to metaldRefs.
  - If no identifier exists, must make a copy of the model to add an identifier.
- Should include ports for elements which are replaced or deleted.
  - Provides an interface to facilitate modular design of models.
  - Simplifies user interface since only ports need to be presented as potential locations for replacements or deletions.
  - Simplifies maintenance of models, since replacements and deletions remain valid as long as the ports remain unchanged.
  - Using ports to forward connectivity to nested subModels rather than recursive SBaseRefs further simplifies model maintenance.
  - Assuming one has a local copy of their models, the use of ports does not limit the use of the *comp* package.

# Import/Export with Packages

- All tools can import SBML L3V1 models that use layout and comp:

```
SBMLReader reader = new SBMLReader();
SBMLDocument document = reader.readSBML(filename);
CompModel compModel = (CompModelPlugin)
    document.getModel().getPlugin("comp");
SBMLDocument flatDoc = new SBMLDocument();
Model flatModel = compModel.flattenModel();
flatDoc.setModel(flatModel);
flatDoc.enablePackage(CompExtension.getXmlnsL3V1V1(),
    "comp", false);
flatDoc.enablePackage(LayoutExtension.getXmlnsL3V1V1(),
    "layout", false);
```

- Similarly, all tools using layout and comp packages can export a flattened SBML L3V1 core model.

# Current Package Status

- *layout* package
  - Supported by `libsbml` and several tools.
  - Specification needs to be updated.
- *comp* package
  - Supported by `libsbml`, `Antimony`, and `iBioSim`.
  - Exchange has been demonstrated.
  - Specification is complete and ready for final approval.
- *distrib/arrays* package
  - Software support needed.
  - Draft specification exists for *distrib*.
  - Old proposal for *arrays* exists.
  - Please come to our breakout on this topic for further discussion.