CellDesigner Tutorial

Akira Funahashi
Keio University
The Systems Biology Institute
22nd Aug. 2008

Overview

- Introduction of CellDesigner
  - SBML (Systems Biology Markup Language)
  - SBGN (Graphical Notation)
- How to build a model with CellDesigner
- How to create CellDesigner plugin

The Systems Biology Institute

Software Infrastructure

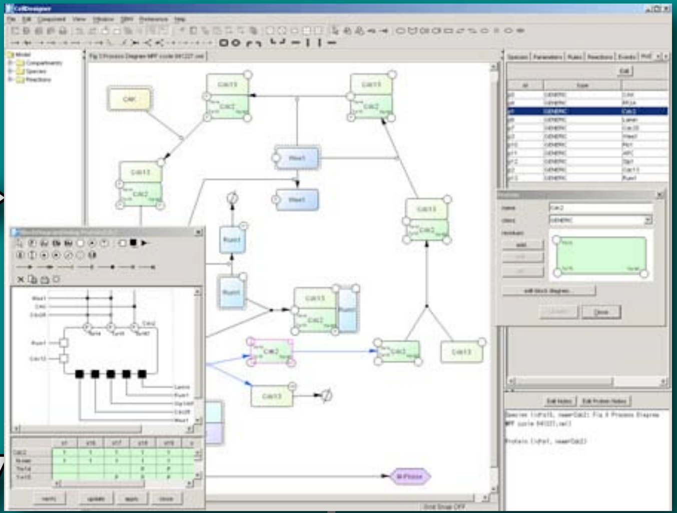
Model representation

Standard representation method of biological models

SBML Systems Biology Markup Language

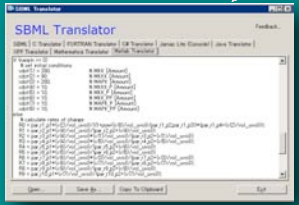
SBGN Systems Biology Graphical Notation

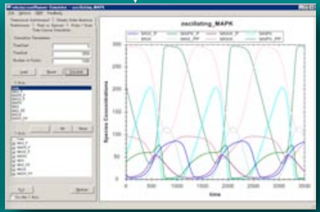
Software tools

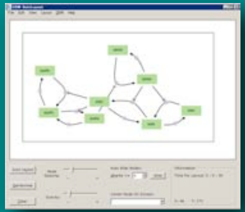


CellDesigner

SBW Systems Biology Workbench







Translator RoadRunner AutoLayout

Database

SABIO

KEGG Kyoto Encyclopedia of Genes and Genomes

PubMed

BioModels

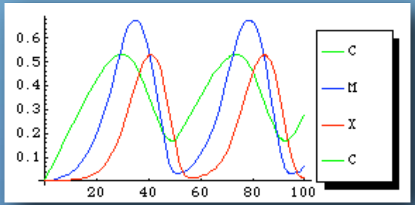
SGD

iHOP information hyperlinked over proteins

The Systems Biology Institute

CellDesigner

SBML Systems Biology Markup Language + SBGN Systems Biology Graphical Notation +



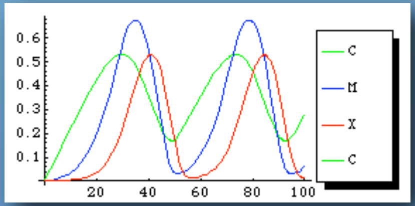
+ SGD KEGG GO iHOP PubMed SABIO BioModels

=

The Systems Biology Institute

CellDesigner

SBML Systems Biology Markup Language + SBGN Systems Biology Graphical Notation +



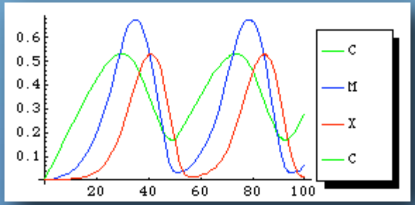
+ SGD KEGG GO iHOP PubMed SABIO BioModels

= CellDesigner

SBI The Systems Biology Institute

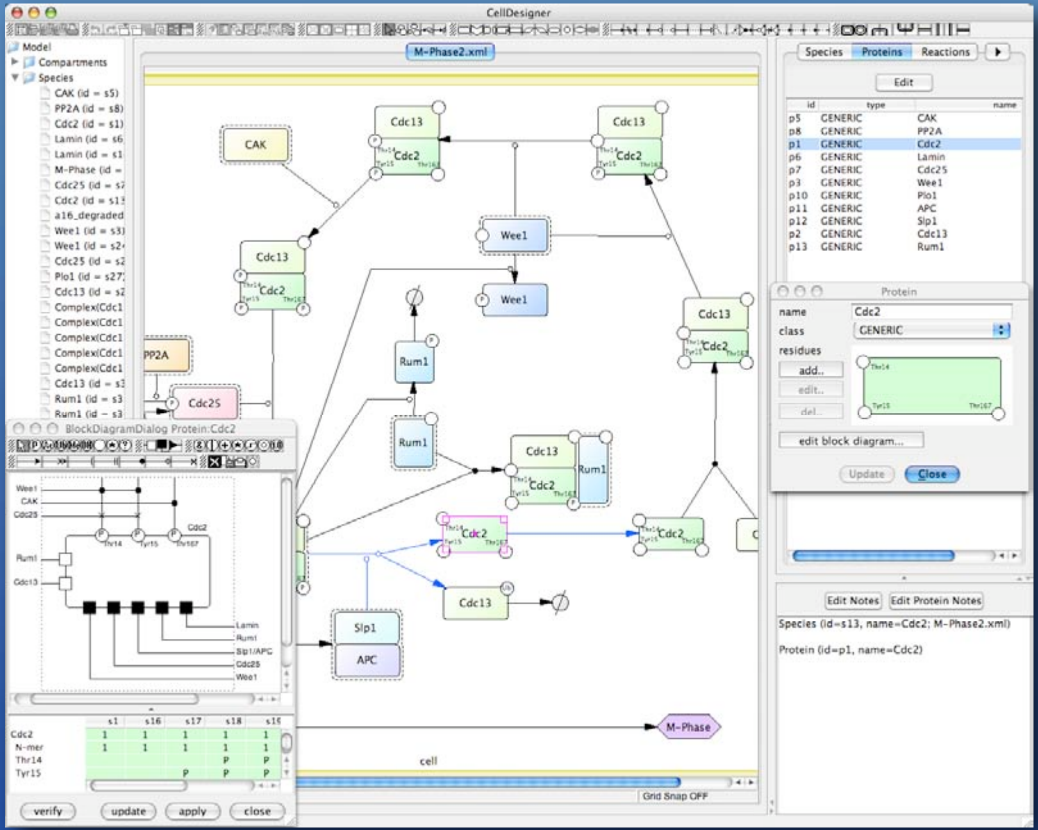
CellDesigner

SBML Systems Biology Markup Language + SBGN Systems Biology Graphical Notation +



+ SGD KEGG GO iHOP PubMed SABIO BioModels

= CellDesigner



id	type	name
p5	GENERIC	CAK
p8	GENERIC	PP2A
p1	GENERIC	Cdc2
p6	GENERIC	Lamin
p7	GENERIC	Cdc25
p3	GENERIC	Wee1
p10	GENERIC	Plo1
p11	GENERIC	APC
p12	GENERIC	Slp1
p2	GENERIC	Cdc13
p13	GENERIC	Rum1

name: Cdc2
class: GENERIC
residues

Species (id=s13, name=Cdc2; M-Phase2.xml)

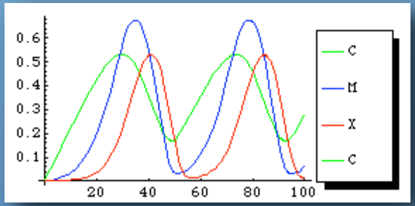
Protein (id=p1, name=Cdc2)

Modeling tool for biochemical and gene-regulatory network

SBI The Systems Biology Institute

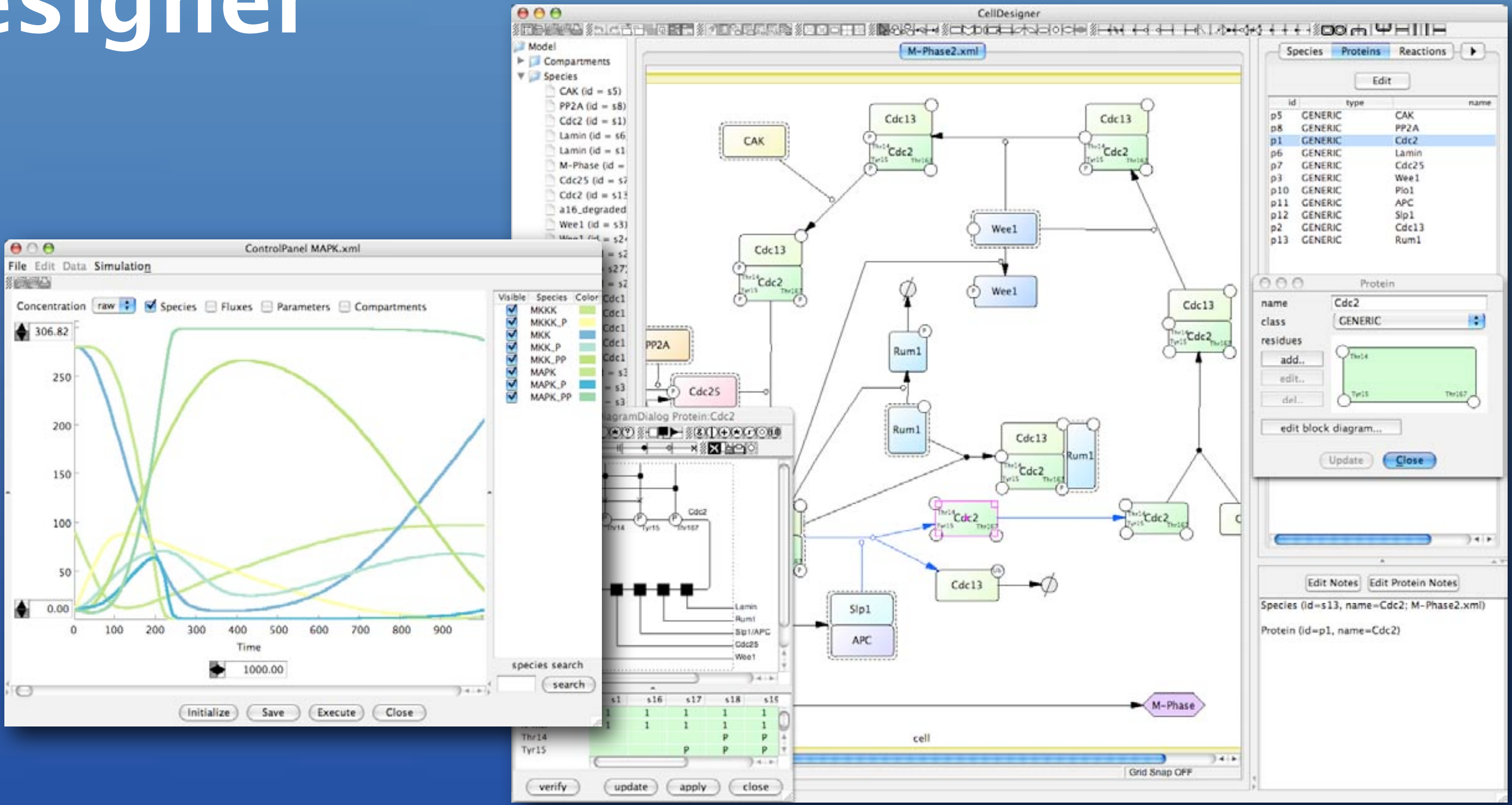
CellDesigner

SBML Systems Biology Markup Language + SBGN Systems Biology Graphical Notation +



+ SGD KEGG iHOP PubMed SABIO BioModels

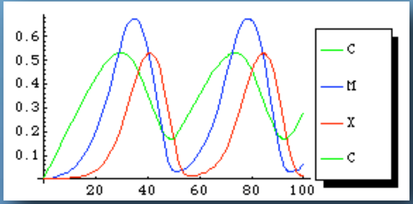
= CellDesigner



Modeling tool for biochemical and gene-regulatory network

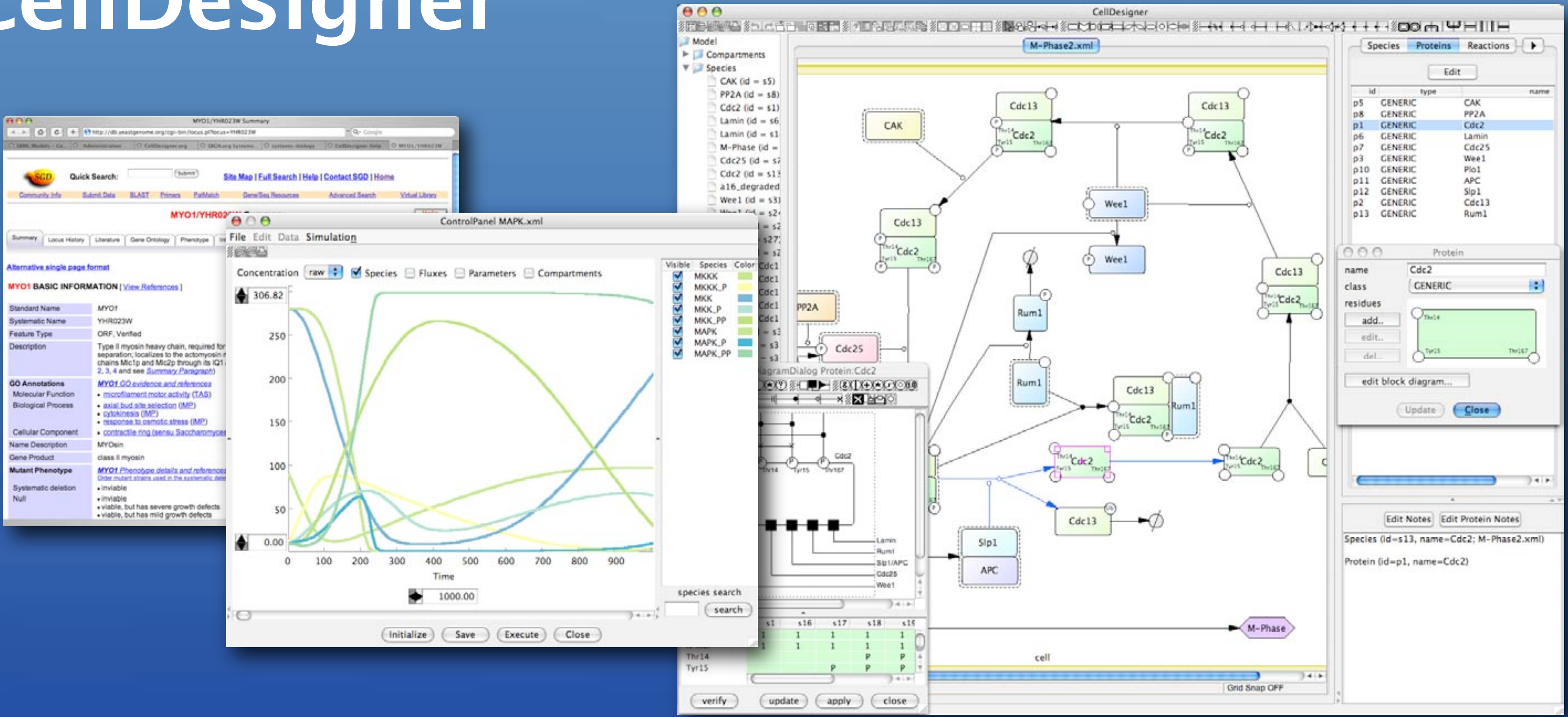
CellDesigner

SBML Systems Biology Markup Language + SBGN Systems Biology Graphical Notation +



+ SGD KEGG iHOP PubMed SABIO BioModels

= CellDesigner



MYO1/YHR023W Summary
SGD
Quick Search:
Site Map | Full Search | Help | Contact SGD | Home
MYO1/YHR023W Summary
Alternative single page format
MYO1 BASIC INFORMATION | View References |
ControlPanel MAPK.xml
File Edit Data Simulation
Concentration raw Species Fluxes Parameters Compartments
Time
Initialize Save Execute Close
CellDesigner
M-Phase2.xml
Species Proteins Reactions
Protein
name Cdc2
class GENERIC
Update Close
Edit Notes Edit Protein Notes
verify update apply close

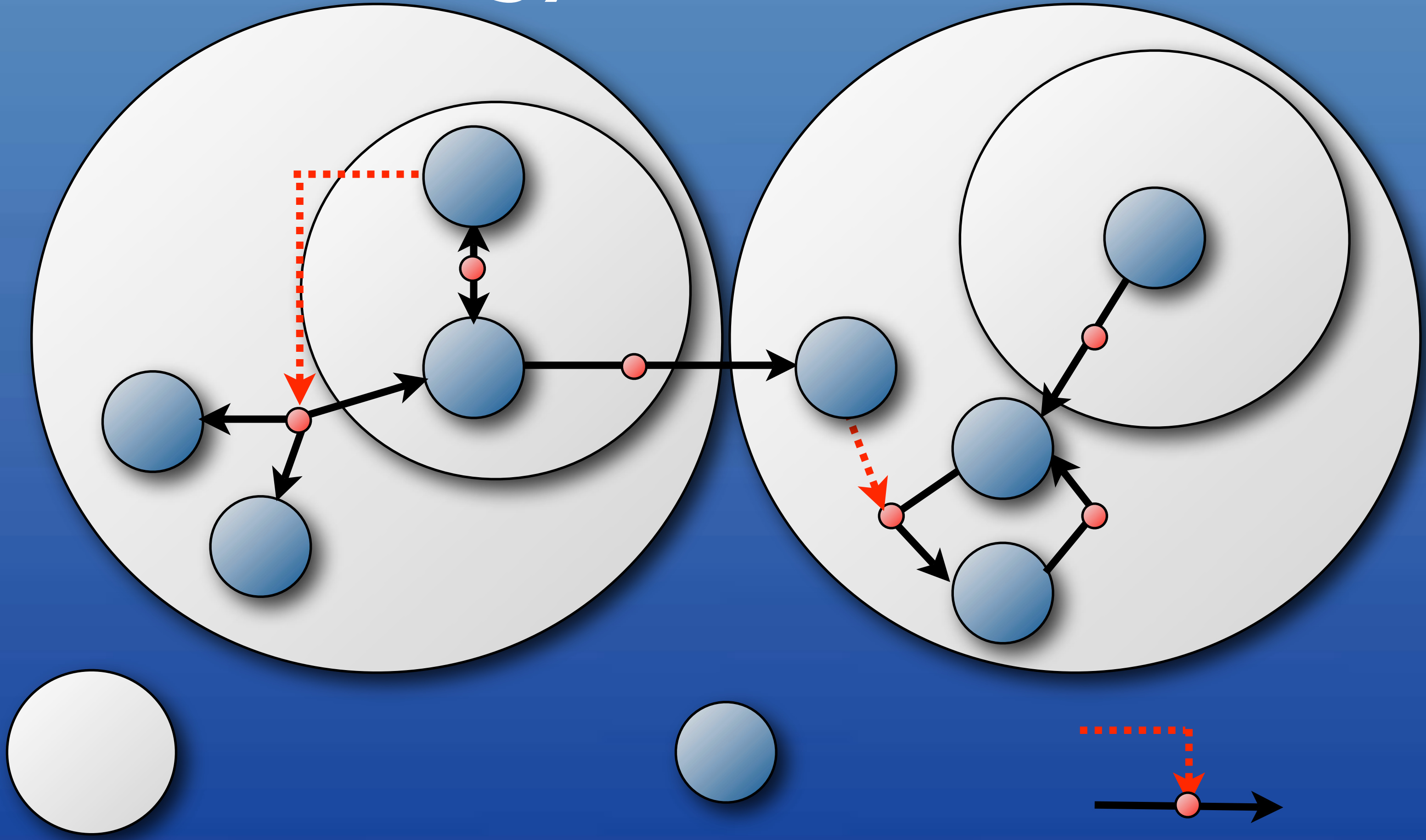
Standard Name MYO1
Systematic Name YHR023W
Feature Type ORF, Verified
Description Type II myosin heavy chain, required for ...

Species (id=s13, name=Cdc2; M-Phase2.xml)
Protein (id=p1, name=Cdc2)

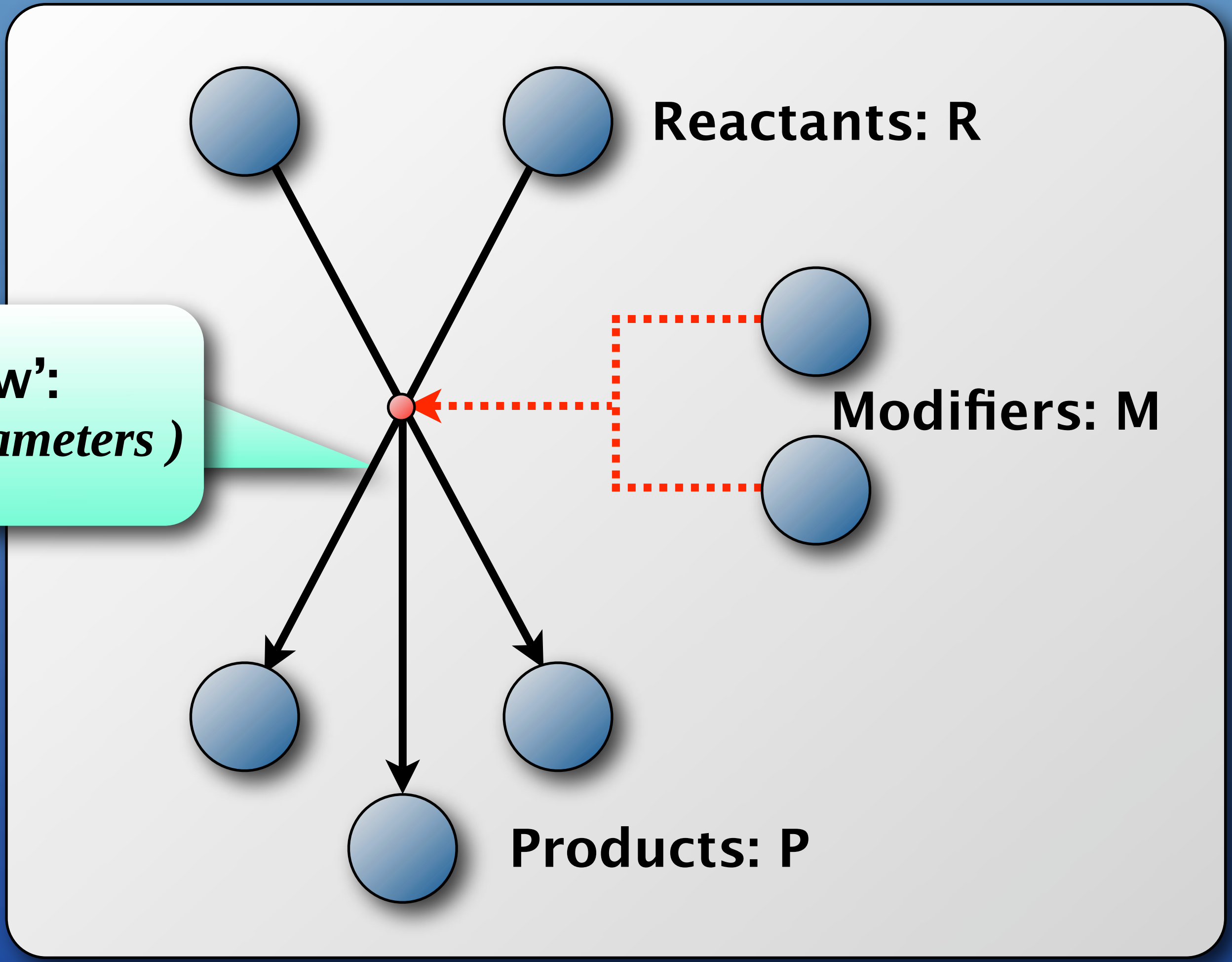
Modeling tool for biochemical and gene-regulatory network

SBML

- SBML (Systems Biology Markup Language)
- A machine-readable format (XML) for representing computational models in systems biology



Reactions According to SBML



'Kinetic law':
$v = f(\,R, P, M, \text{parameters}\,)$

The Systems Biology Institute

What does SBML look like?

Biochemical reaction

SBML

What does SBML look like?

Biochemical reaction

SBML

What does SBML look like?

Biochemical reaction

SBML

What does SBML look like?

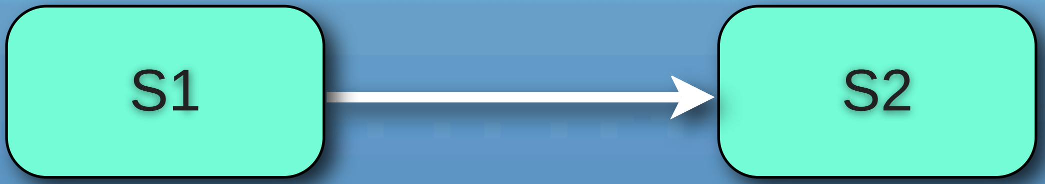
Biochemical reaction

```
<listOfSpecies>
  <species id="s1" name="s1" compartment="default"
initialAmount="0" charge="0"/>
  <species id="s2" name="s2" compartment="default"
initialAmount="0" charge="0"/>
</listOfSpecies>
```

SBML

What does SBML look like?

Biochemical reaction

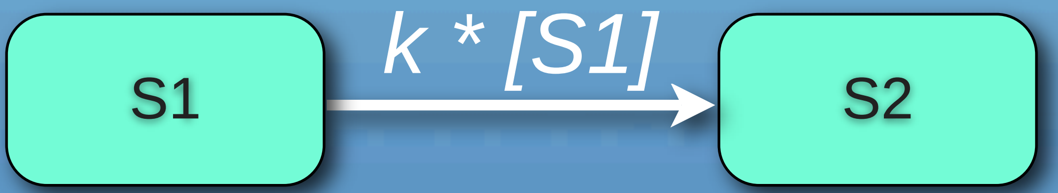


```
<listOfSpecies>
  <species id="s1" name="s1" compartment="default"
initialAmount="0" charge="0"/>
  <species id="s2" name="s2" compartment="default"
initialAmount="0" charge="0"/>
</listOfSpecies>
```

SBML

What does SBML look like?

Biochemical reaction

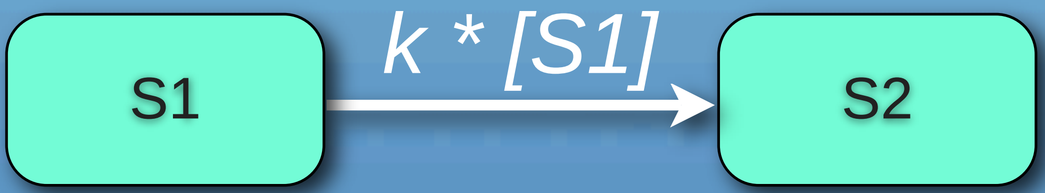


```
<listOfSpecies>
  <species id="s1" name="s1" compartment="default"
initialAmount="0" charge="0"/>
  <species id="s2" name="s2" compartment="default"
initialAmount="0" charge="0"/>
</listOfSpecies>
```

SBML

What does SBML look like?

Biochemical reaction

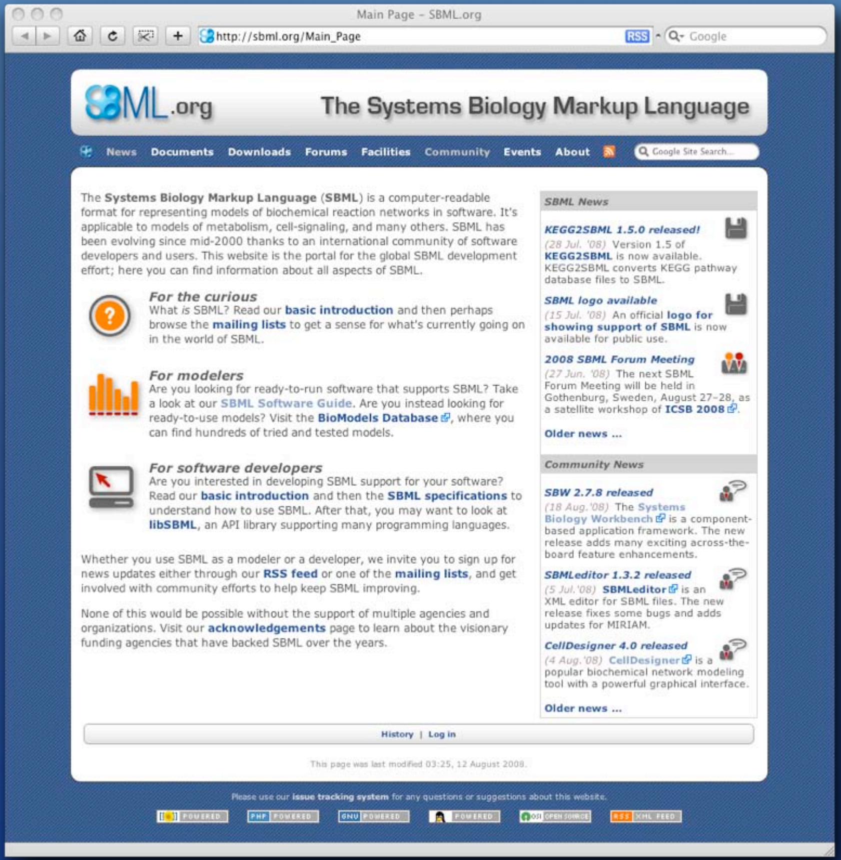


```
<listOfSpecies>
  <species id="s1" name="s1" compartment="default"
initialAmount="0" charge="0"/>
  <species id="s2" name="s2" compartment="default"
initialAmount="0" charge="0"/>
</listOfSpecies>
<listOfReactions>
  <reaction id="re1" reversible="false" fast="false">
  <listOfReactants>
    <speciesReference species="s1"/>
  </listOfReactants>
  <listOfProducts>
    <speciesReference species="s2"/>
  </listOfProducts>
  </reaction>
  <kineticLaw formula=”k*s1”>
  </kineticLaw>
</listOfReactions>
```

SBML

Applications Supporting SBML

- Over 130 software packages support SBML
- http://sbml.org



Main Page – SBML.org

http://sbml.org/Main_Page

SBML.org The Systems Biology Markup Language

News Documents Downloads Forums Facilities Community Events About

The **Systems Biology Markup Language** (**SBML**) is a computer-readable format for representing models of biochemical reaction networks in software. It's applicable to models of metabolism, cell-signaling, and many others. SBML has been evolving since mid-2000 thanks to an international community of software developers and users. This website is the portal for the global SBML development effort; here you can find information about all aspects of SBML.

For the curious
What *is* SBML? Read our **basic introduction** and then perhaps browse the **mailing lists** to get a sense for what's currently going on in the world of SBML.

For modelers
Are you looking for ready-to-run software that supports SBML? Take a look at our **SBML Software Guide**. Are you instead looking for ready-to-use models? Visit the **BioModels Database**, where you can find hundreds of tried and tested models.

For software developers
Are you interested in developing SBML support for your software? Read our **basic introduction** and then the **SBML specifications** to understand how to use SBML. After that, you may want to look at **libSBML**, an API library supporting many programming languages.

Whether you use SBML as a modeler or a developer, we invite you to sign up for news updates either through our **RSS feed** or one of the **mailing lists**, and get involved with community efforts to help keep SBML improving.

None of this would be possible without the support of multiple agencies and organizations. Visit our **acknowledgements** page to learn about the visionary funding agencies that have backed SBML over the years.

SBML News

KEGG2SBML 1.5.0 released!
(28 Jul. '08) Version 1.5 of **KEGG2SBML** is now available. KEGG2SBML converts KEGG pathway database files to SBML.

SBML logo available
(15 Jul. '08) An official **logo for showing support of SBML** is now available for public use.

2008 SBML Forum Meeting
(27 Jun. '08) The next SBML Forum Meeting will be held in Gothenburg, Sweden, August 27–28, as a satellite workshop of **ICSB 2008**.

Older news ...

Community News

SBW 2.7.8 released
(18 Aug. '08) The **Systems Biology Workbench** is a component-based application framework. The new release adds many exciting across-the-board feature enhancements.

SBMLeditor 1.3.2 released
(5 Jul. '08) **SBMLeditor** is an XML editor for SBML files. The new release fixes some bugs and adds updates for MIRIAM.

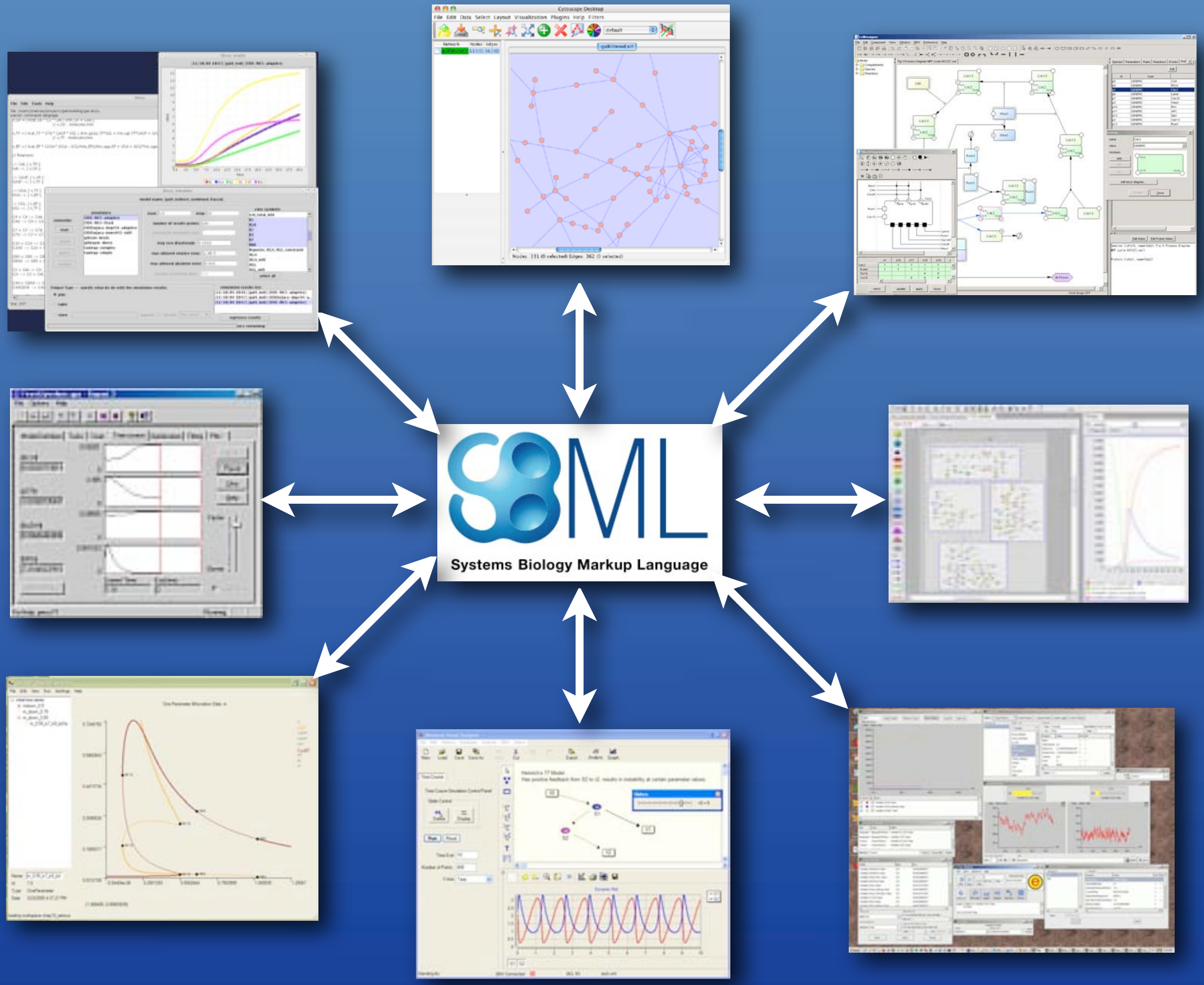
CellDesigner 4.0 released
(4 Aug. '08) **CellDesigner** is a popular biochemical network modeling tool with a powerful graphical interface.

Older news ...

History | Log in

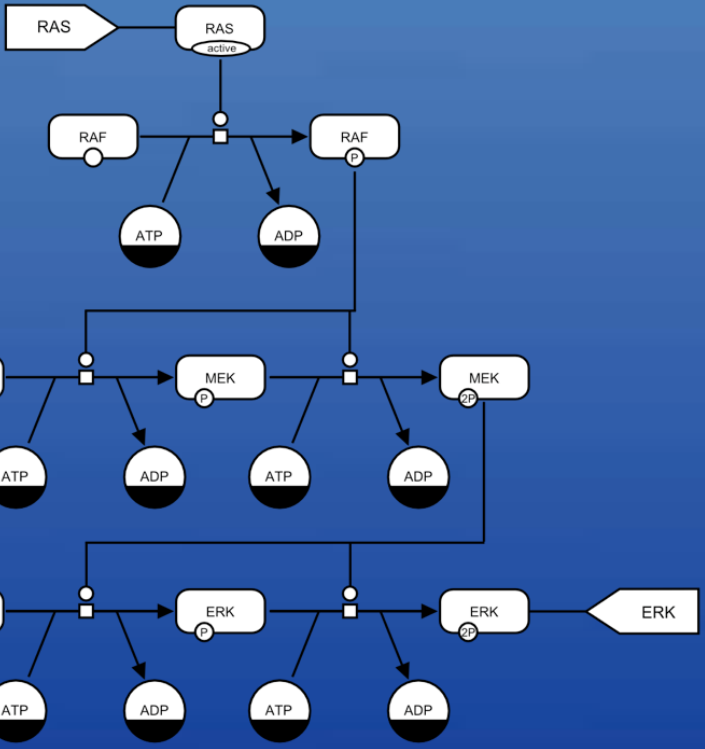
This page was last modified 03:25, 12 August 2008.

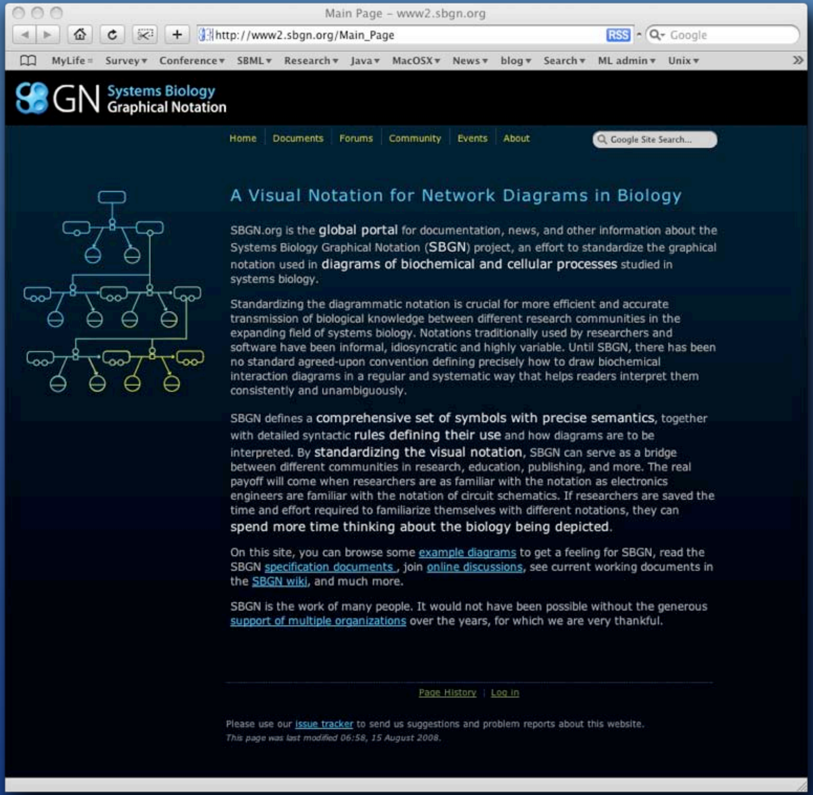
Please use our **issue tracking system** for any questions or suggestions about this website.



SBGN

- A Visual Notation for Network Diagrams in Biology
- Representation of Biochemical and Cellular Processes studied in Systems Biology





A Visual Notation for Network Diagrams in Biology

SBGN.org is the **global portal** for documentation, news, and other information about the Systems Biology Graphical Notation (**SBGN**) project, an effort to standardize the graphical notation used in **diagrams of biochemical and cellular processes** studied in systems biology.

Standardizing the diagrammatic notation is crucial for more efficient and accurate transmission of biological knowledge between different research communities in the expanding field of systems biology. Notations traditionally used by researchers and software have been informal, idiosyncratic and highly variable. Until SBGN, there has been no standard agreed-upon convention defining precisely how to draw biochemical interaction diagrams in a regular and systematic way that helps readers interpret them consistently and unambiguously.

SBGN defines a **comprehensive set of symbols with precise semantics**, together with detailed syntactic **rules defining their use** and how diagrams are to be interpreted. By **standardizing the visual notation**, SBGN can serve as a bridge between different communities in research, education, publishing, and more. The real payoff will come when researchers are as familiar with the notation as electronics engineers are familiar with the notation of circuit schematics. If researchers are saved the time and effort required to familiarize themselves with different notations, they can **spend more time thinking about the biology being depicted**.

On this site, you can browse some example diagrams to get a feeling for SBGN, read the SBGN specification documents , join online discussions, see current working documents in the SBGN wiki, and much more.

SBGN is the work of many people. It would not have been possible without the generous support of multiple organizations over the years, for which we are very thankful.

Page History | Log in

Please use our issue tracker to send us suggestions and problem reports about this website.
This page was last modified 06:58, 15 August 2008.

http://sbgn.org

SBGN community

- BioModels Database (UK)
- BioNetGen (USA)
- BioPAX
- BioUML (Russia)
- CellDesigner (Japan)
- CellML (New Zealand)
- COPASI (Germany)
- Cytoscape (USA)
- Design Suite (USA)
- EPE, EPN (UK)
- INOH (Japan)
- JDesigner (USA)
- Narrator (UK)
- NetBuilder
- Panther (USA)
- ProcessDB
- ProMot (Germany)
- QBT (USA)
- SABIO-RK (Germany)
- SBML Layout extension
- Taverna (UK)
- VCell (USA)

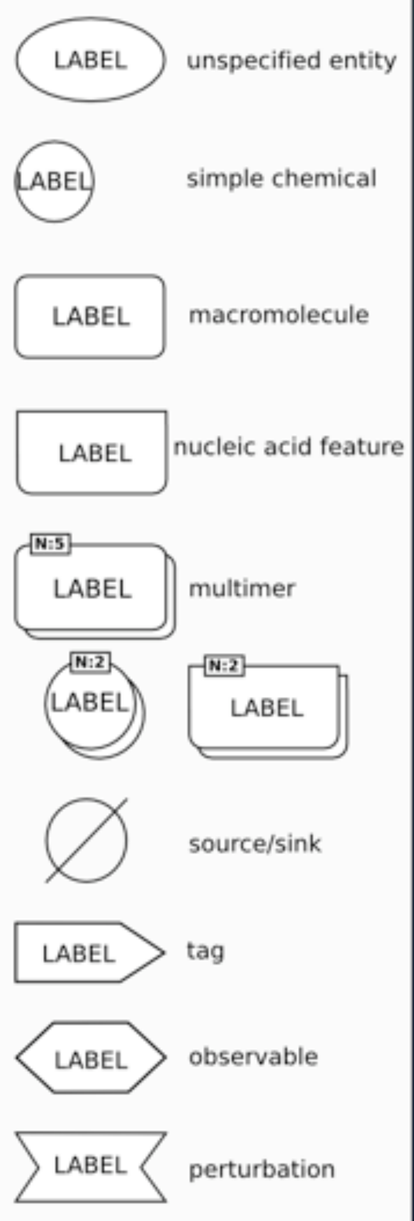
And more...

SBI The Systems Biology Institute

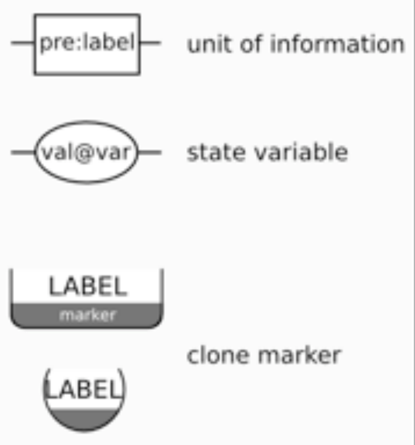
SBGN Process Diagram Level-1

SYSTEMS BIOLOGY GRAPHICAL NOTATION REFERENCE CARD

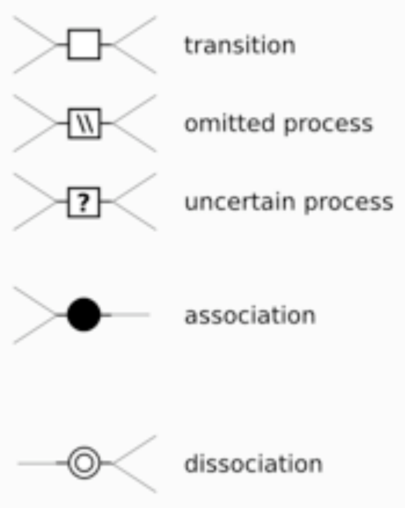
Entity Pool Nodes



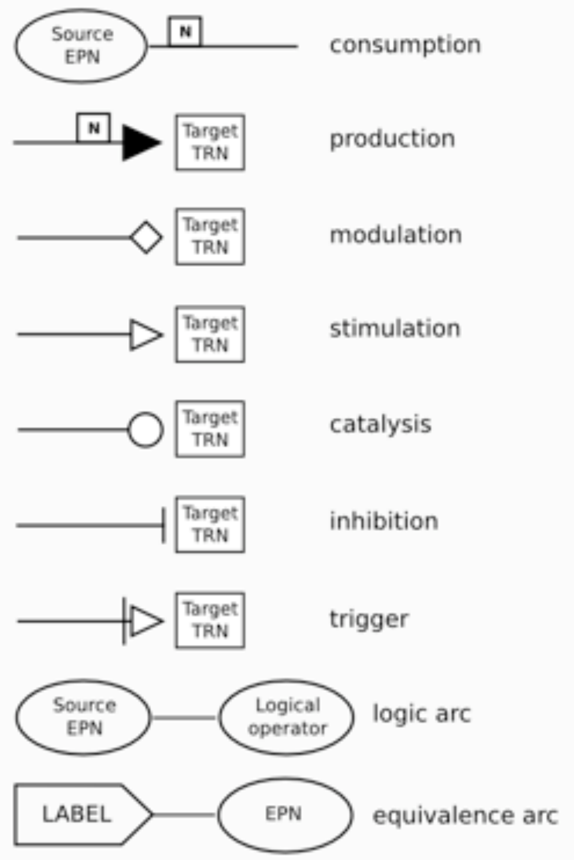
Auxiliary units



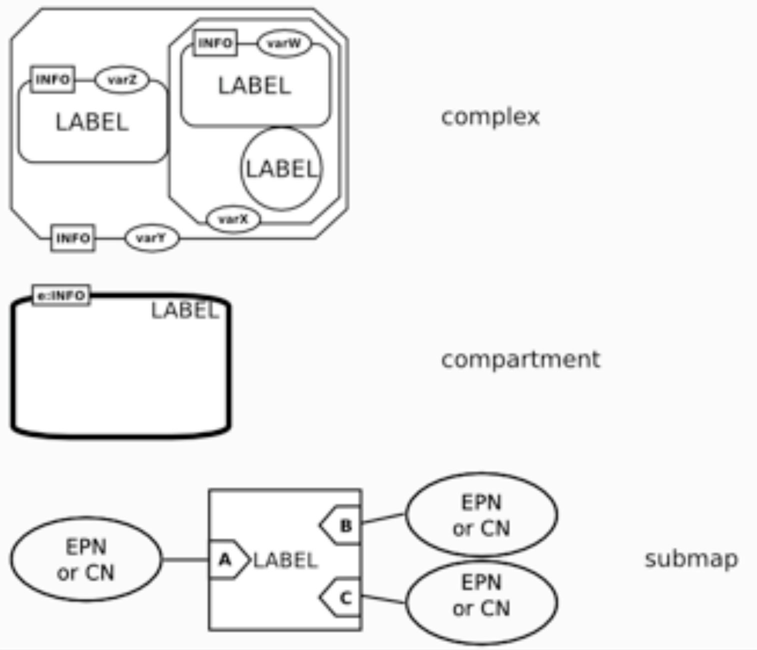
Process Nodes



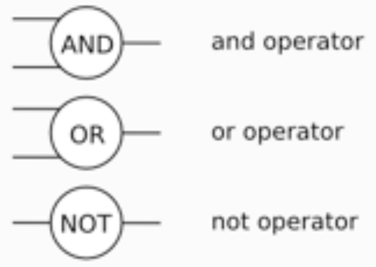
Connecting Arcs



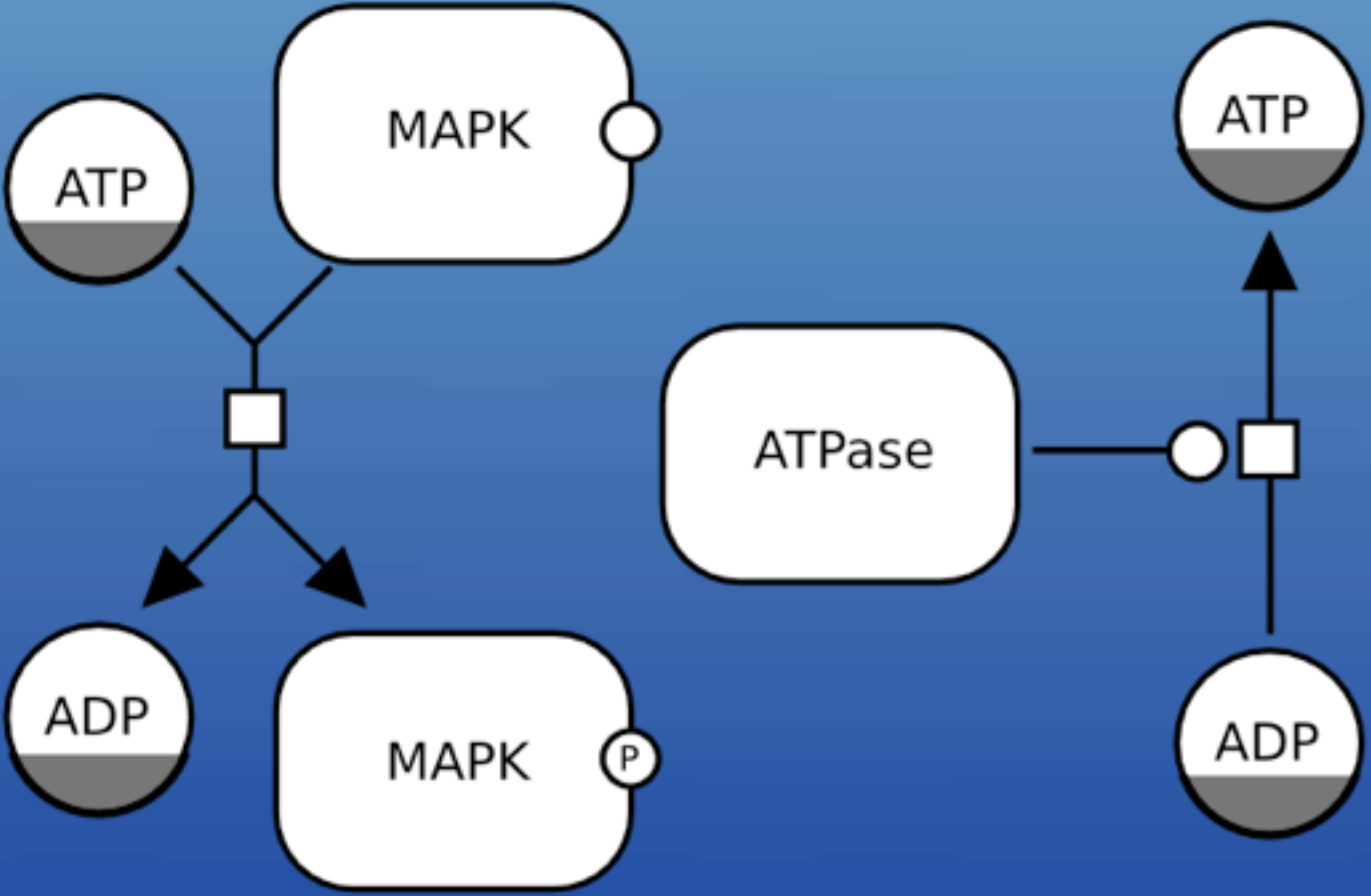
Container Nodes



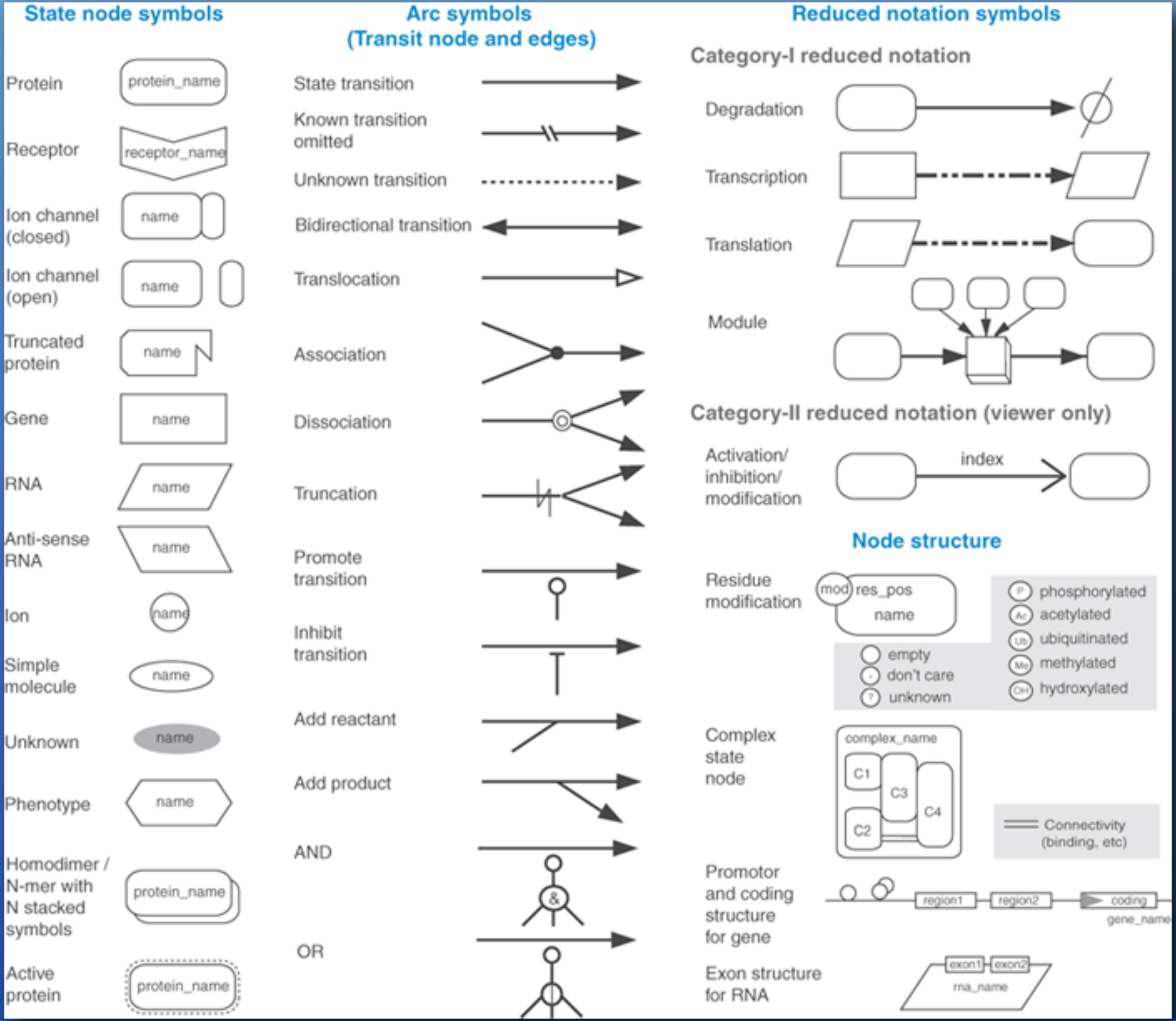
Logical Operators



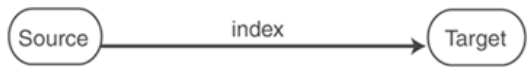
SBGN Process Diagram Level-1



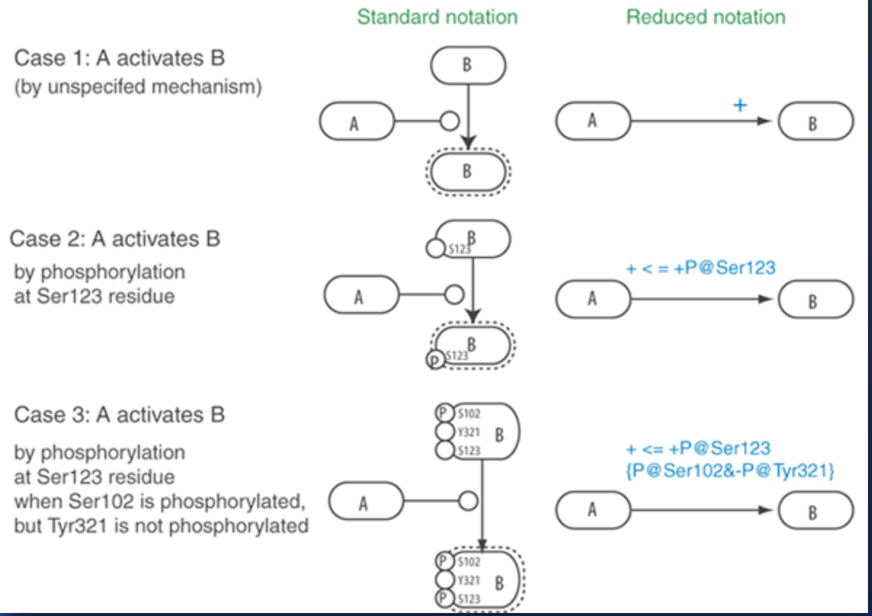
CellDesigner Notation



Syntax for index on category-II reduced notation



```
EffectDescription = Result ImmediateEffect Condition | SimpleSentence

    Result              = TRANSITION ('+' | '-' | empty) '<=' | empty
    ImmediateEffect     = TERM_IE
    Condition           = empty | '{' TERM_COND '}'
    SimpleSentence      = ('+' | '-' | TRANSITION | '?')

    TERM_IE             = RESIDUE_IE | TERM_IE OP TERM_IE
    TERM_COND           = RESIDUE_COND | TERM_COND OP TERM_COND
    RESIDUE_IE          = ('+' | '-') MODIFICATION ('@' TYPE [0-9]+ SUBUNIT | empty)
    RESIDUE_COND        = ('-' | empty) MODIFICATION '@' TYPE [0-9]+
    OP                  = ('&' | '|')
    MODIFICATION        = (P | Me | Ac | Ub | Hy) | (P | M | A | U | H)
    TYPE                = (Tyr | Ser | Thr) | (Y | S | T)
    SUBUNIT             = empty | '/' SUBUNIT_NAME
    TRANSITION          = [a-zA-Z]+[0-9]*
    SUBUNIT_NAME        = [a-zA-Z0-9]+
```



Standard notation
Reduced notation
Case 1: A activates B (by unspecifed mechanism)
A
B
+
Case 2: A activates B by phosphorylation at Ser123 residue
S123
P
+ <= +P@Ser123
Case 3: A activates B by phosphorylation at Ser123 residue when Ser102 is phosphorylated, but Tyr321 is not phosphorylated
S102
Y321
S123
+ <= +P@Ser123 {P@Ser102&-P@Tyr321}

Kitano, H. et al. "Using process diagrams for the graphical representation of biological networks", *Nature Biotechnology* **23**(8), 961 - 966 (2005)

SBI The Systems Biology Institute

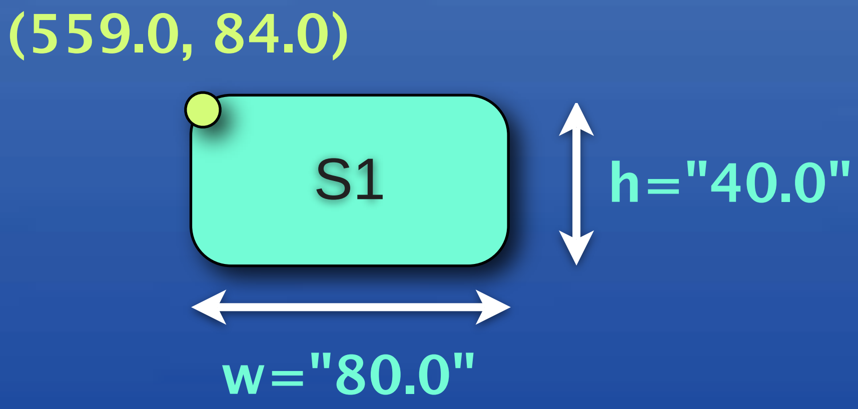
Graphical Notation ↔ SBML

- Species type, Reaction type is stored in <annotation> for each species, reactions
- Layout information is stored separately

```
<sbml>
  <model>
    <annotation>
      layout information
    </annotation>
    <listOfSpecies>
      <species>
        <annotation>species type</annotation>
      </species>
  </model>
</sbml>
```

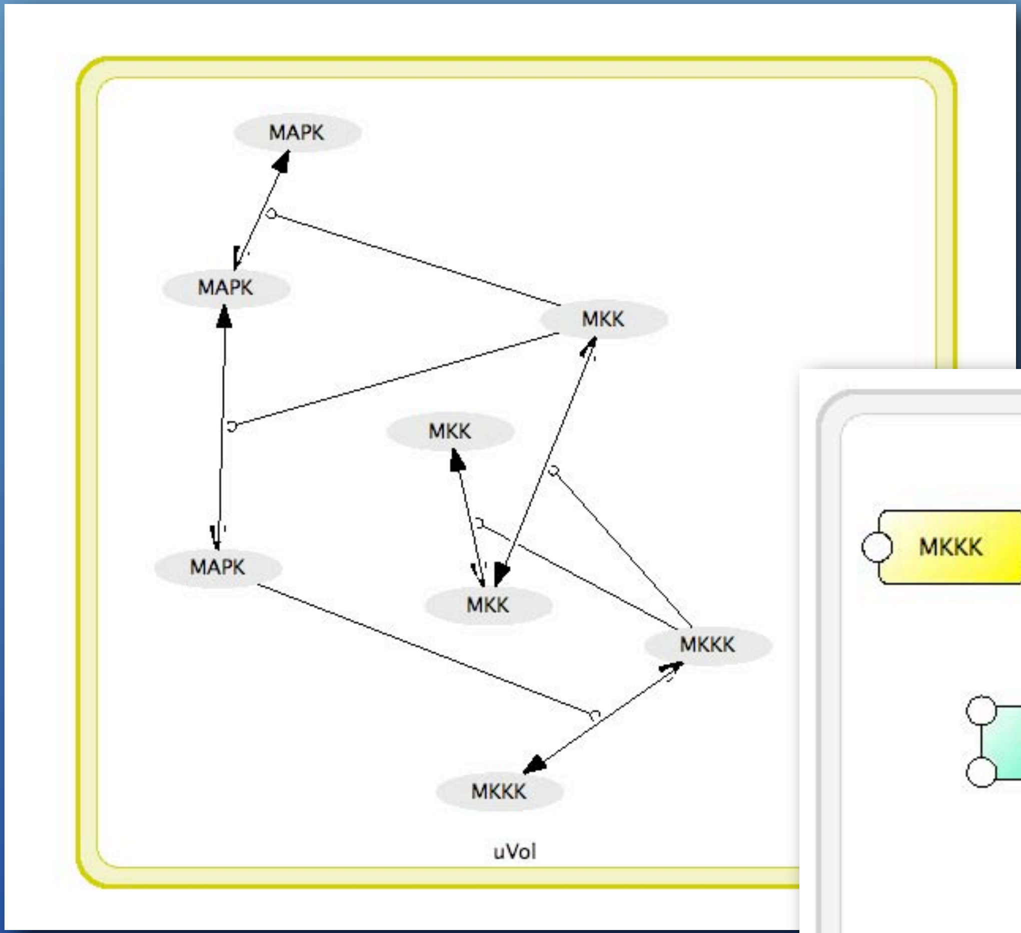
Graphical Notation ↔ SBML

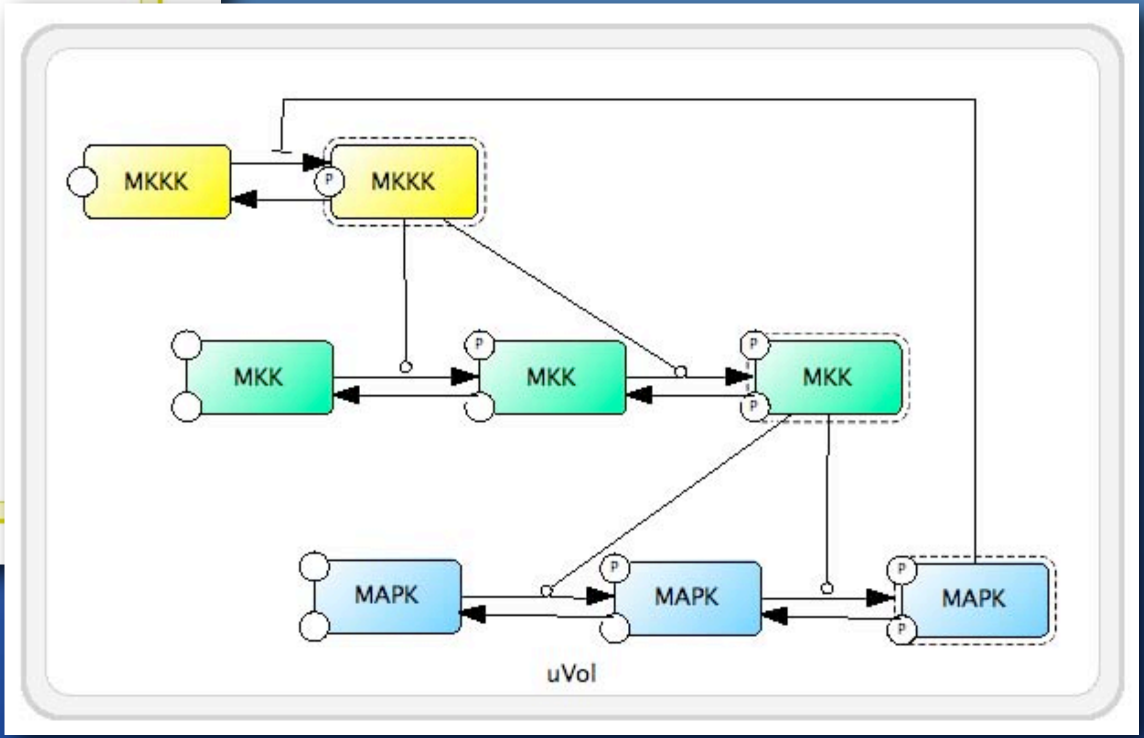
```
<celldesigner:speciesAlias compartmentAlias="ca3" id="a1" species="s1">
   <celldesigner:activity>active</celldesigner:activity>
   <celldesigner:bounds h="40.0" w="80.0" x="559.0" y="184.0">
   </celldesigner:bounds>
   <celldesigner:singleLine width="1.0"></celldesigner:singleLine>
   <celldesigner:paint color="ffb3d2ff" scheme="Gradation">
   </celldesigner:paint>
</celldesigner:speciesAlias>
```



SBML w/ or w/o Graphical Notation

Pure SBML (w/o Graphical Notation)

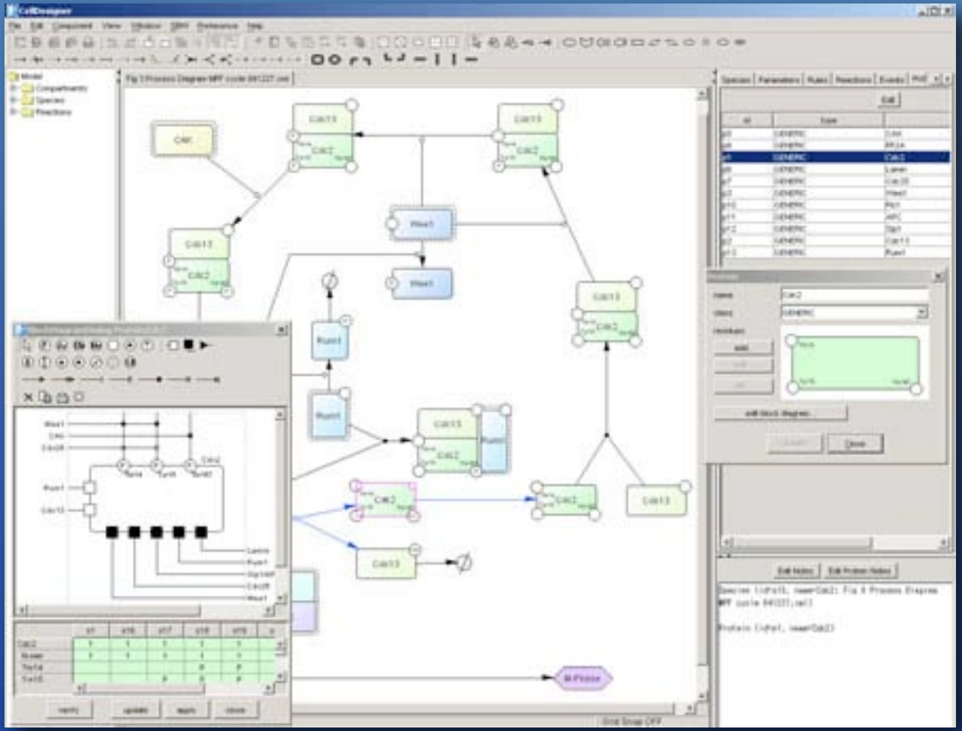




w/ Graphical Notation

CellDesigner 4.0.1

- SBML support
- Graphical notation (SBGN)
- Built-in simulator (SBML ODE Solver, COPASI)
- Integrate with Analysis tool, other simulators through SBW
- Database connection
- Export to PDF, PNG, etc.
- Freely available
- Supported Environment
  - Windows (XP or later)
  - Mac OS X (Tiger, Leopard)
  - Linux



http://celldesigner.org

What's new

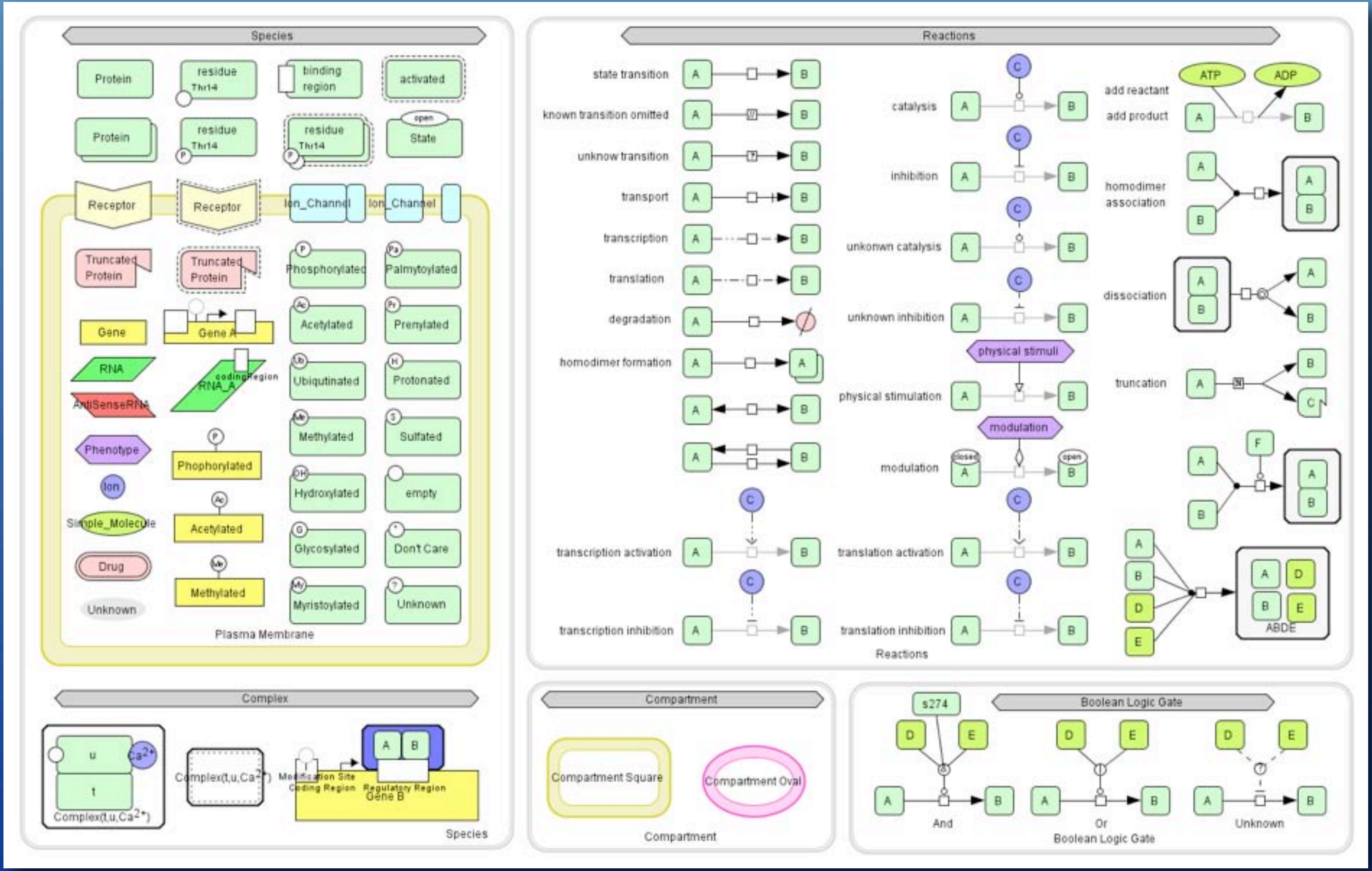
- Enhanced graphical notation (SBGN Level-1 draft)
- Integration with COPASI
- Plugin development framework
- GUI improvement
- Layer function
- libSBML 3

CellDesigner™
Ver. 4.0.1

(C) 2003 - 2008 The Systems Biology Institute. All rights reserved.
http://celldesigner.org

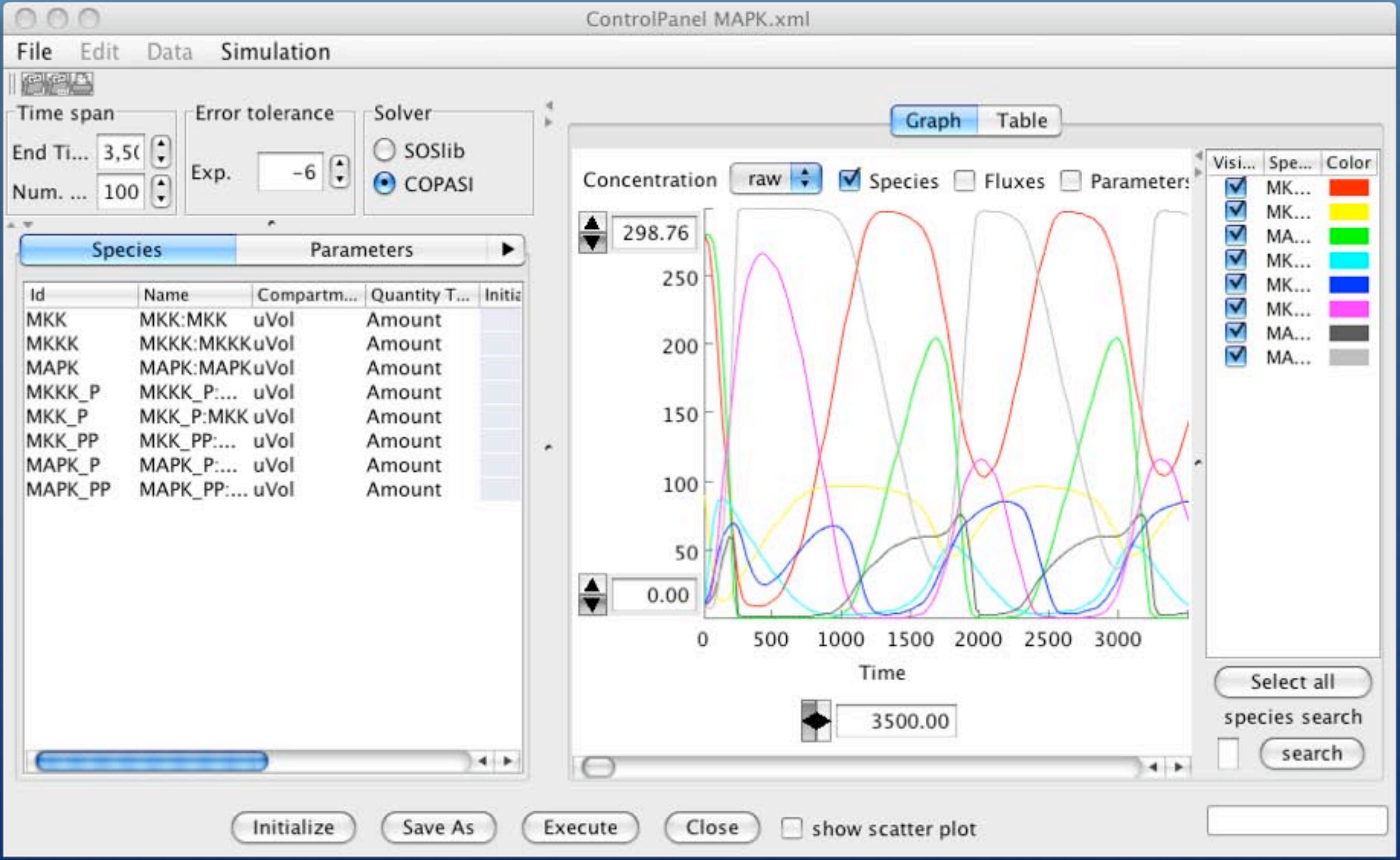
Enhanced Graphical Notation

- CellDesigner 4 supports SBGN Level-1 draft



Integration with COPASI

- Can call COPASI as a solver



Time span
End Ti... 3,5(
Num. ... 100

Error tolerance
Exp. -6

Solver
SOSlib
[✓ COPASI]

Id	Name	Compartm...	Quantity T...	Initia
MKK	MKK:MKK	uVol	Amount	
MKKK	MKKK:MKKK	uVol	Amount	
MAPK	MAPK:MAPK	uVol	Amount	
MKKK_P	MKKK_P:...	uVol	Amount	
MKK_P	MKK_P:MKK	uVol	Amount	
MKK_PP	MKK_PP:...	uVol	Amount	
MAPK_P	MAPK_P:...	uVol	Amount	
MAPK_PP	MAPK_PP:...	uVol	Amount	

Concentration raw ✓ Species Fluxes Parameters

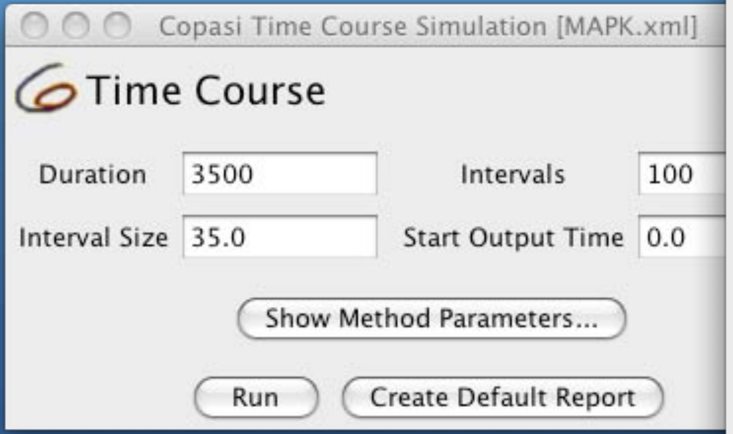
298.76
0.00
3500.00
Time

Visi...	Spe...	Color
✓	MK...	🟥
✓	MK...	🟨
✓	MA...	🟩
✓	MK...	🟦
✓	MK...	🟦
✓	MK...	🟪
✓	MA...	⬛
✓	MA...	⬜

show scatter plot

Integration with COPASI

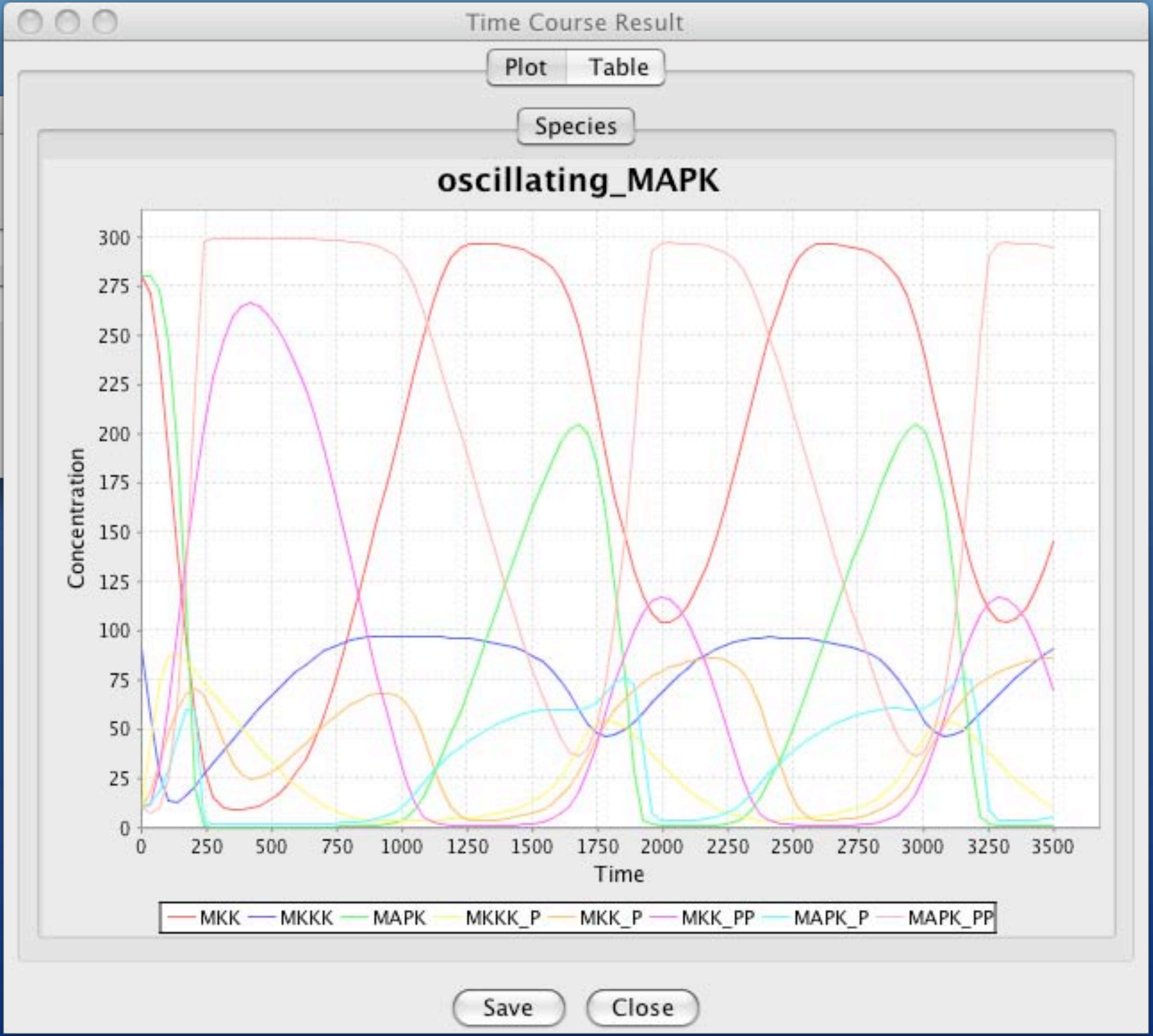
- Can call COPASI as a solver



Copasi Time Course Simulation [MAPK.xml]

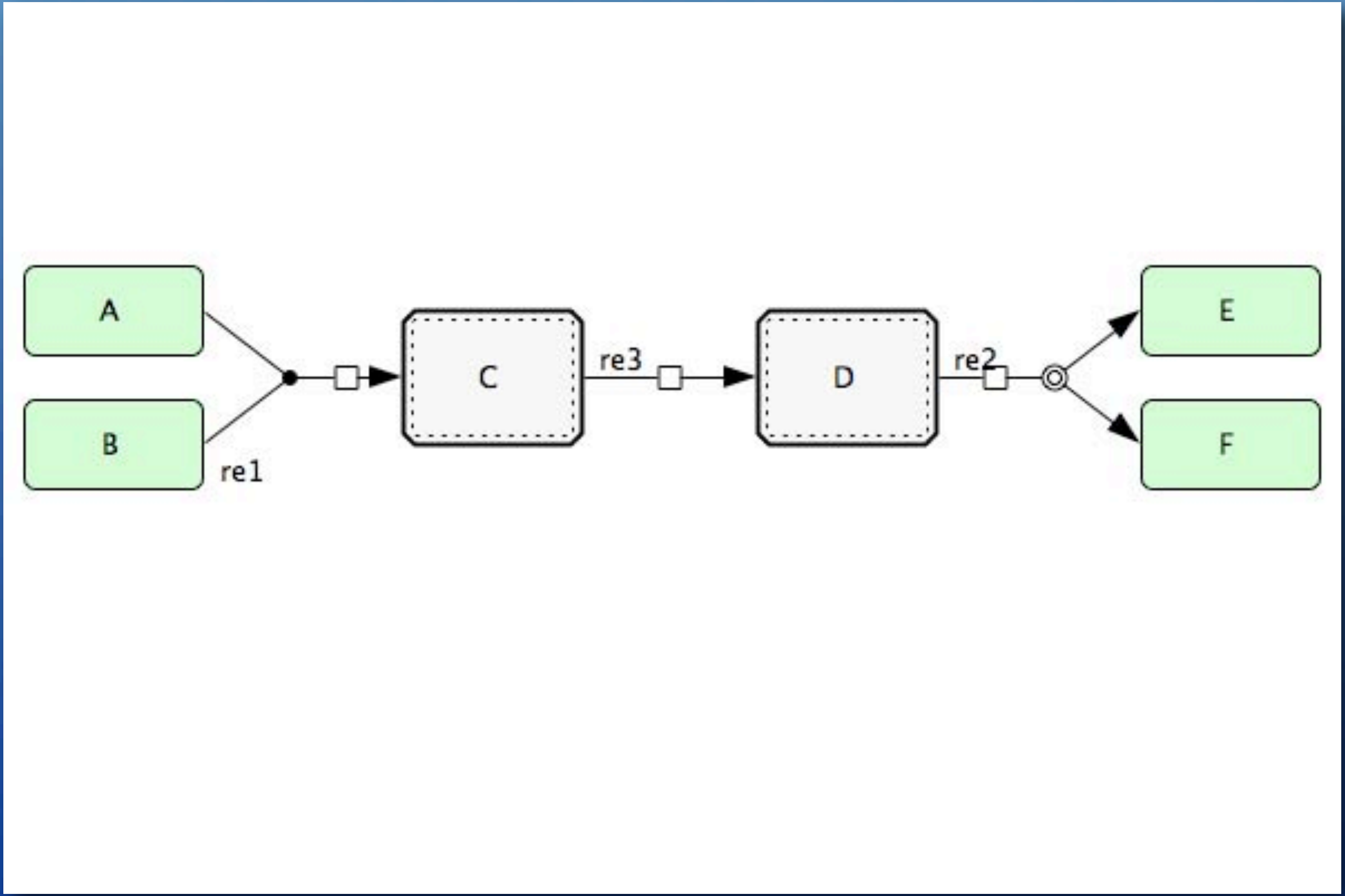
Time Course

Duration	3500	Intervals	100
Interval Size	35.0	Start Output Time	0.0



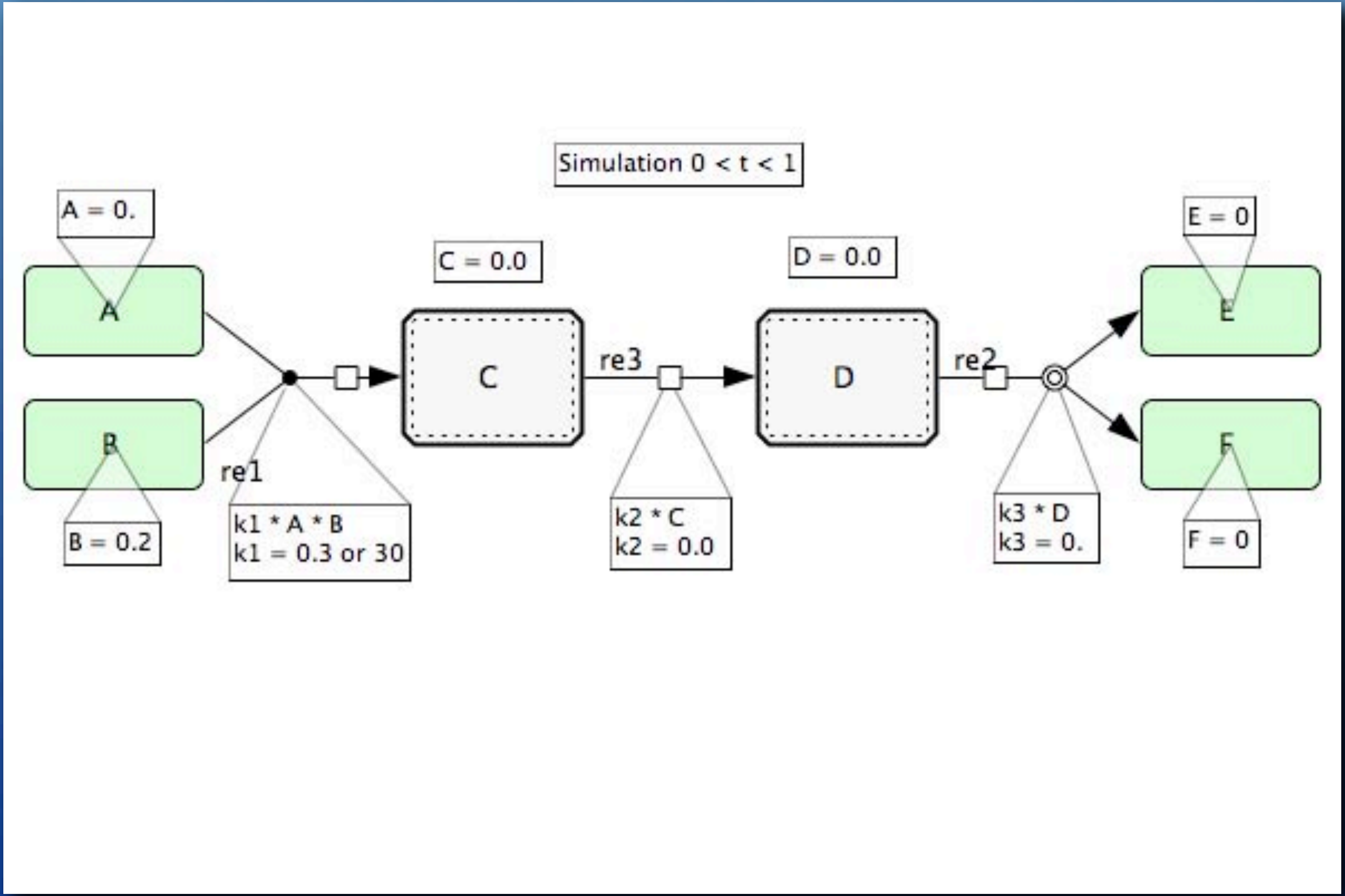
Layer function

- Add graphical / text object to your model



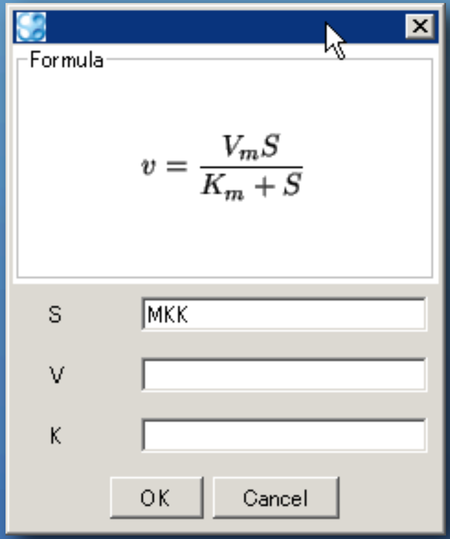
Layer function

- Add graphical / text object to your model



GUI improvement

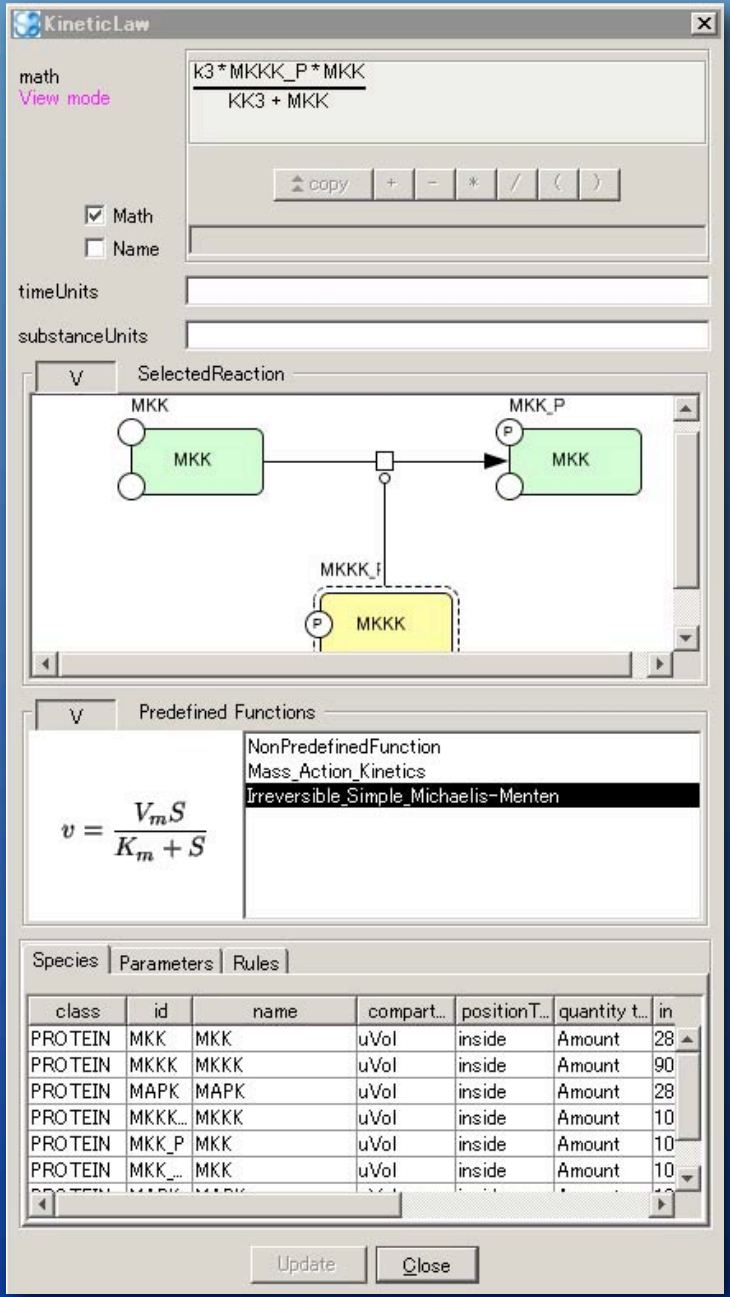
- Enhanced Kinetic Law Editor



Formula

$$v = \frac{V_m S}{K_m + S}$$

S	MKK
V	
K	



KineticLaw

math
View mode

$$\frac{k3*MKKK\_P*MKK}{KK3 + MKK}$$

[✓] Math
[ ] Name

timeUnits

substanceUnits

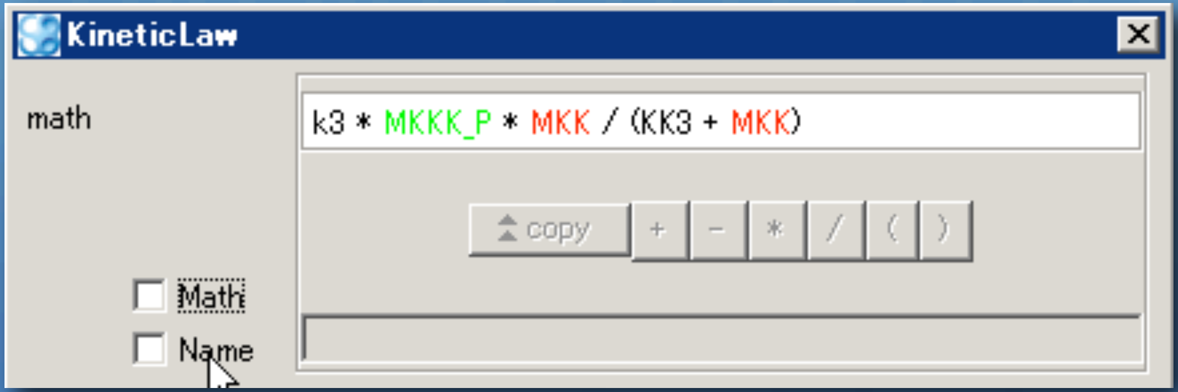
$$v = \frac{V_m S}{K_m + S}$$

NonPredefinedFunction
Mass_Action_Kinetics
Irreversible_Simple_Michaelis-Menten

class	id	name	compart...	positionT...	quantity t...	in
PROTEIN	MKK	MKK	uVol	inside	Amount	28
PROTEIN	MKKK	MKKK	uVol	inside	Amount	90
PROTEIN	MAPK	MAPK	uVol	inside	Amount	28
PROTEIN	MKKK...	MKKK	uVol	inside	Amount	10
PROTEIN	MKK_P	MKK	uVol	inside	Amount	10
PROTEIN	MKK_...	MKK	uVol	inside	Amount	10

GUI improvement

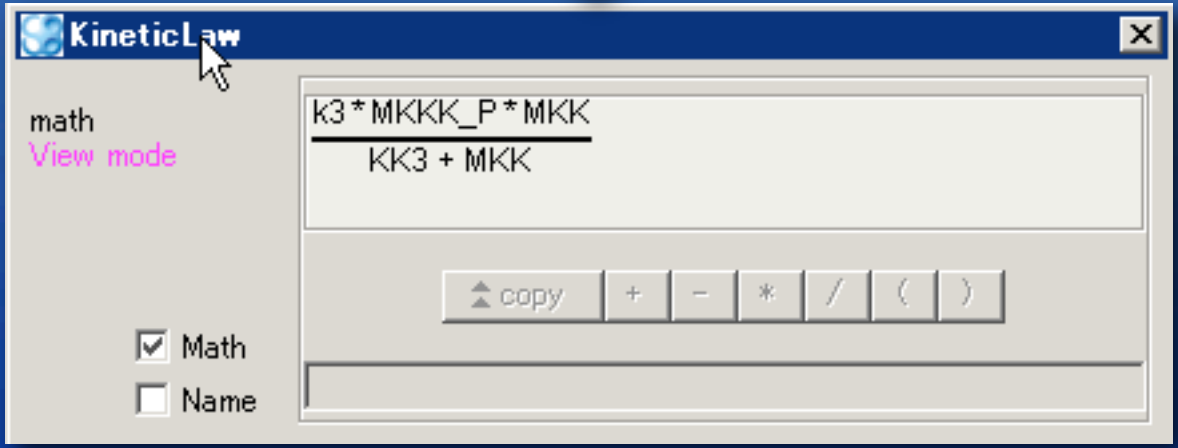
- Enhanced Kinetic Law Editor



KineticLaw

math: k3 * MKKK_P * MKK / (KK3 + MKK)

☐ Math
☐ Name



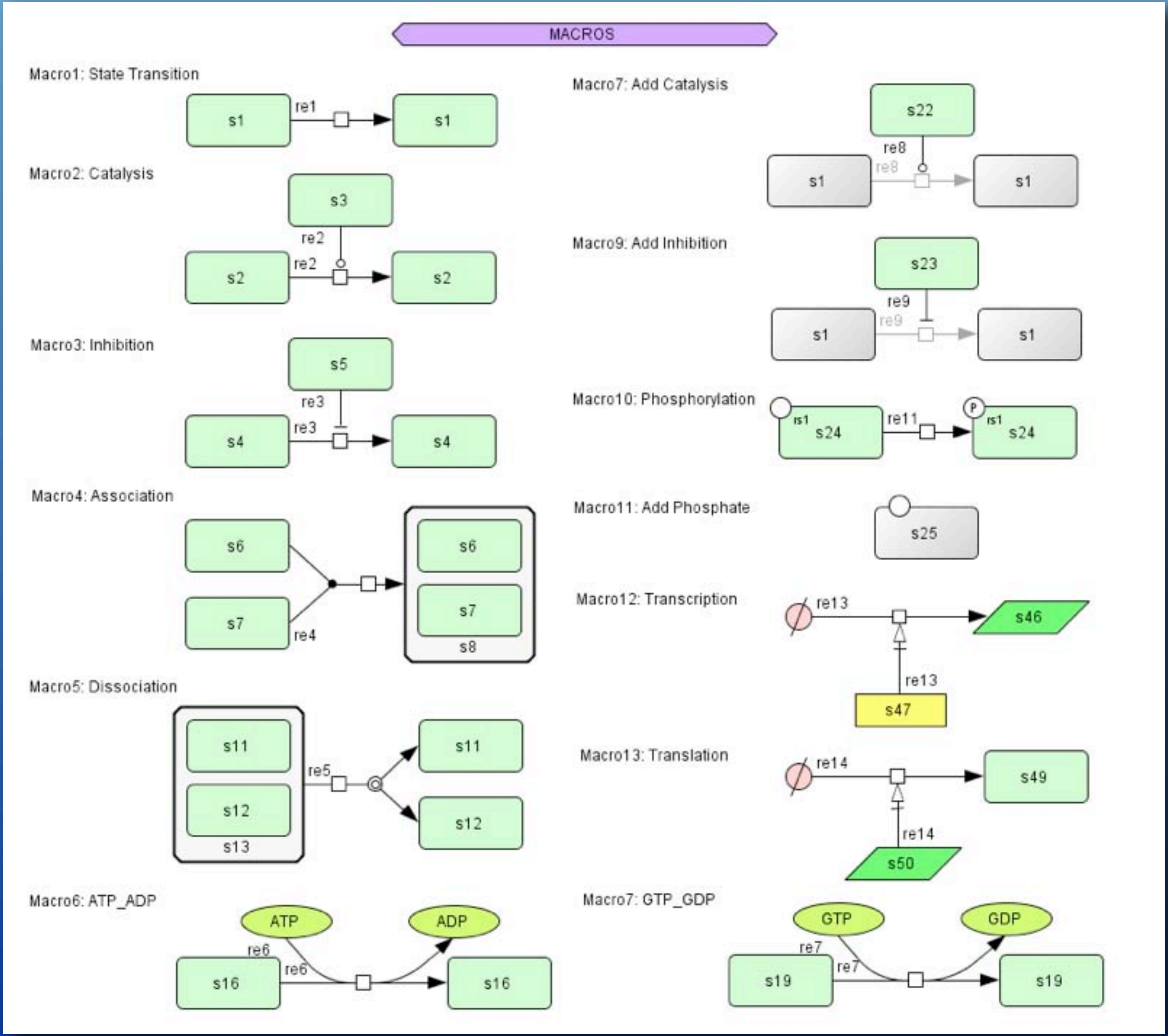
KineticLaw

math
View mode

$$\frac{k3 * MKKK\_P * MKK}{KK3 + MKK}$$

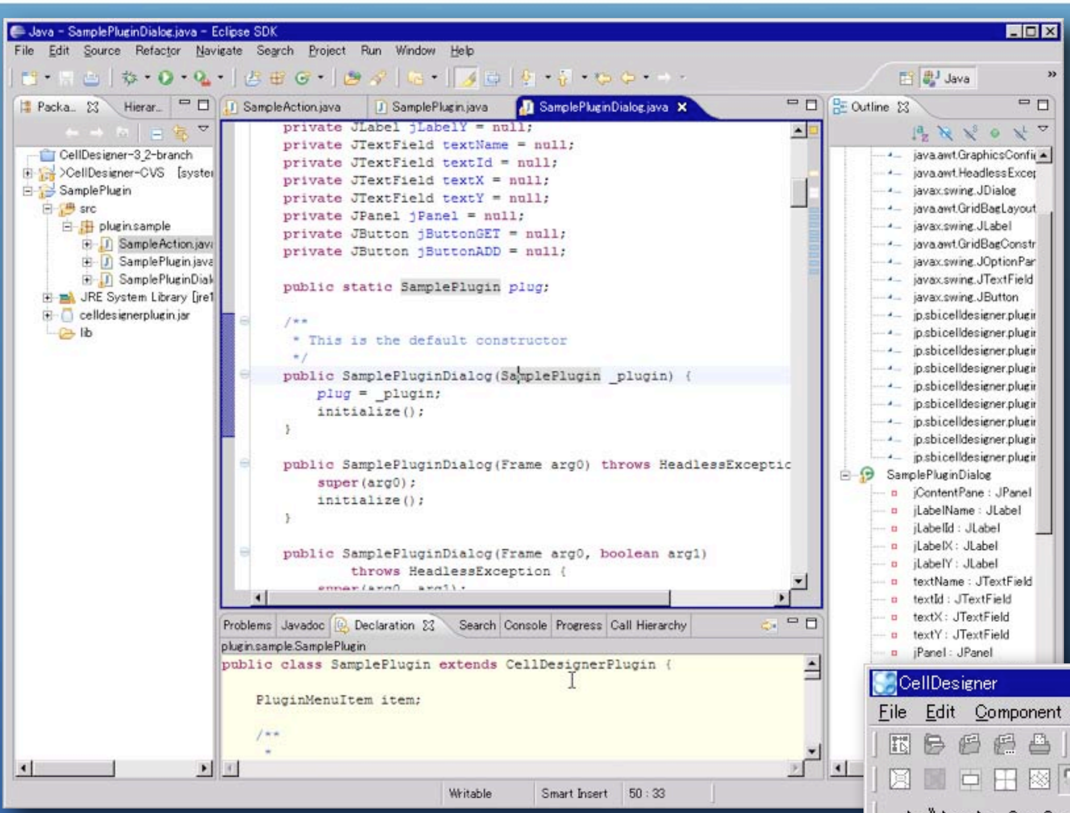
☑ Math
☐ Name

Macros


MACROS
Macro1: State Transition
s1
re1
s1
Macro2: Catalysis
s3
re2
re2
s2
s2
Macro3: Inhibition
s5
re3
re3
s4
s4
Macro4: Association
s6
s7
re4
s6
s7
s8
Macro5: Dissociation
s11
s12
s13
re5
s11
s12
Macro6: ATP_ADP
ATP
ADP
re6
re6
s16
s16
Macro7: Add Catalysis
s22
re8
re8
s1
s1
Macro9: Add Inhibition
s23
re9
re9
s1
s1
Macro10: Phosphorylation
rs1
s24
re11
P
rs1
s24
Macro11: Add Phosphate
s25
Macro12: Transcription
re13
s46
re13
s47
Macro13: Translation
re14
s49
re14
s50
Macro7: GTP_GDP
GTP
GDP
re7
re7
s19
s19

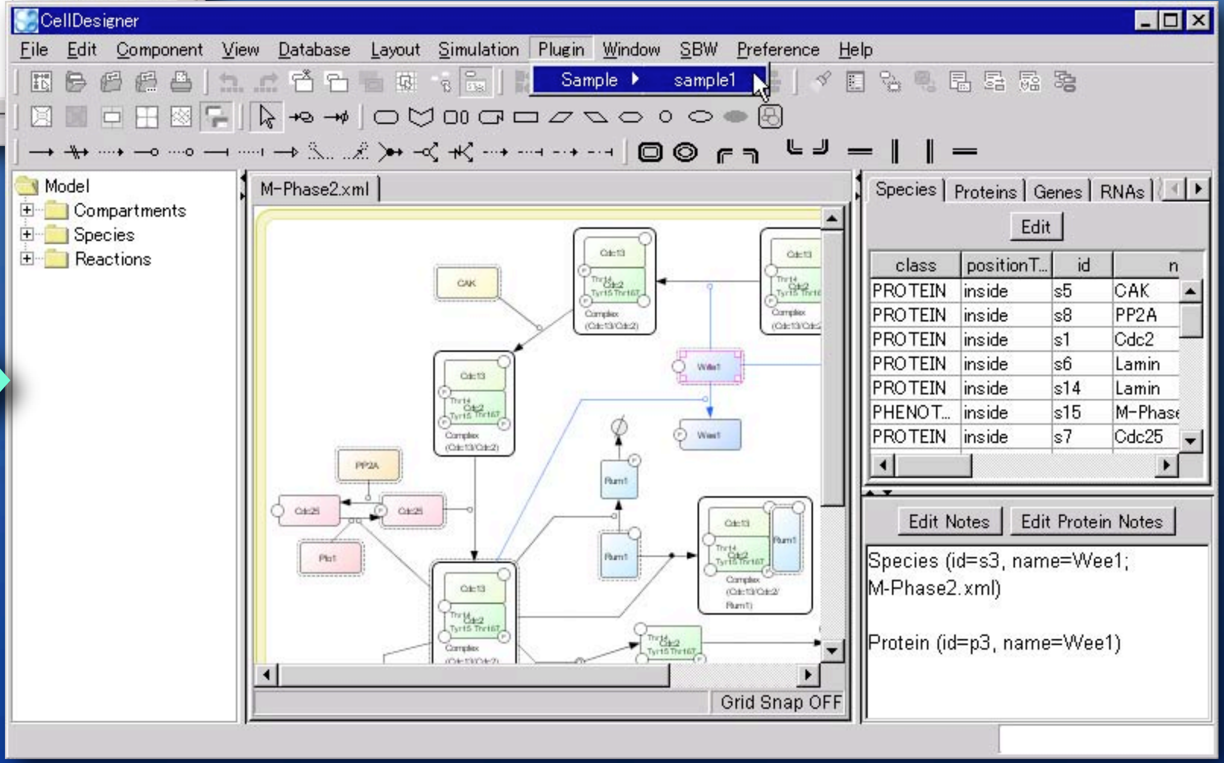
Plugin development

- Develop plugin on Eclipse
- Call plugin from [Plugin] menu on CellDesigner



```
    private JLabel jLabelY = null;
    private JTextField textName = null;
    private JTextField textId = null;
    private JTextField textX = null;
    private JTextField textY = null;
    private JPanel jPanel = null;
    private JButton jButtonGET = null;
    private JButton jButtonADD = null;

    public static SamplePlugin plug;

    /**
     * This is the default constructor
     */
    public SamplePluginDialog(SamplePlugin _plugin) {
        plug = _plugin;
        initialize();
    }

    public SamplePluginDialog(Frame arg0) throws HeadlessExceptic
        super(arg0);
        initialize();
    }

    public SamplePluginDialog(Frame arg0, boolean arg1)
            throws HeadlessException {
```

```
plugin.sample.SamplePlugin
public class SamplePlugin extends CellDesignerPlugin {

    PluginMenuItem item;

    /**
     *
```



class	positionT...	id	n
PROTEIN	inside	s5	CAK
PROTEIN	inside	s8	PP2A
PROTEIN	inside	s1	Cdc2
PROTEIN	inside	s6	Lamin
PROTEIN	inside	s14	Lamin
PHENOT...	inside	s15	M-Phase
PROTEIN	inside	s7	Cdc25

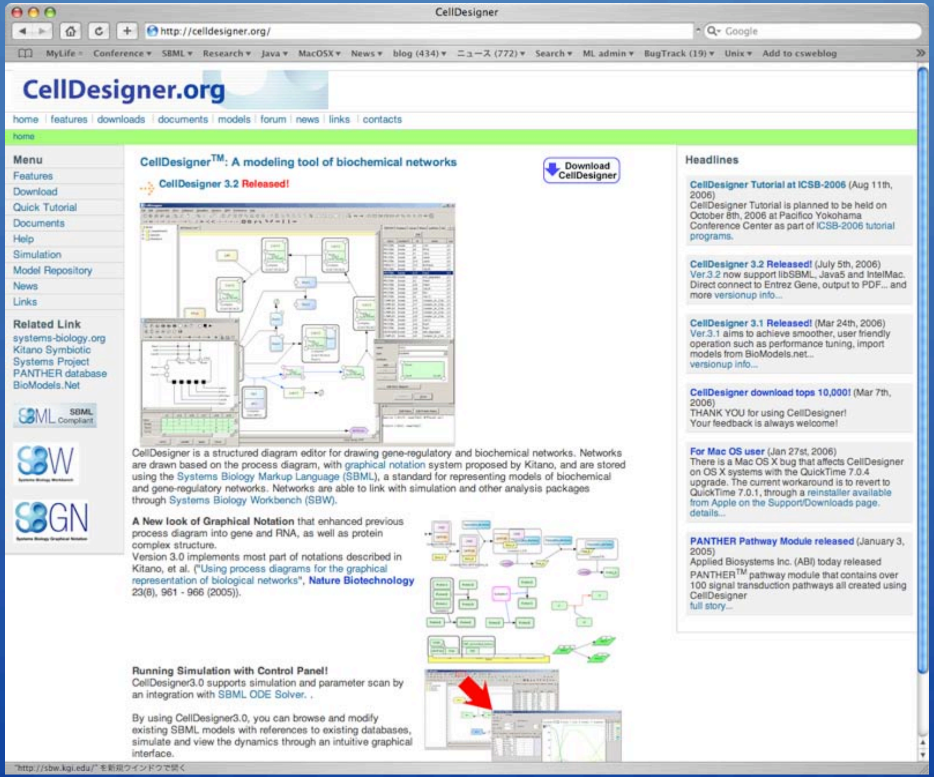
Species (id=s3, name=Wee1; M-Phase2.xml)

Protein (id=p3, name=Wee1)

Download

- Please download CellDesigner 4.0.1 from

http://celldesigner.org/


CellDesigner
http://celldesigner.org/
Google
MyLife Conference ▾ SBML ▾ Research ▾ Java ▾ MacOSX ▾ News ▾ blog (434) ▾ ニュース (772) ▾ Search ▾ ML admin ▾ BugTrack (19) ▾ Unix ▾ Add to csweblog
CellDesigner.org
home | features | downloads | documents | models | forum | news | links | contacts
home
Menu
Features
Download
Quick Tutorial
Documents
Help
Simulation
Model Repository
News
Links
Related Link
systems-biology.org
Kitano Symbiotic Systems Project
PANTHER database
BioModels.Net
SBML Compliant
SBW
SBGN
Download CellDesigner
"http://sbw.kgi.edu/" を新規ウインドウで開く

CellDesigner^{TM}: A modeling tool of biochemical networks

CellDesigner 3.2 Released!

CellDesigner is a structured diagram editor for drawing gene-regulatory and biochemical networks. Networks are drawn based on the process diagram, with graphical notation system proposed by Kitano, and are stored using the Systems Biology Markup Language (SBML), a standard for representing models of biochemical and gene-regulatory networks. Networks are able to link with simulation and other analysis packages through Systems Biology Workbench (SBW).

A New look of Graphical Notation that enhanced previous process diagram into gene and RNA, as well as protein complex structure.
Version 3.0 implements most part of notations described in Kitano, et al. ("Using process diagrams for the graphical representation of biological networks", **Nature Biotechnology** 23(8), 961 - 966 (2005)).

Running Simulation with Control Panel!
CellDesigner3.0 supports simulation and parameter scan by an integration with SBML ODE Solver. .

By using CellDesigner3.0, you can browse and modify existing SBML models with references to existing databases, simulate and view the dynamics through an intuitive graphical interface.

Headlines

CellDesigner Tutorial at ICSB-2006 (Aug 11th, 2006)
CellDesigner Tutorial is planned to be held on October 8th, 2006 at Pacifico Yokohama Conference Center as part of ICSB-2006 tutorial programs.

CellDesigner 3.2 Released! (July 5th, 2006)
Ver.3.2 now support libSBML, Java5 and IntelMac. Direct connect to Entrez Gene, output to PDF... and more versionup info...

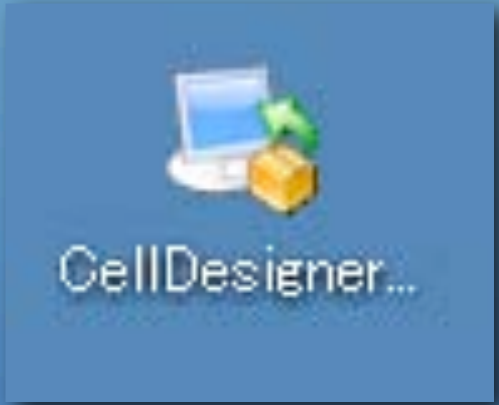
CellDesigner 3.1 Released! (Mar 24th, 2006)
Ver.3.1 aims to achieve smoother, user friendly operation such as performance tuning, import models from BioModels.net...
versionup info...

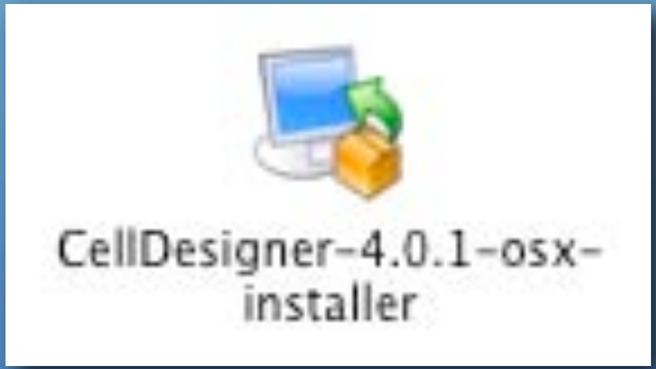
CellDesigner download tops 10,000! (Mar 7th, 2006)
THANK YOU for using CellDesigner!
Your feedback is always welcome!

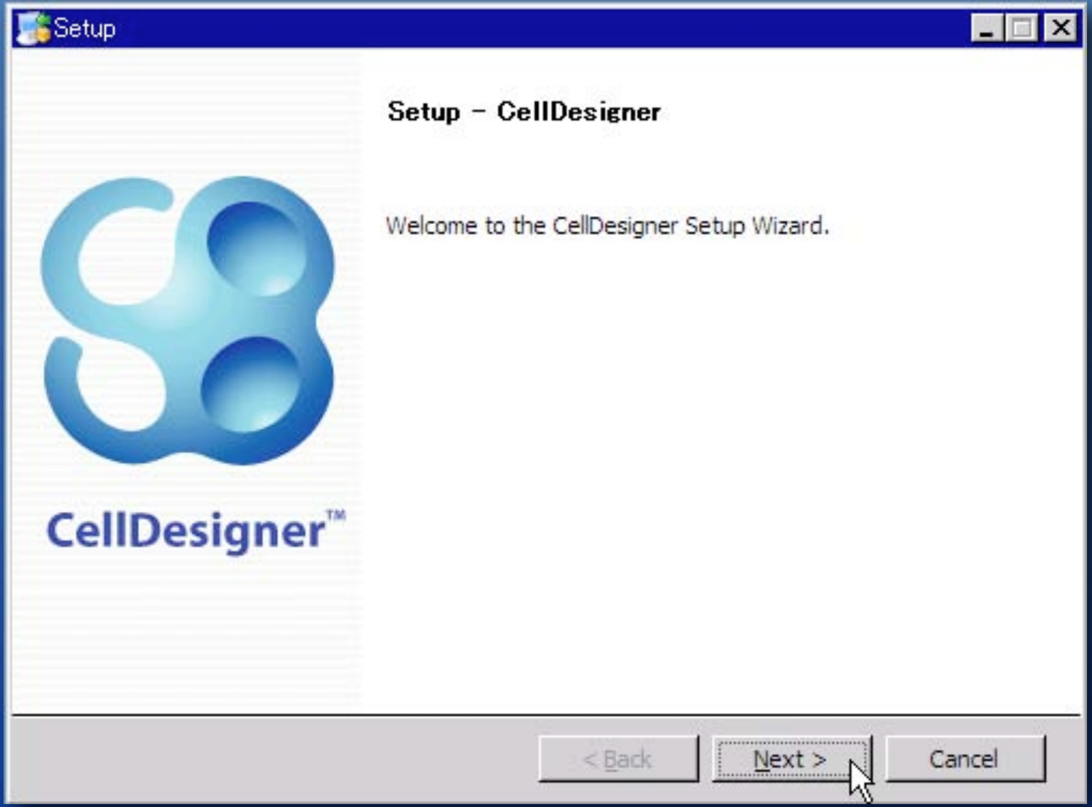
For Mac OS user (Jan 27st, 2006)
There is a Mac OS X bug that affects CellDesigner on OS X systems with the QuickTime 7.0.4 upgrade. The current workaround is to revert to QuickTime 7.0.1, through a reinstaller available from Apple on the Support/Downloads page.
details...

PANTHER Pathway Module released (January 3, 2005)
Applied Biosystems Inc. (ABI) today released PANTHER^{TM} pathway module that contains over 100 signal transduction pathways all created using CellDesigner
full story...

Installation





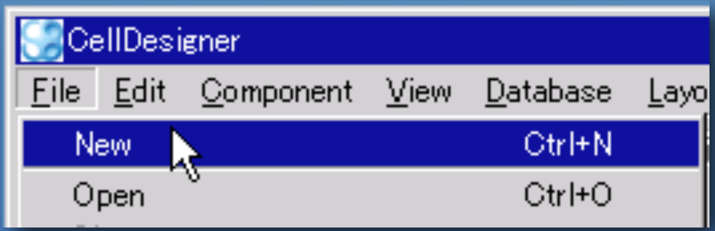


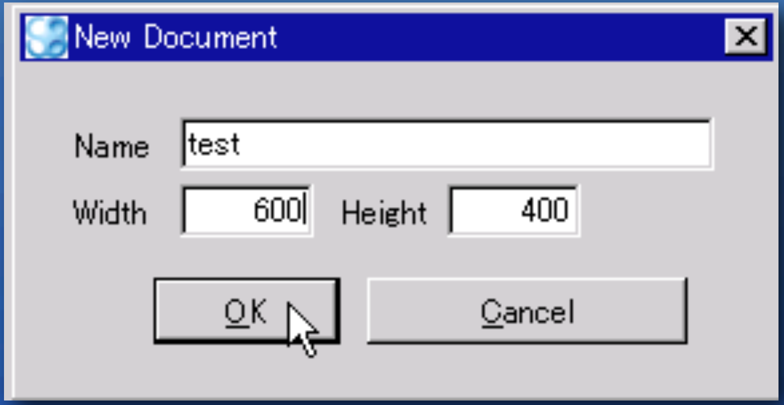
Setup - CellDesigner

Welcome to the CellDesigner Setup Wizard.

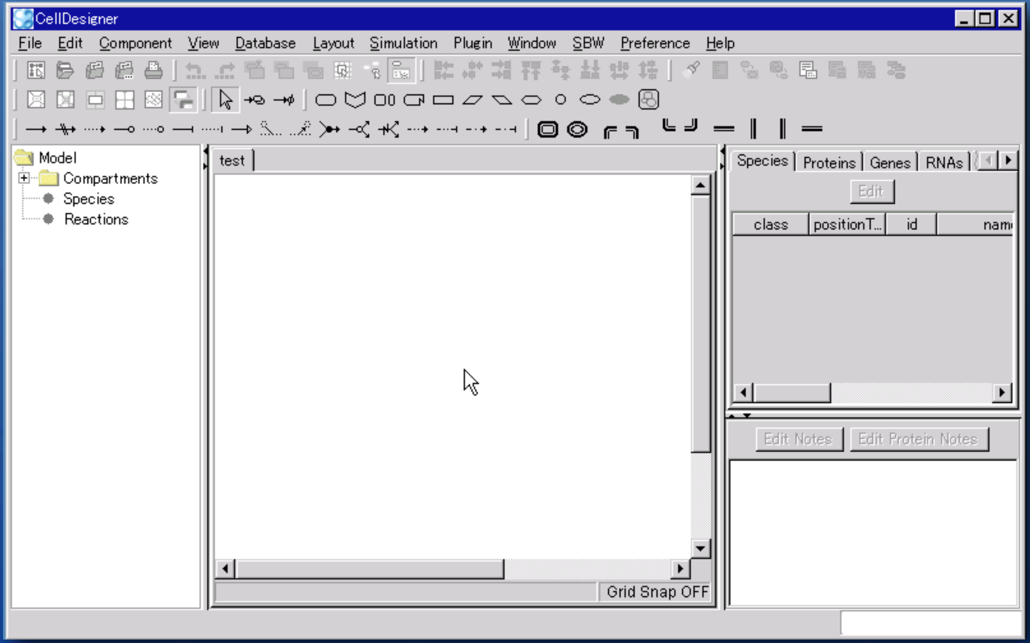
Demonstration

- Create new model:
  - [File] → [New] → input title → [OK]



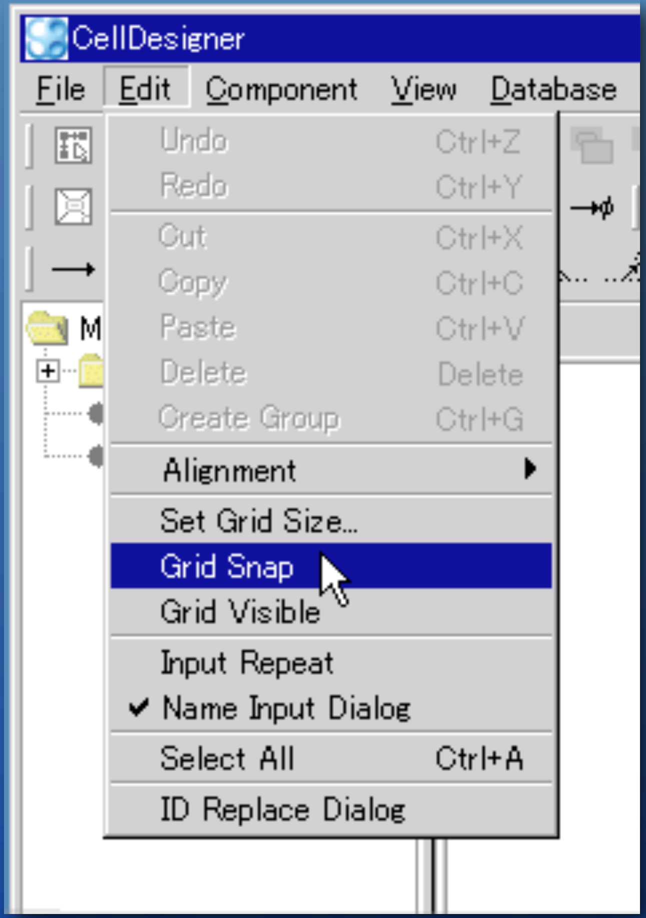


Name: test
Width: 600
Height: 400


CellDesigner
File Edit Component View Database Layout Simulation Plugin Window SBW Preference Help
Model
Compartments
Species
Reactions
test
Species Proteins Genes RNAs
Edit
class positionT... id nam
Edit Notes Edit Protein Notes
Grid Snap OFF

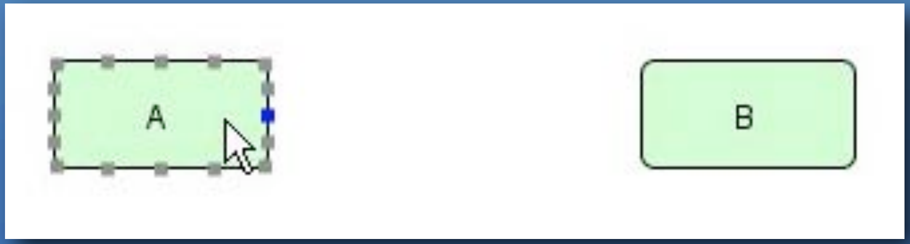
Tips

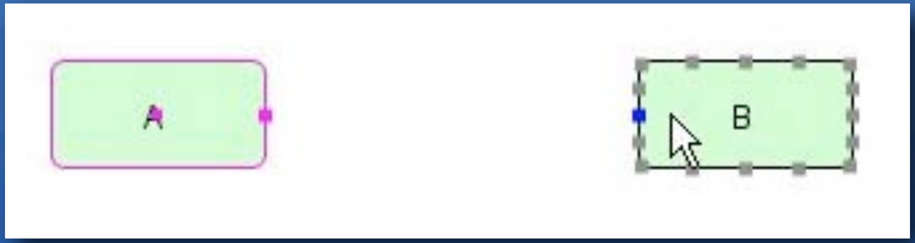
- Enable [Grid Snap] will help you draw your model much easier

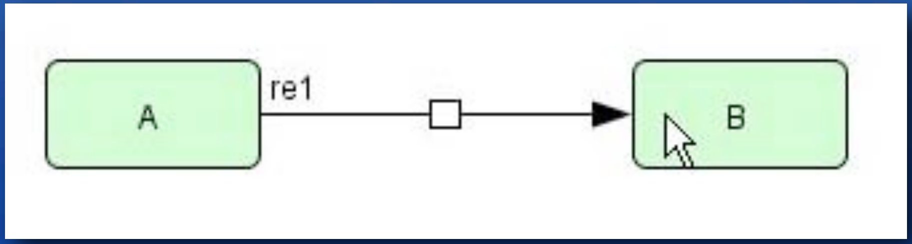


Create Reaction

- Create Protein “A” and “B”
- Draw “State transition” arrow from “A” to “B”

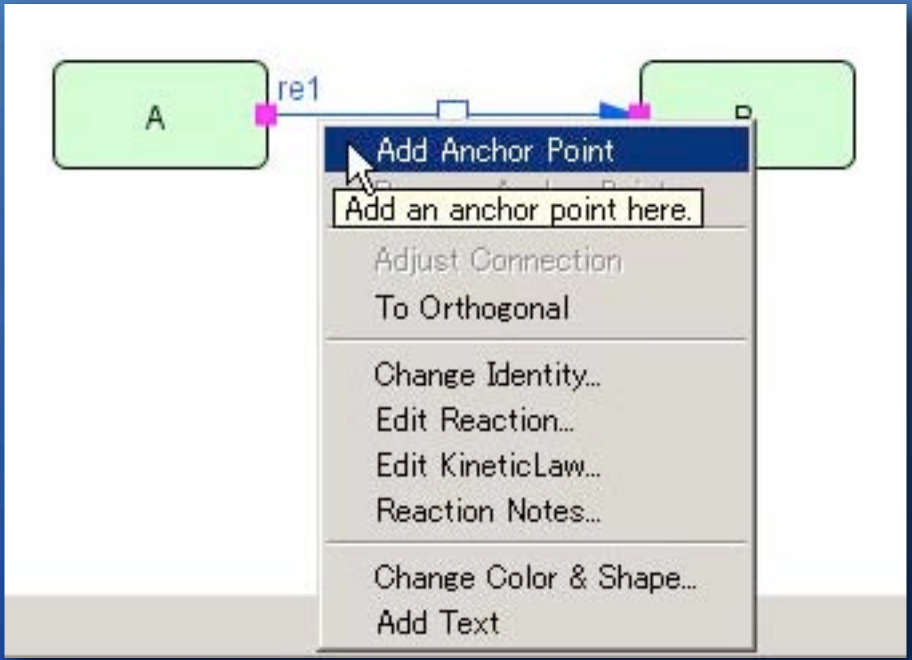


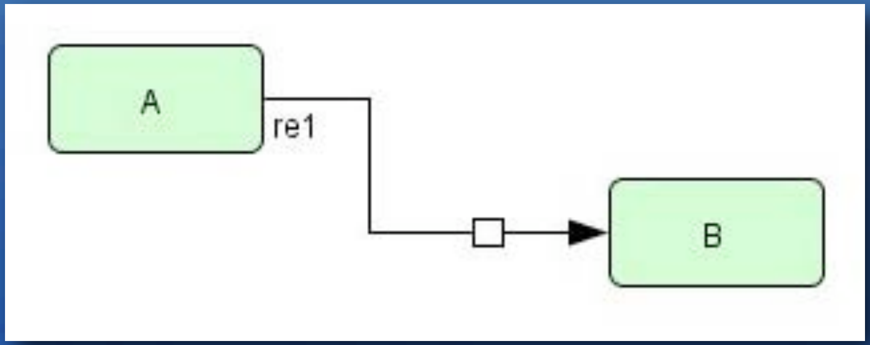




Add Anchor Point

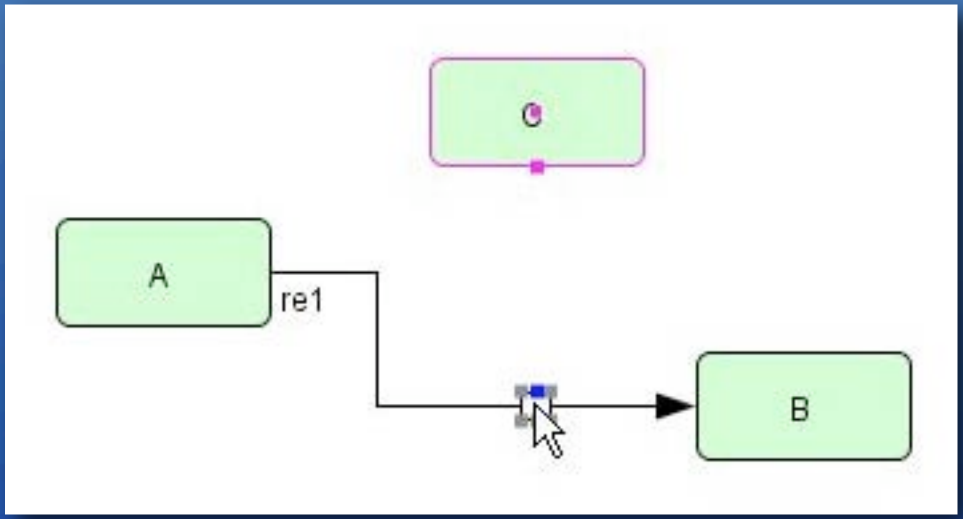
- Add 2 anchor points to reaction
- Drag reaction and anchor point to change its shape

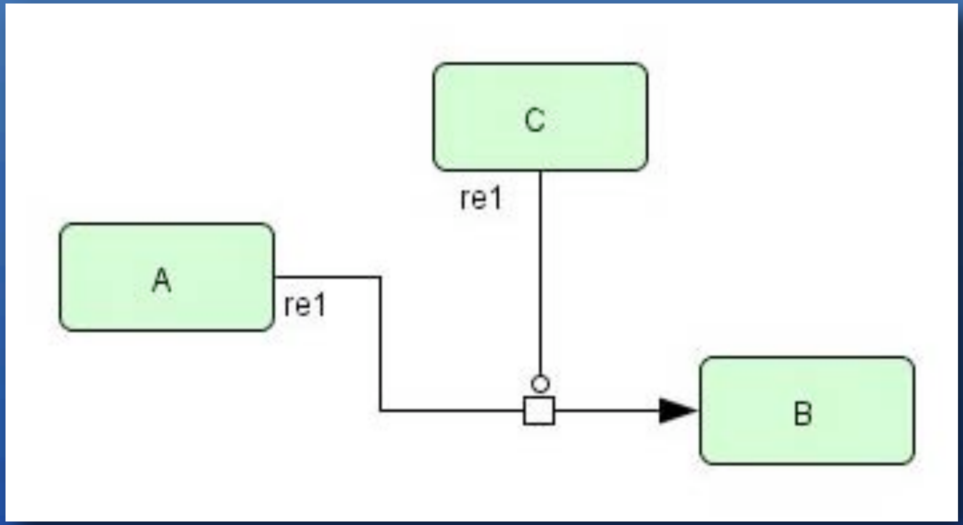




Add Catalysis reaction

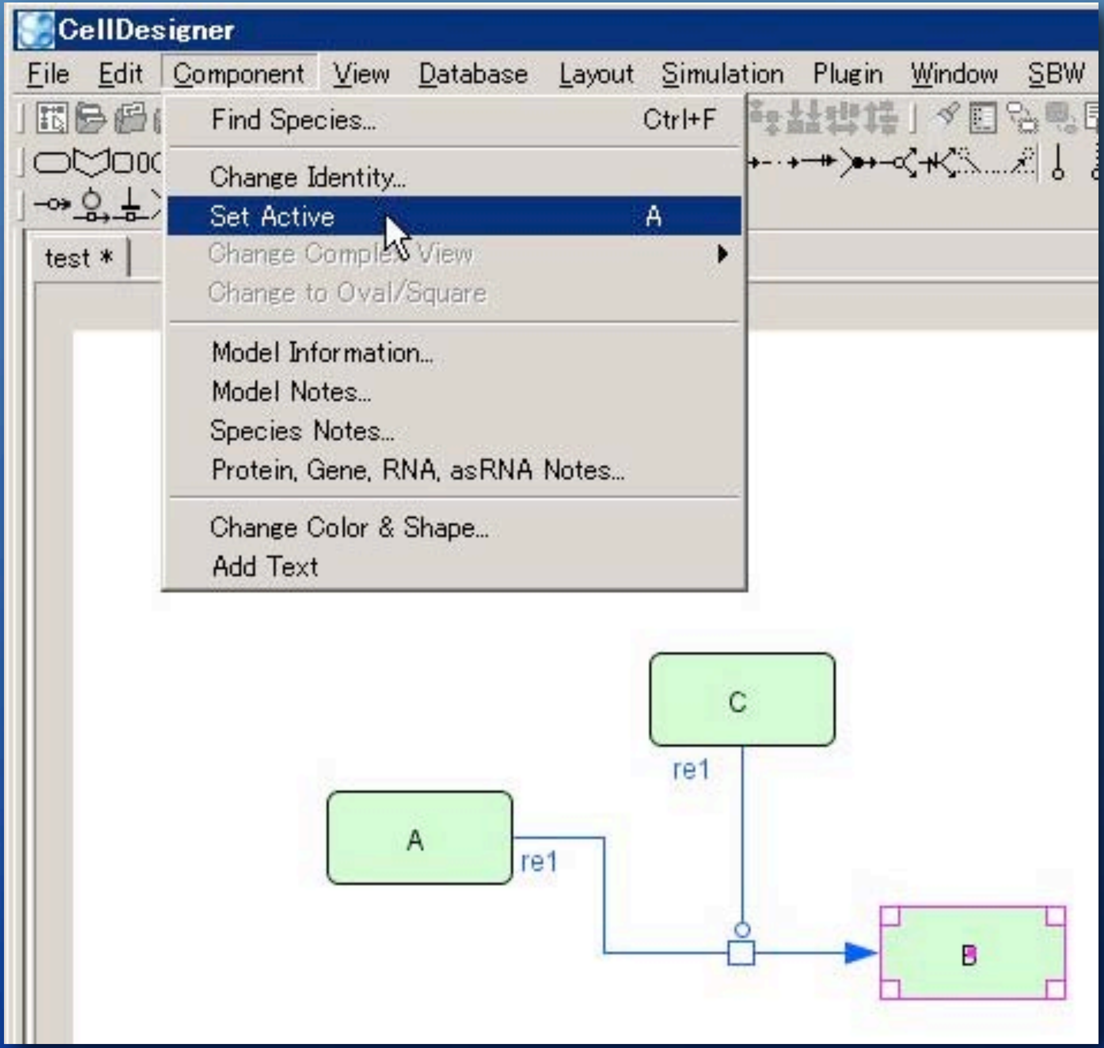
- Add Protein “C”
- Add Catalysis reaction from “C” to the reaction

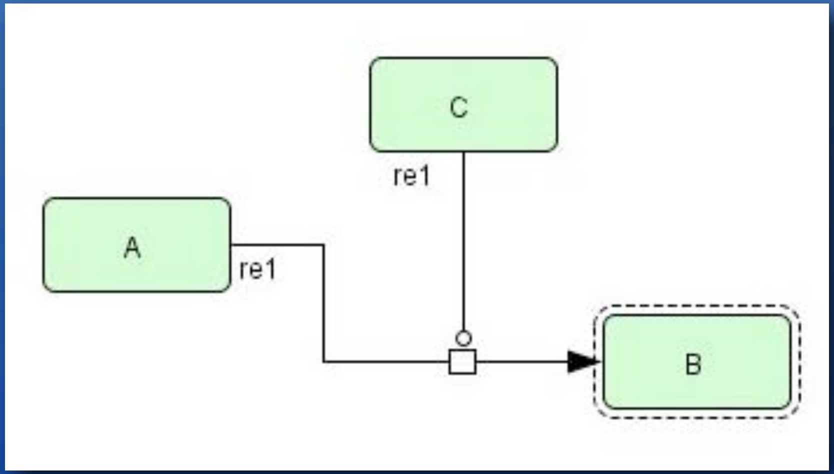




Set Active state

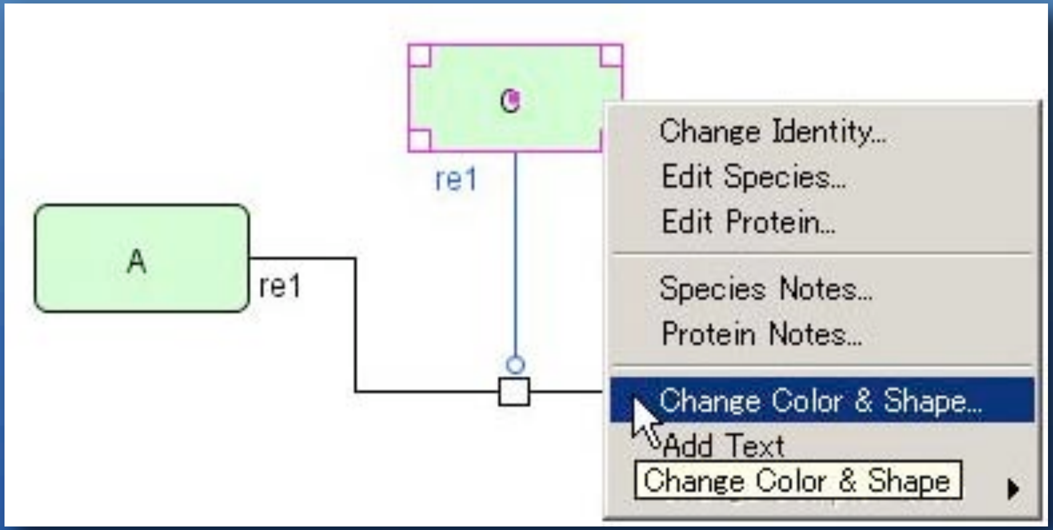
- Select Protein “B”
- [Component] → [Set Active]

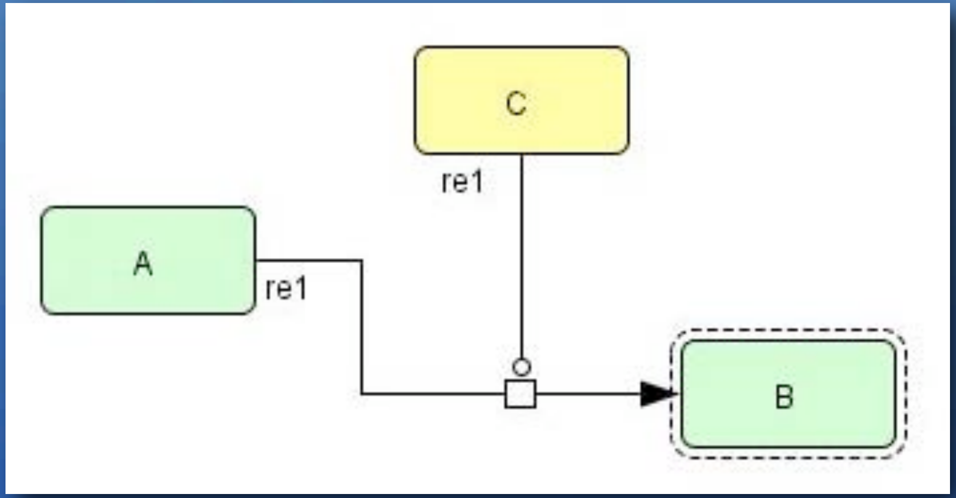




Change Color

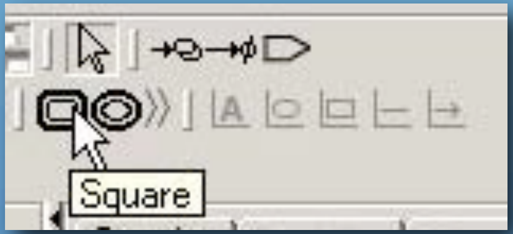
- Right click on Protein “C”
- Select [Change Color & Shape...]



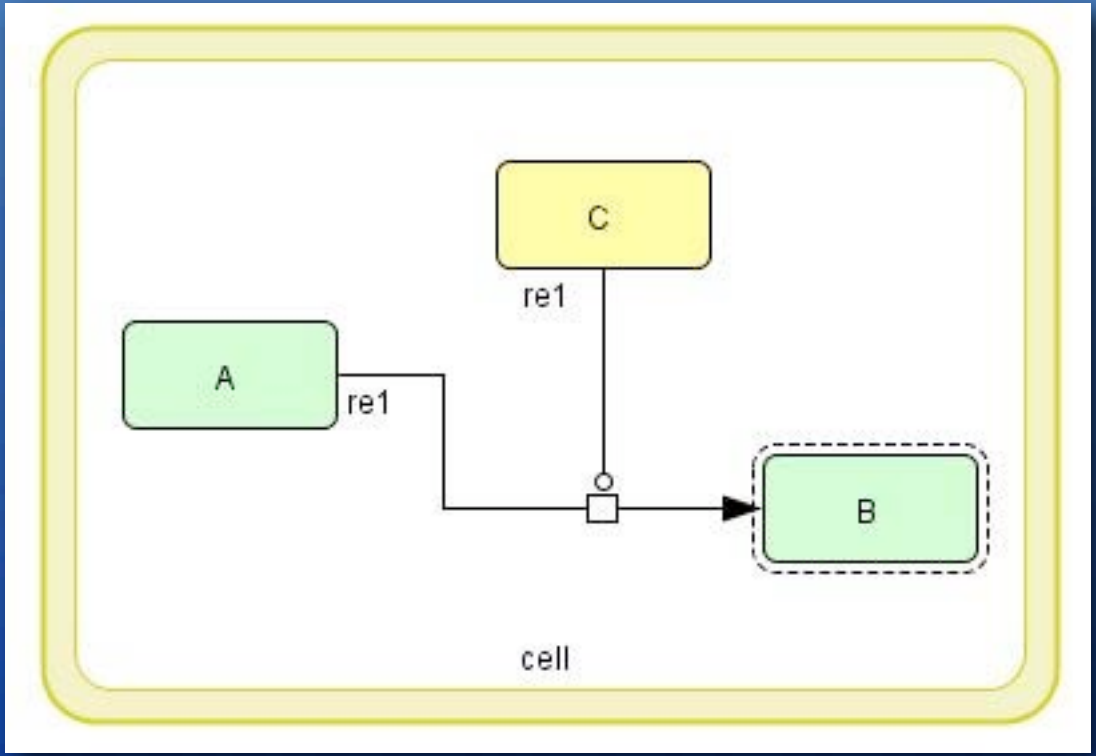


Compartment

- Click [Compartment] icon

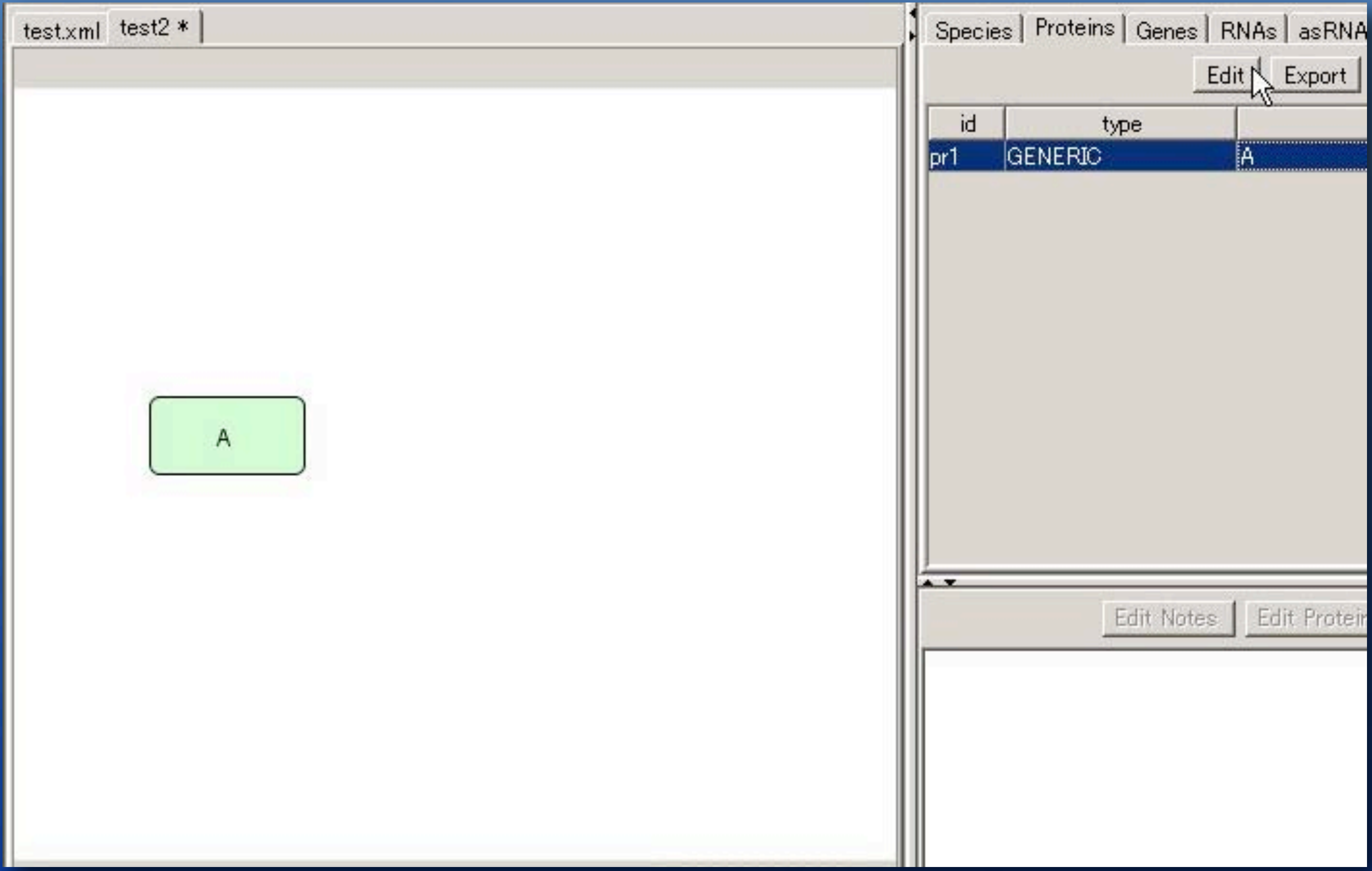


- Drag mouse cursor to specify its area
- Input name of compartment



Add Residue to Protein

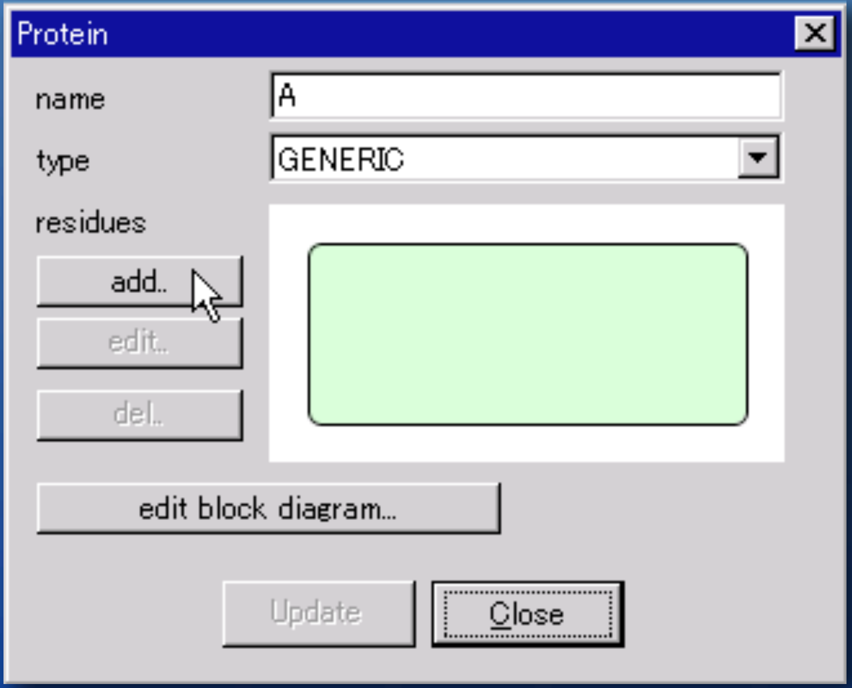
- Create new model (test2)
- Create Protein "A"
- Select Protein "A" in [Proteins] Tab
- Click [Edit] button



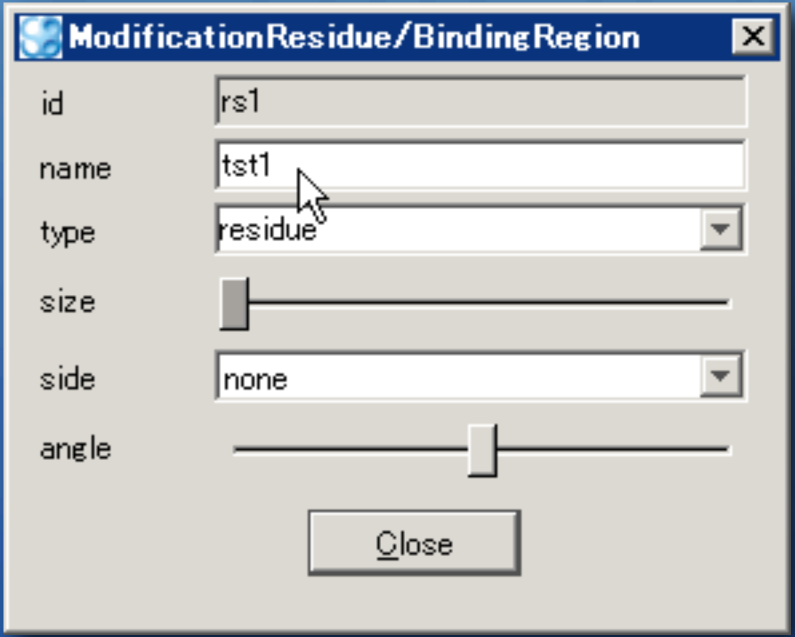
id	type	
pr1	GENERIC	A

Add Residue to Protein

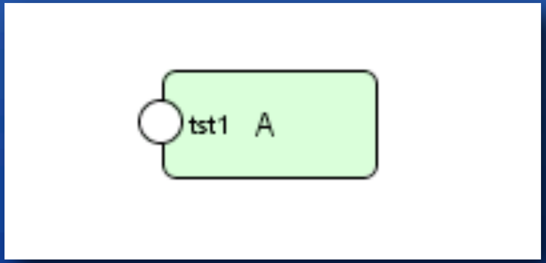
- Click [add] button on [Protein] dialog
- Input name for the residue (tst1)
- Click [Close] button
- Click [Update] Button



name: A
type: GENERIC
residues

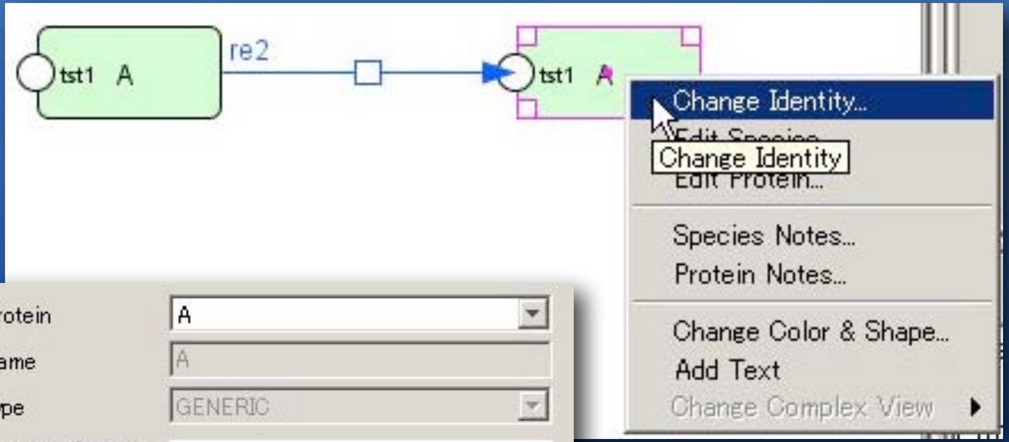


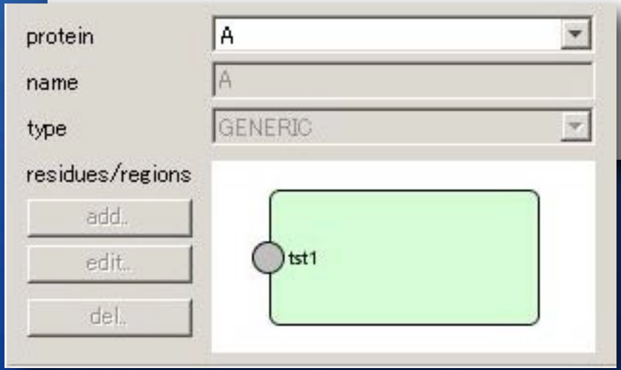
id: rs1
name: tst1
type: residue
side: none



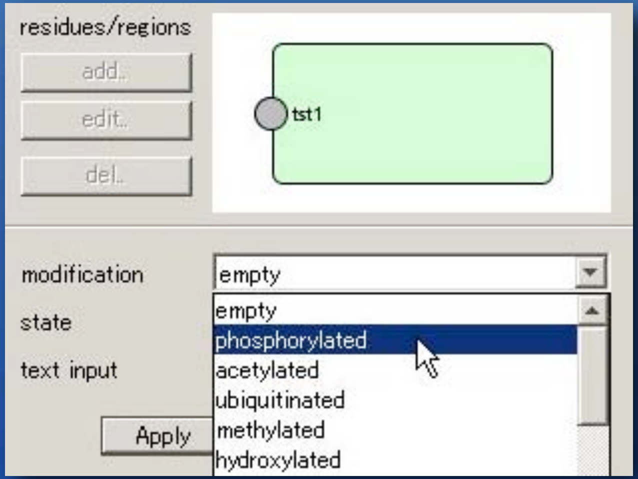
Add Residue to Protein

- Copy & Paste Protein “A” and then draw “State Transition” arrow
- Right Click on “A” (right side) and select [Change Identity...]
- Click residue “tst1” in Dialog
- Select [phosphorylated] in modification

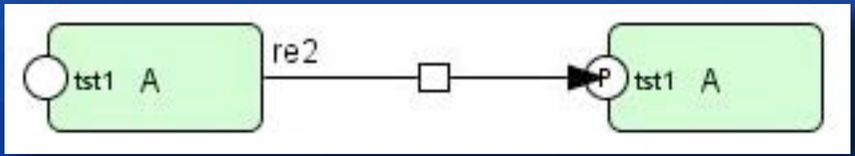




protein	A
name	A
type	GENERIC
residues/regions	

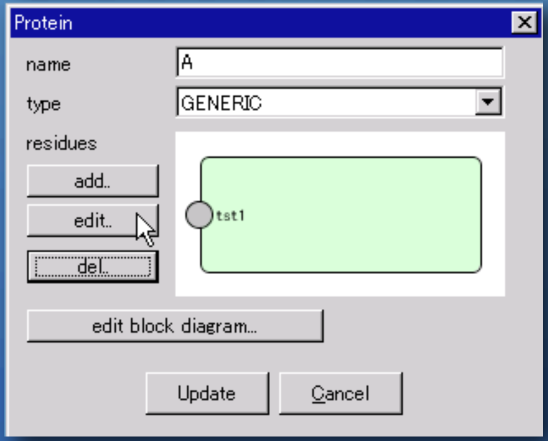


modification: empty

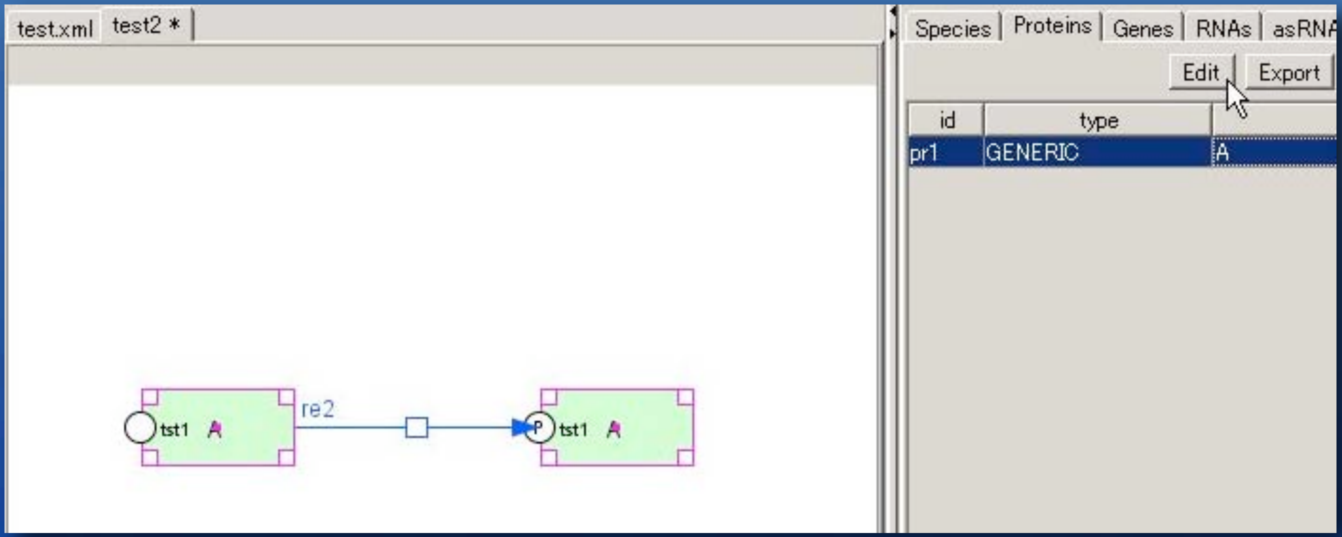


Change position of Residue

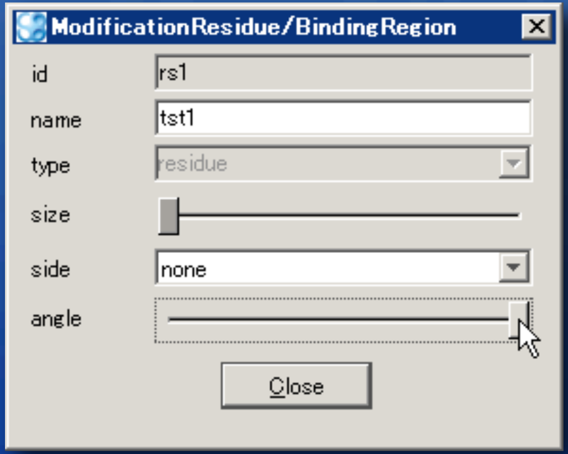
- Select Protein "A" in [Proteins] Tab
- Click [Edit] button
- Click residue "tst1" in Dialog
- Click [edit] button
- Drag [angle] slidebar


Protein

name	A
type	GENERIC
residues	tst1



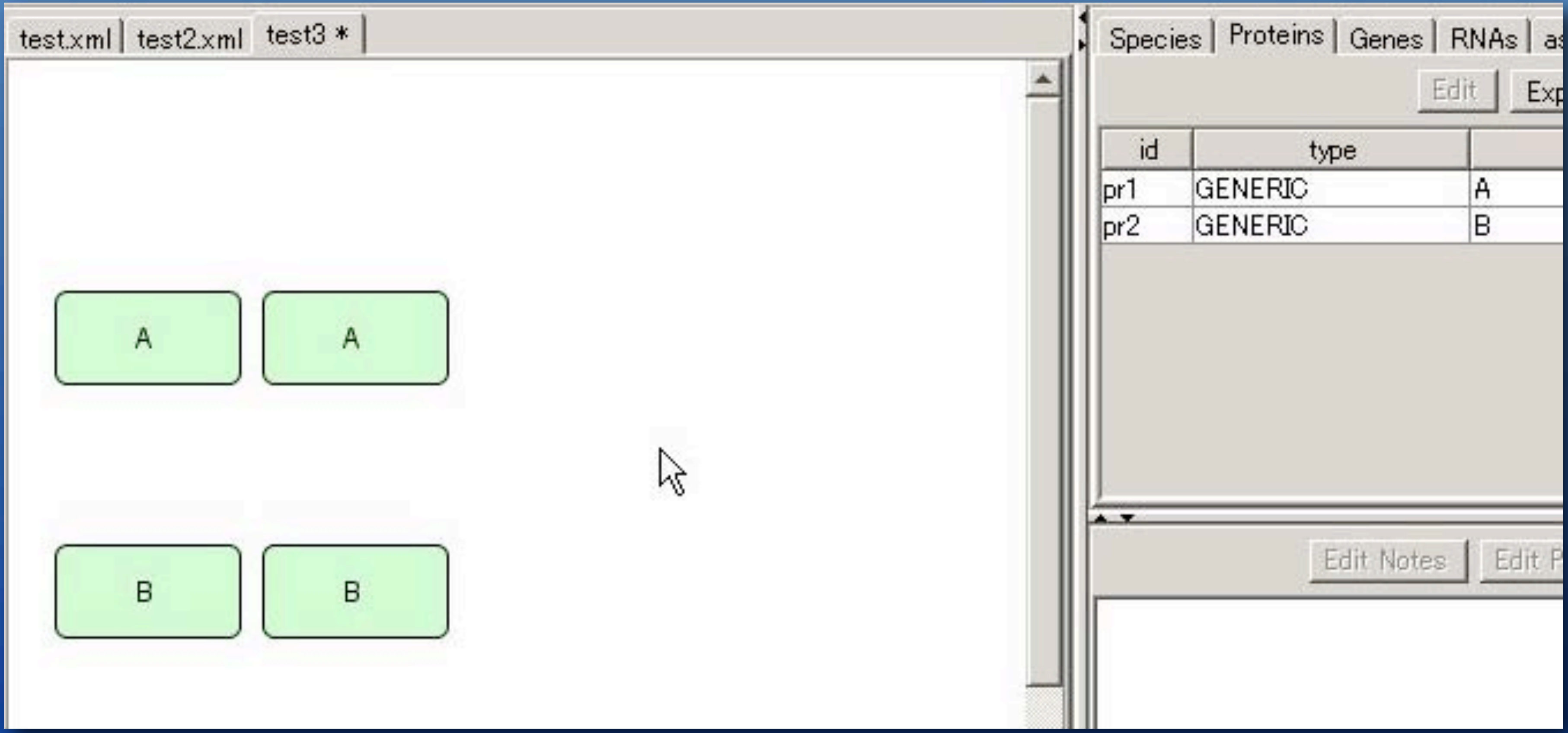
id	type	
pr1	GENERIC	A


ModificationResidue/BindingRegion

id	rs1
name	tst1
type	residue
size	
side	none
angle	

Complex

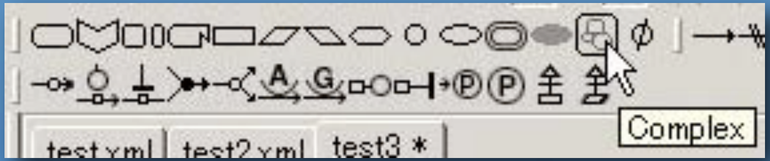
- Create new model (test3)
- Create Proteins “A” and “B”
- Copy & Paste both “A” and “B”



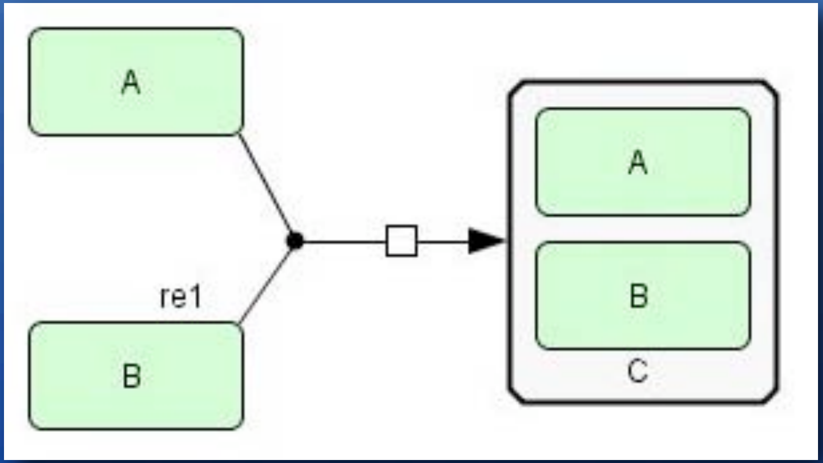
id	type	
pr1	GENERIC	A
pr2	GENERIC	B

Complex

- Click [Complex] icon and create complex "C"

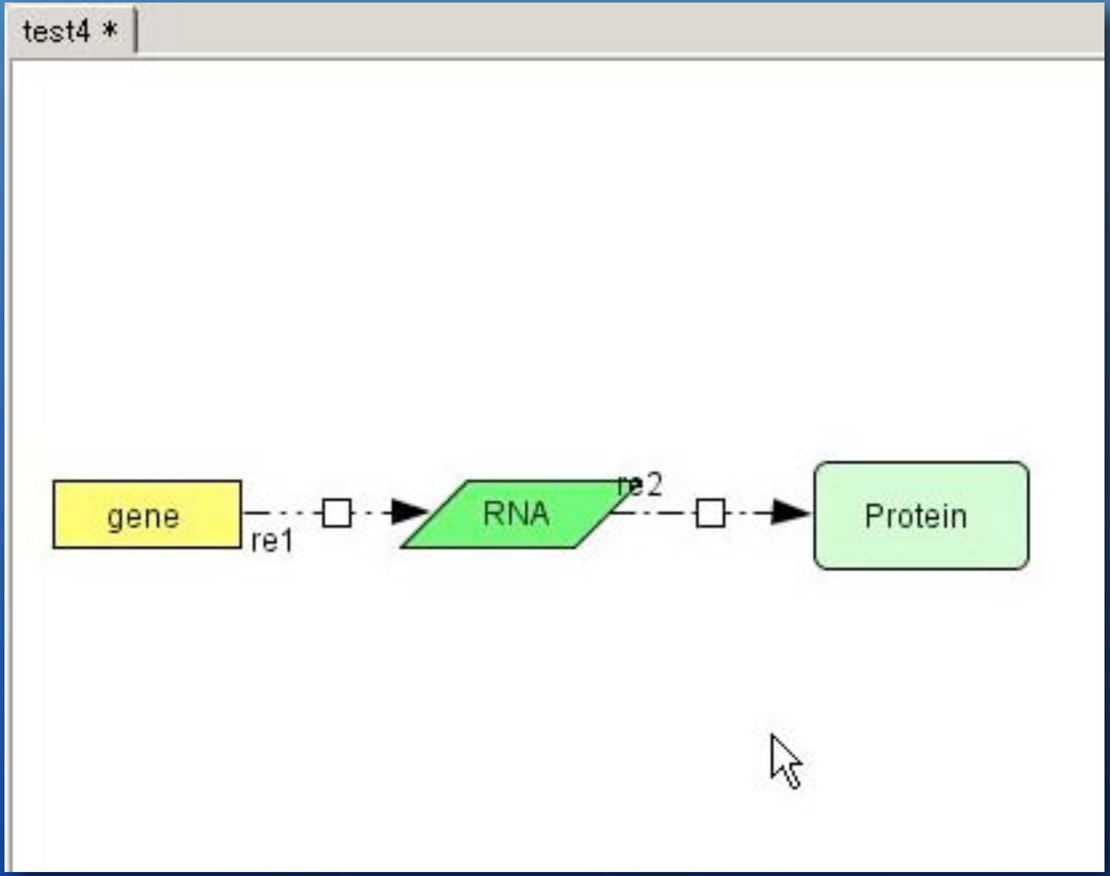


- Drag Protein "A" and "B" into complex C
- Draw "Heterodimer Association" arrow



Gene & RNA

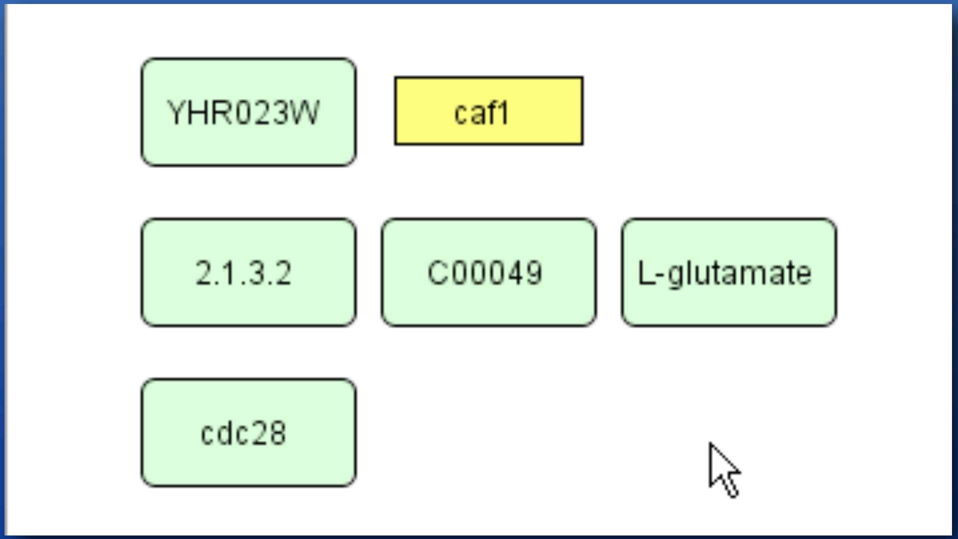
- Create new model (test4)
- Create gene, RNA and Protein
- Draw “Transcription” and “Translation”



See “geneRNA40.xml” for more examples

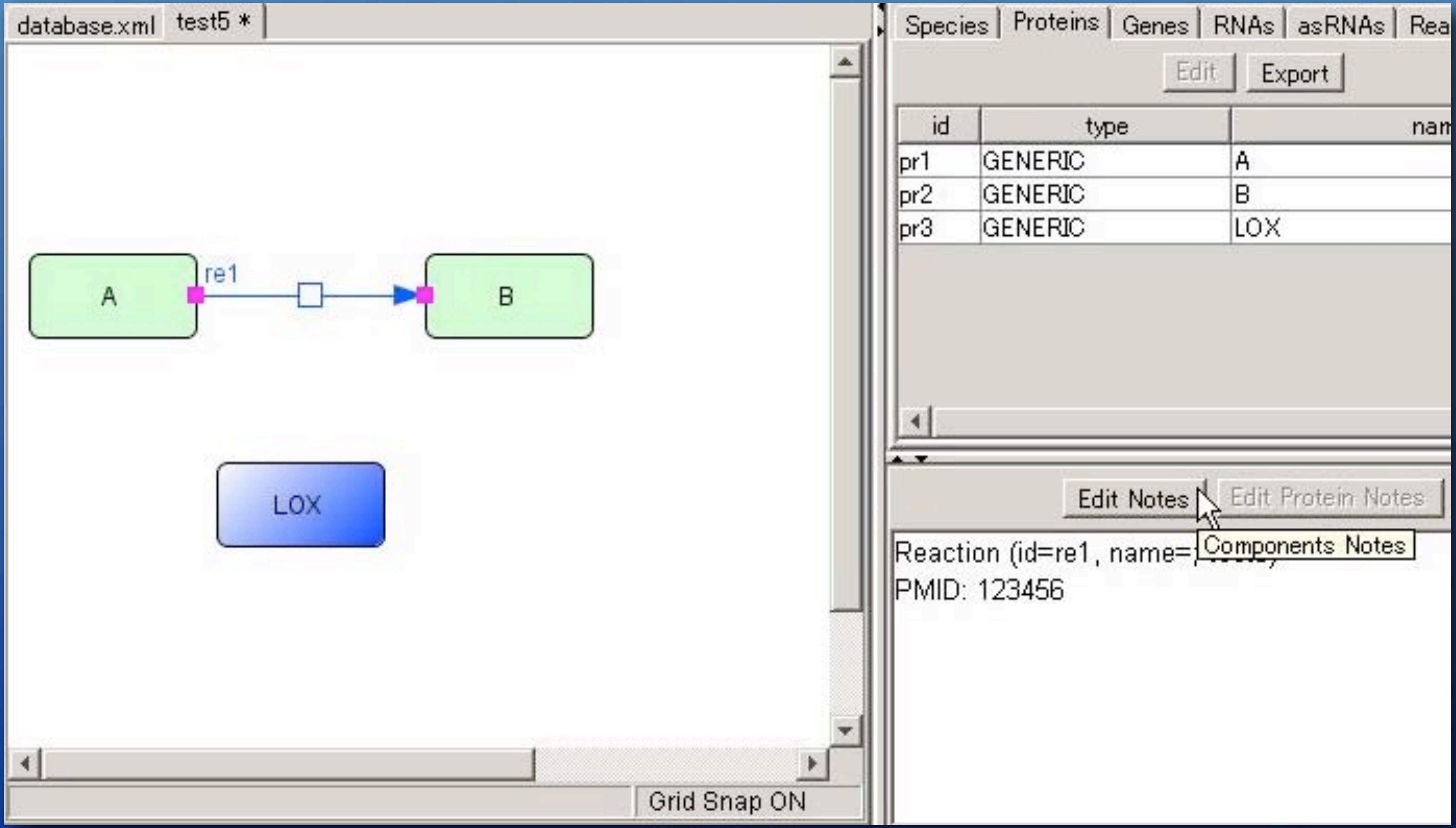
Database connection

- Search Database by Name:
  - SGD
  - DBGET
  - iHOP
  - Entrez Gene
  - Genome Network Platform



Database connection

- Search Database by Notes:
  - PubMed: PMID: 123456
  - Entrez Gene: GeneID: 4015

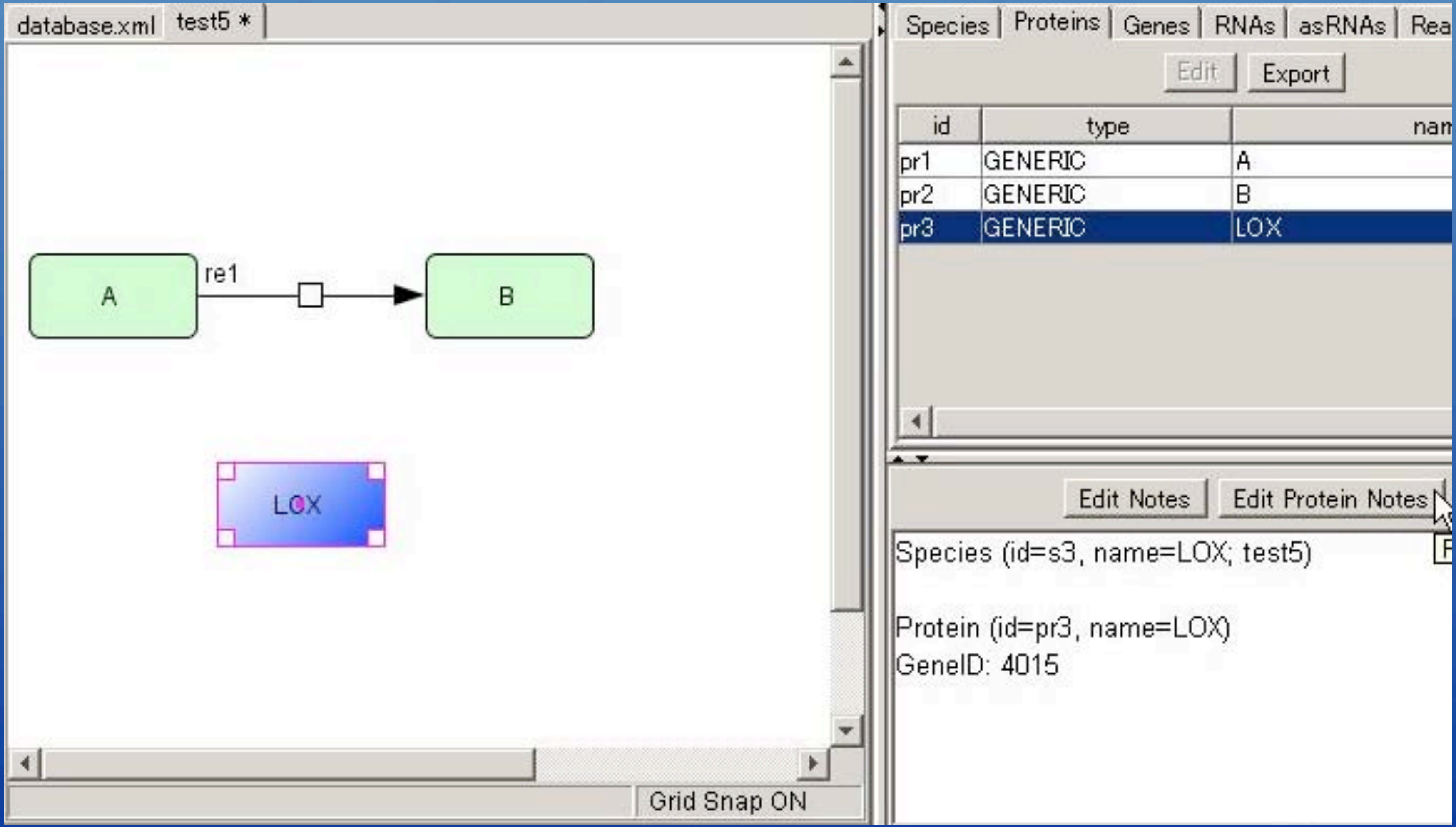


id	type	nam
pr1	GENERIC	A
pr2	GENERIC	B
pr3	GENERIC	LOX

Reaction (id=re1, name=
PMID: 123456

Database connection

- Search Database by Notes:
  - PubMed: PMID: 123456
  - Entrez Gene: GeneID: 4015



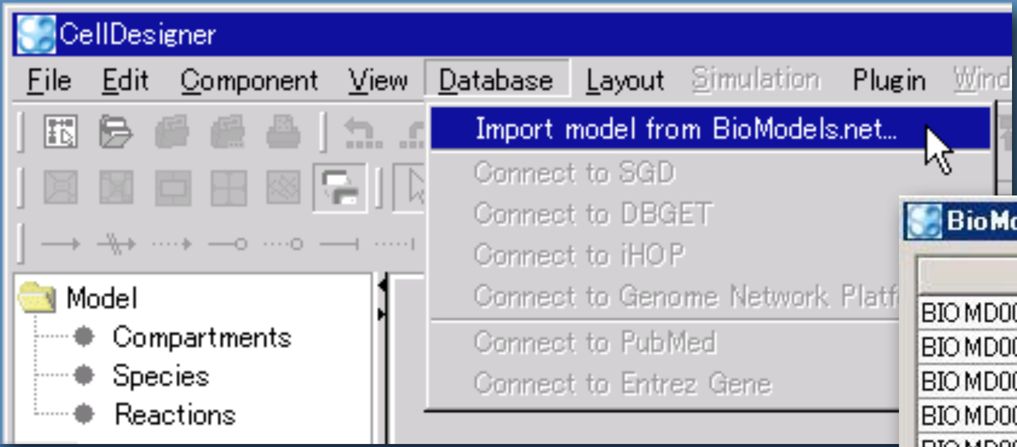
id	type	nam
pr1	GENERIC	A
pr2	GENERIC	B
pr3	GENERIC	LOX

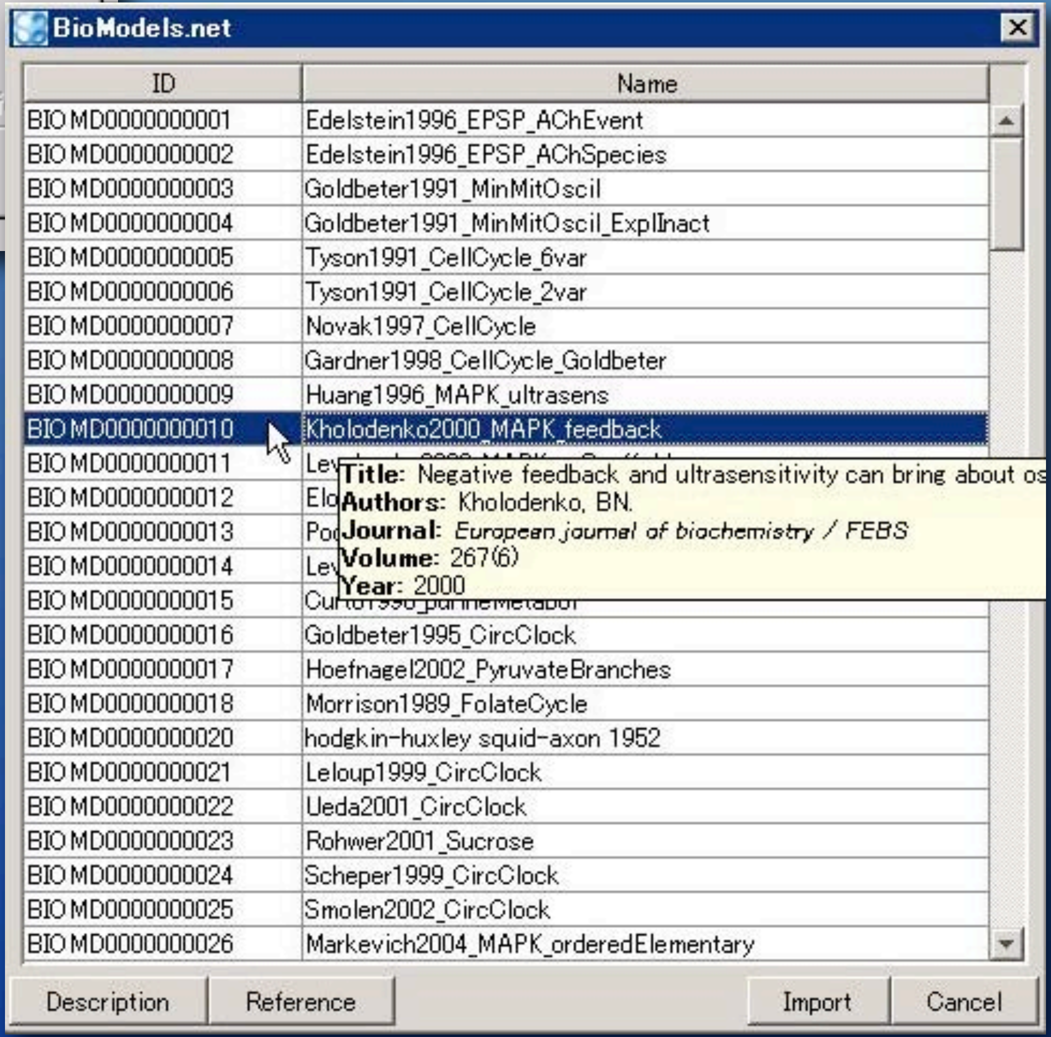
Species (id=s3, name=LOX; test5)

Protein (id=pr3, name=LOX)
GeneID: 4015

Database connection

Import model from BioModels.net





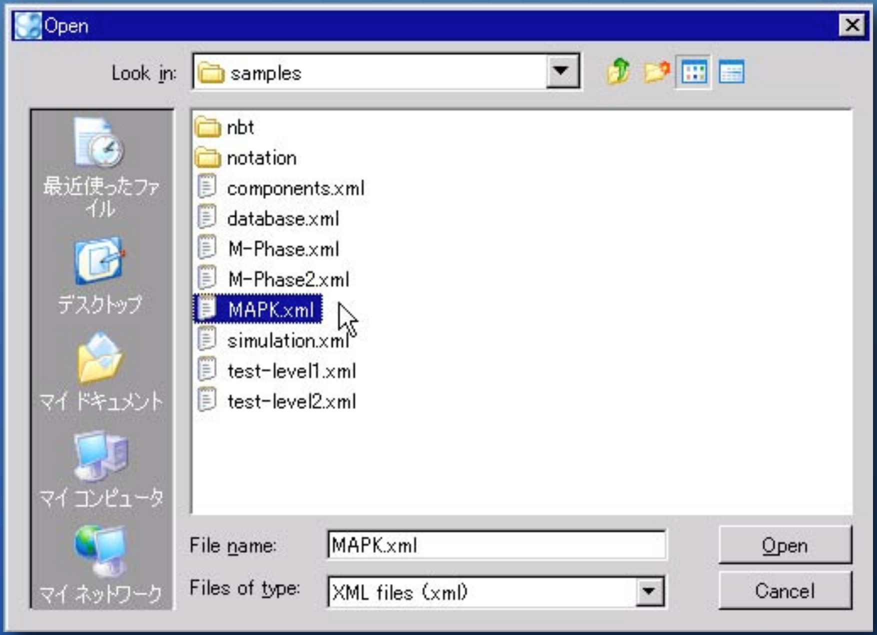
ID	Name
BIOMD0000000001	Edelstein1996_EPSP_AChEvent
BIOMD0000000002	Edelstein1996_EPSP_AChSpecies
BIOMD0000000003	Goldbeter1991_MinMitOscil
BIOMD0000000004	Goldbeter1991_MinMitOscil_ExplInact
BIOMD0000000005	Tyson1991_CellCycle_6var
BIOMD0000000006	Tyson1991_CellCycle_2var
BIOMD0000000007	Novak1997_CellCycle
BIOMD0000000008	Gardner1998_CellCycle_Goldbeter
BIOMD0000000009	Huang1996_MAPK_ultrasens
BIOMD0000000010	Kholodenko2000_MAPK_feedback
BIOMD0000000011	Lev
BIOMD0000000012	Elo
BIOMD0000000013	Po
BIOMD0000000014	Lev
BIOMD0000000015	Cu
BIOMD0000000016	Goldbeter1995_CircClock
BIOMD0000000017	Hoefnagel2002_PyruvateBranches
BIOMD0000000018	Morrison1989_FolateCycle
BIOMD0000000020	hodgkin-huxley squid-axon 1952
BIOMD0000000021	Leloup1999_CircClock
BIOMD0000000022	Ueda2001_CircClock
BIOMD0000000023	Rohwer2001_Sucrose
BIOMD0000000024	Scheper1999_CircClock
BIOMD0000000025	Smolen2002_CircClock
BIOMD0000000026	Markevich2004_MAPK_orderedElementary

Title: Negative feedback and ultrasensitivity can bring about os
Authors: Kholodenko, BN.
Journal: *European journal of biochemistry / FEBS*
Volume: 267(6)
Year: 2000

SBI The Systems Biology Institute

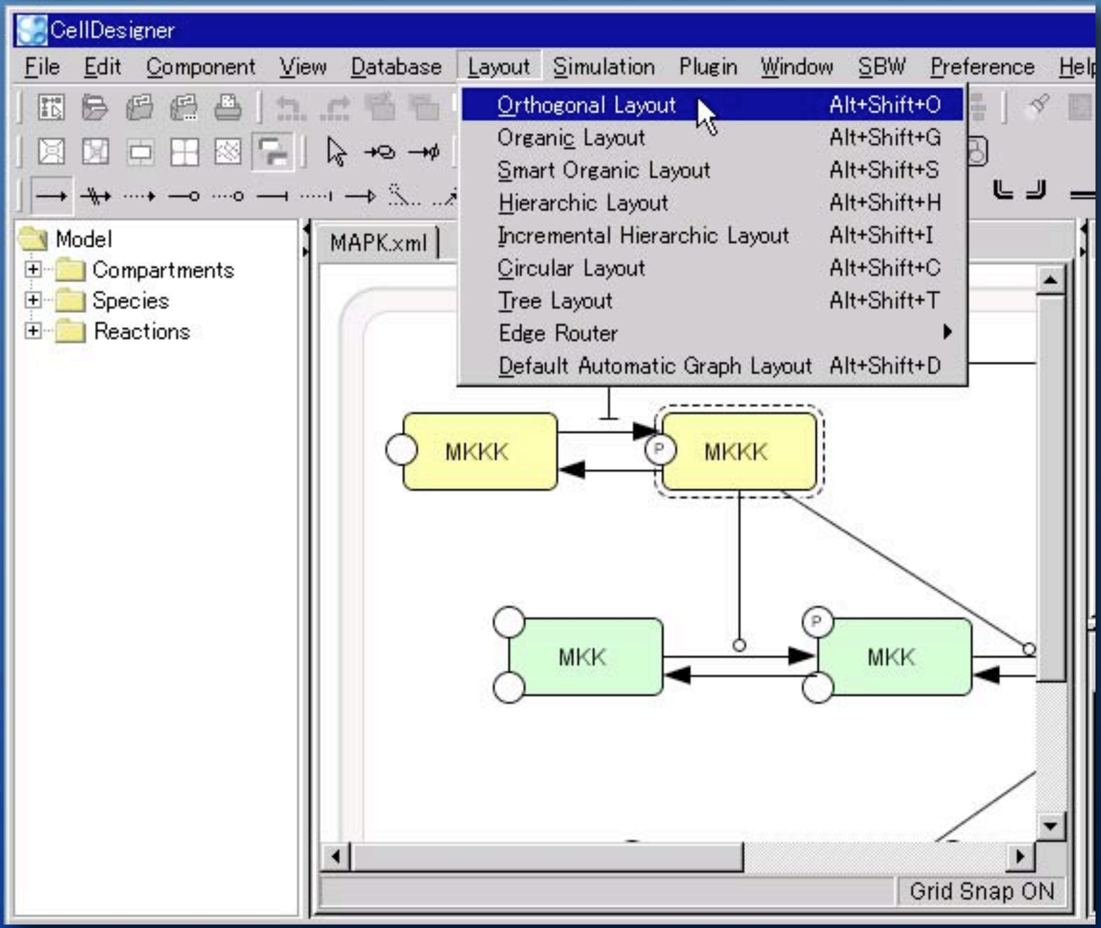
Auto layout

- [File] → [Open] → samples/MAPK.xml
- [Layout] → [Orthogonal Layout]

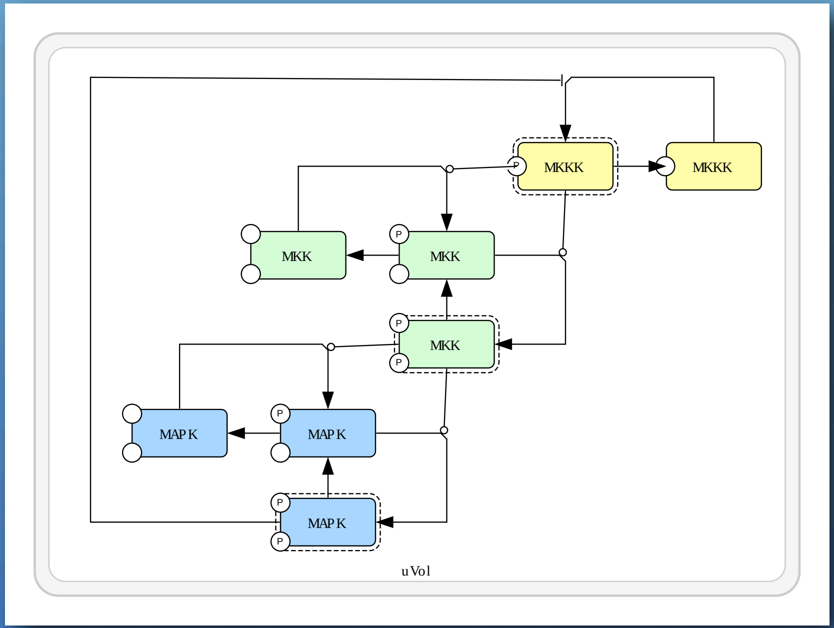


File name: MAPK.xml

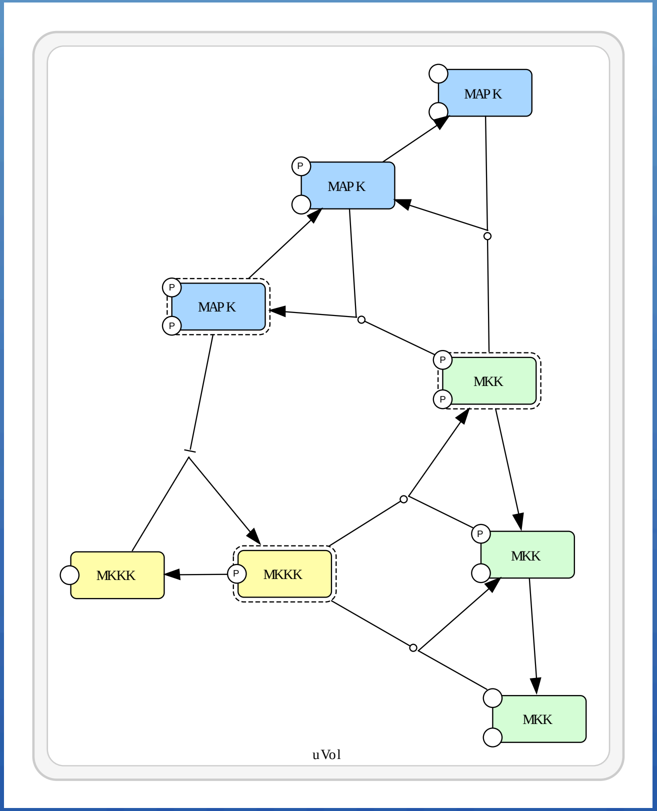
Files of type: XML files (xml)


CellDesigner
File Edit Component View Database Layout Simulation Plugin Window SBW Preference Help
Orthogonal Layout Alt+Shift+O
Organic Layout Alt+Shift+G
Smart Organic Layout Alt+Shift+S
Hierarchic Layout Alt+Shift+H
Incremental Hierarchic Layout Alt+Shift+I
Circular Layout Alt+Shift+C
Tree Layout Alt+Shift+T
Edge Router
Default Automatic Graph Layout Alt+Shift+D
Model
Compartments
Species
Reactions
MAPK.xml
MKKK
P
MKKK
MKK
P
MKK
Grid Snap ON

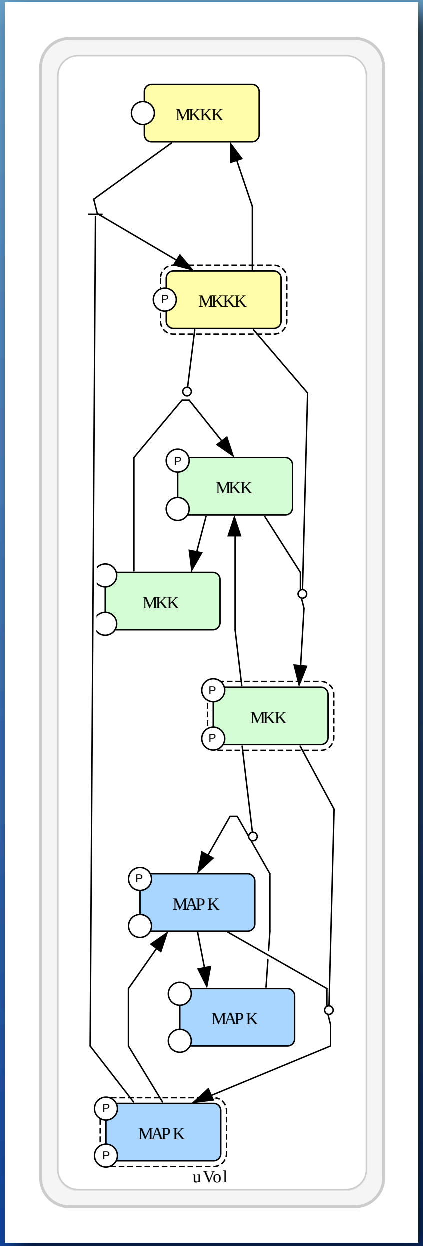
Auto layout



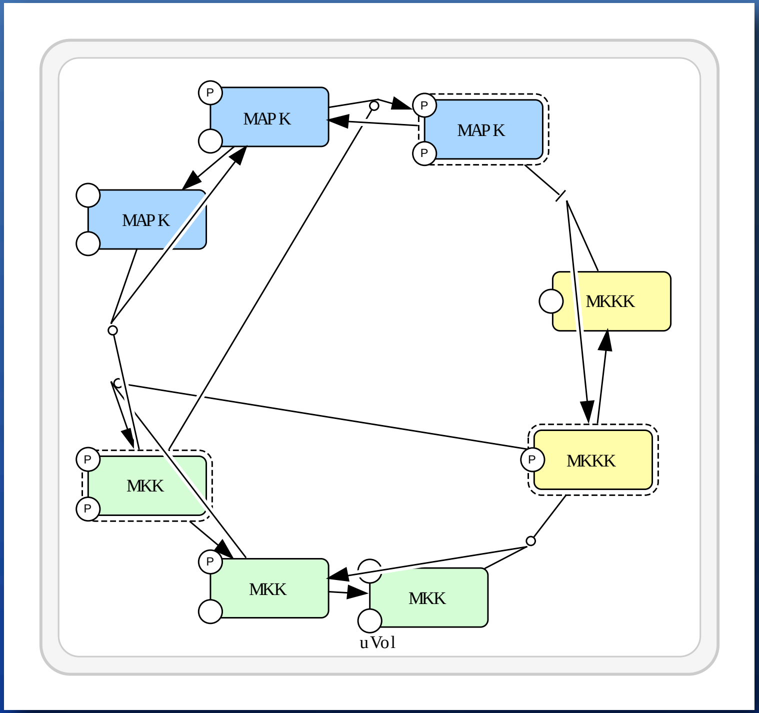
MKKK
MKKK
MKK
MKK
MKK
MAPK
MAPK
MAPK
P
uVol



MAPK
MAPK
MAPK
MKK
MKK
MKK
MKKK
MKKK
P
uVol



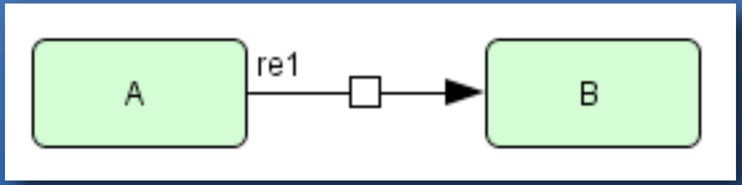
MKKK
MKKK
MKK
MKK
MKK
MAPK
MAPK
MAPK
P
uVol

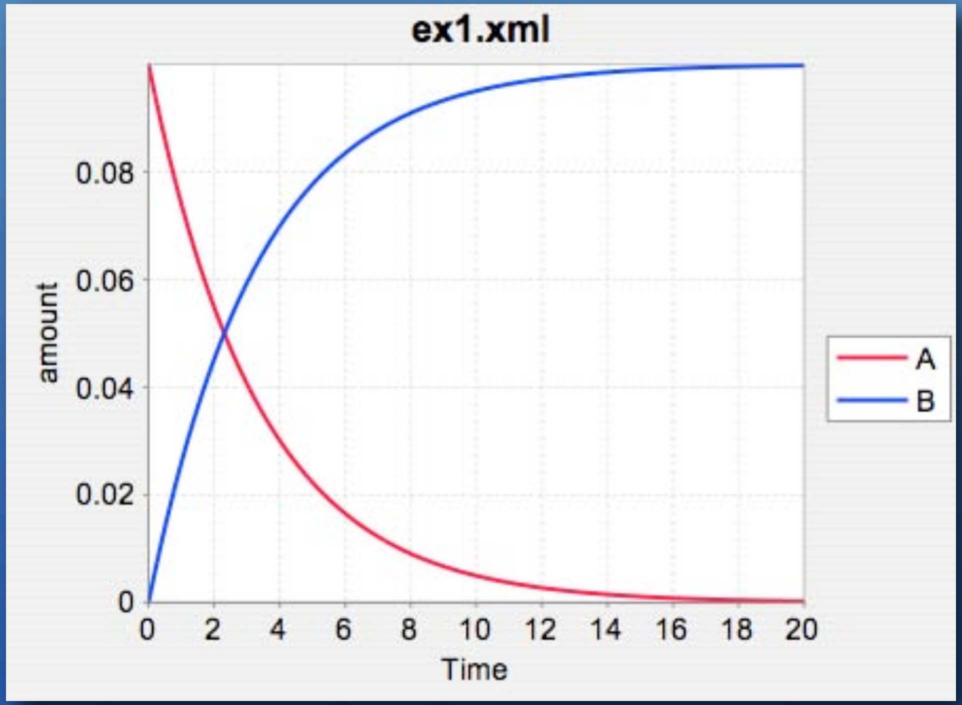


MAPK
MAPK
MAPK
MKKK
MKKK
MKK
MKK
MKK
P
uVol

Simulation (ex1)

- Create following biochemical reaction
- Click [Simulation] → [ControlPanel] and call SBML ODE Solver



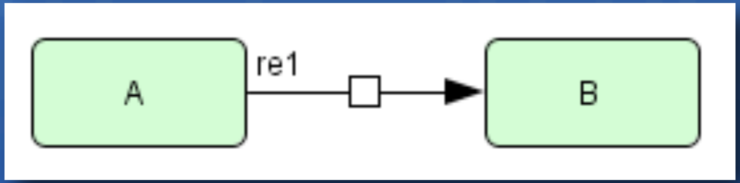


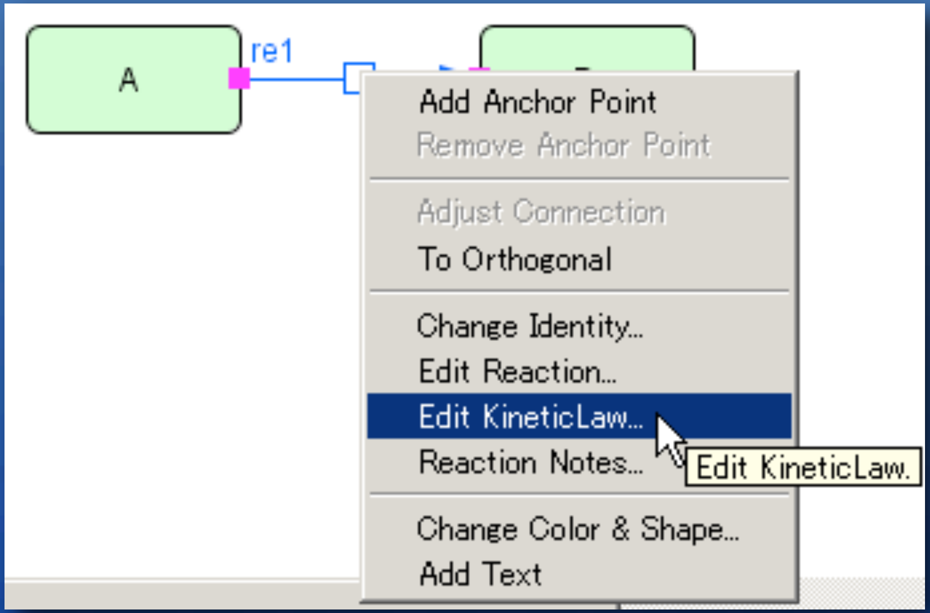
d[B]/d[t] = k * [A]

k = 0.3
A = 0.1
B = 0

Simulation (ex1)

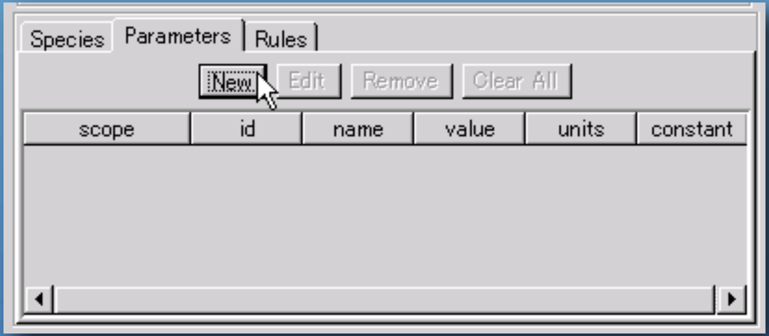
- Create new model (ex1)
- Create reaction
- Right click on the reaction and select [Edit KineticLaw...]





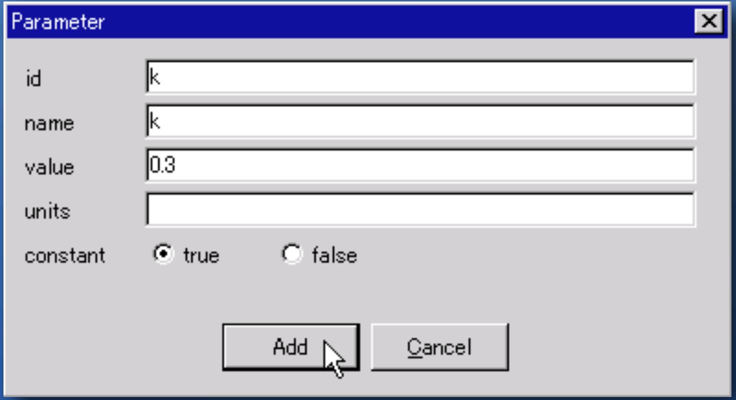
Simulation (ex1)

- Click [New] button on [Parameters] tab

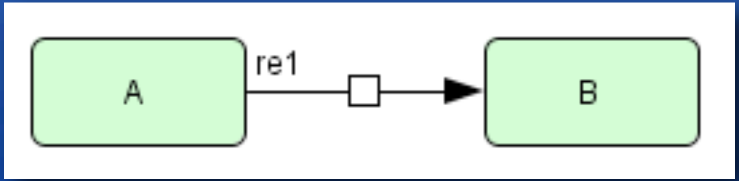


scope	id	name	value	units	constant

- Input values as follows:
  - id: k
  - name: k
  - value: 0.3



id	k
name	k
value	0.3
units	
constant	[✓ true] false



d[B]/d[t] = k * [A]

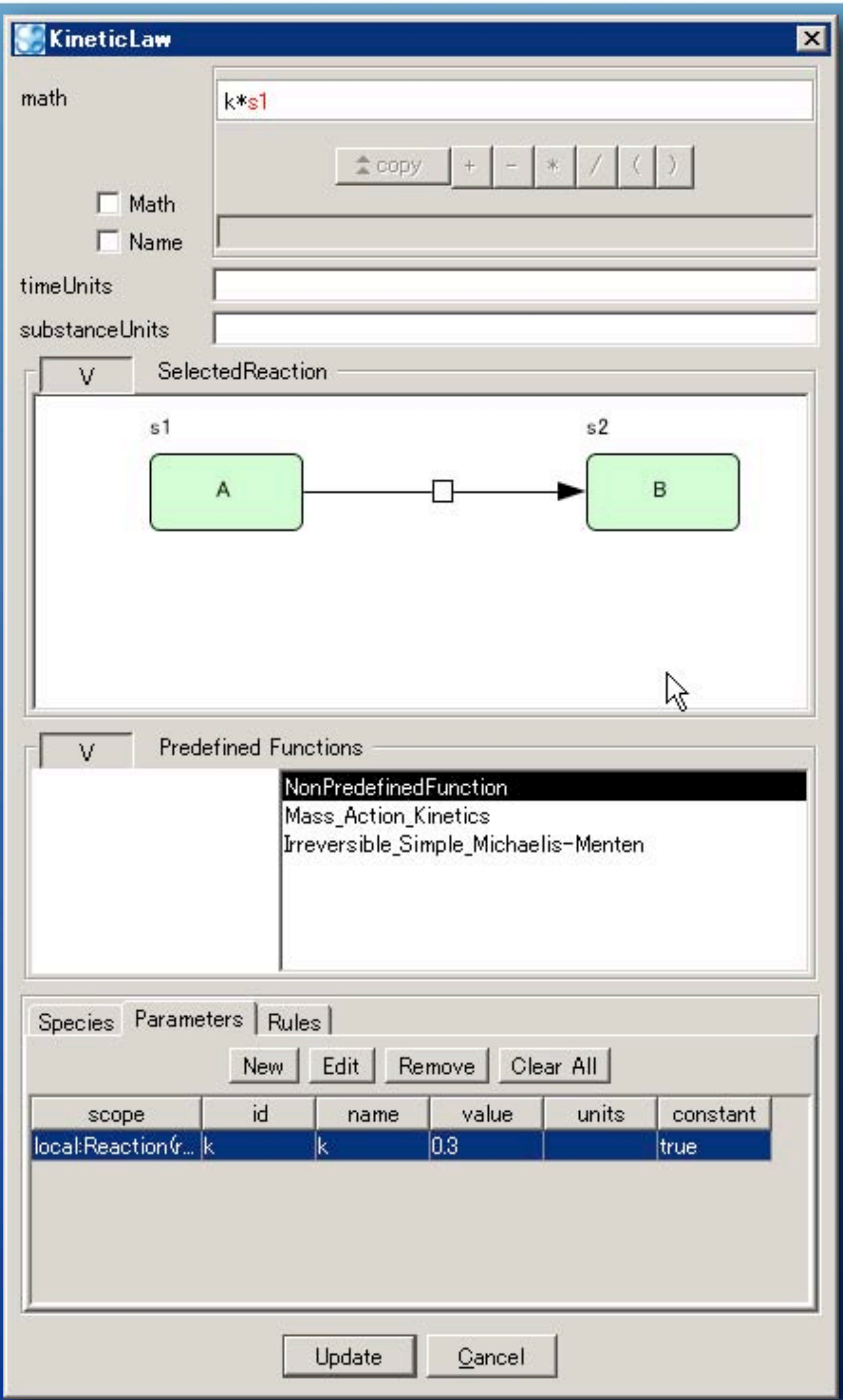
k = 0.3
A = 0.1
B = 0

Simulation (ex1)

- Select parameter "k"
- Click top most text field
- Click [copy] button
- Click [*] button
- Select Protein "A"
- Click top most text field
- Click [copy] button

d[B]/d[t] = k * [A]

k = 0.3
A = 0.1
B = 0



math: k*s1

timeUnits:

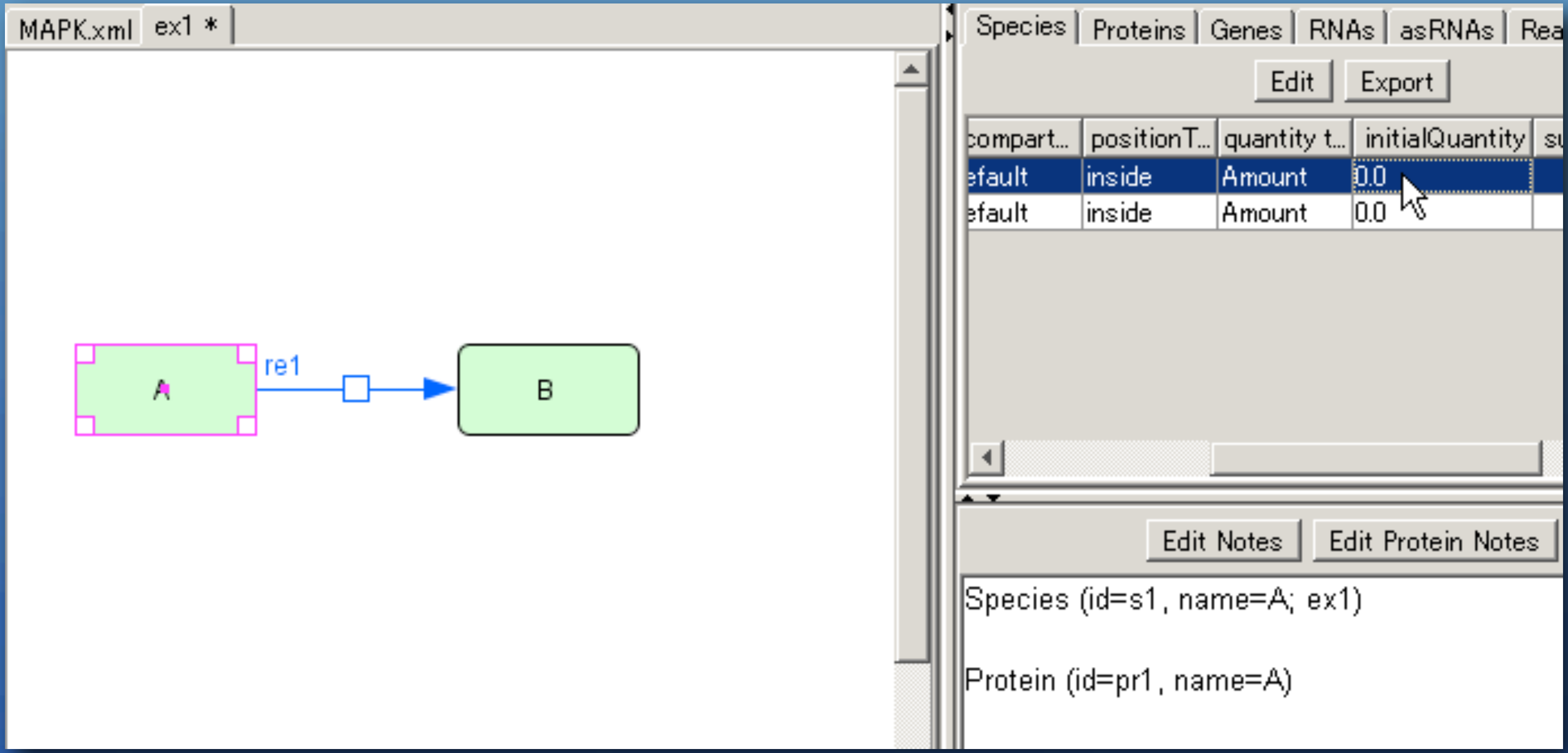
substanceUnits:

NonPredefinedFunction
Mass_Action_Kinetics
Irreversible_Simple_Michaelis-Menten

scope	id	name	value	units	constant
local:Reaction(r...	k	k	0.3		true

Simulation (ex1)

- Double click [initialQuantity] column for Protein "A"



compart...	positionT...	quantity t...	initialQuantity	su
efault	inside	Amount	0.0	
efault	inside	Amount	0.0	

Species (id=s1, name=A; ex1)

Protein (id=pr1, name=A)

- Set value as 0.1

$$\frac{d[B]}{d[t]} = k * [A]$$

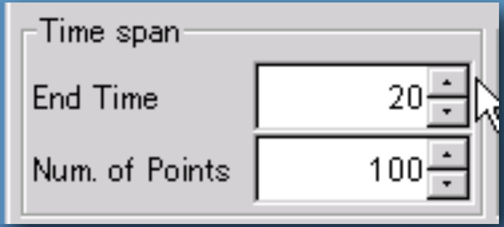
k = 0.3

A = 0.1

B = 0

Simulation (ex1)

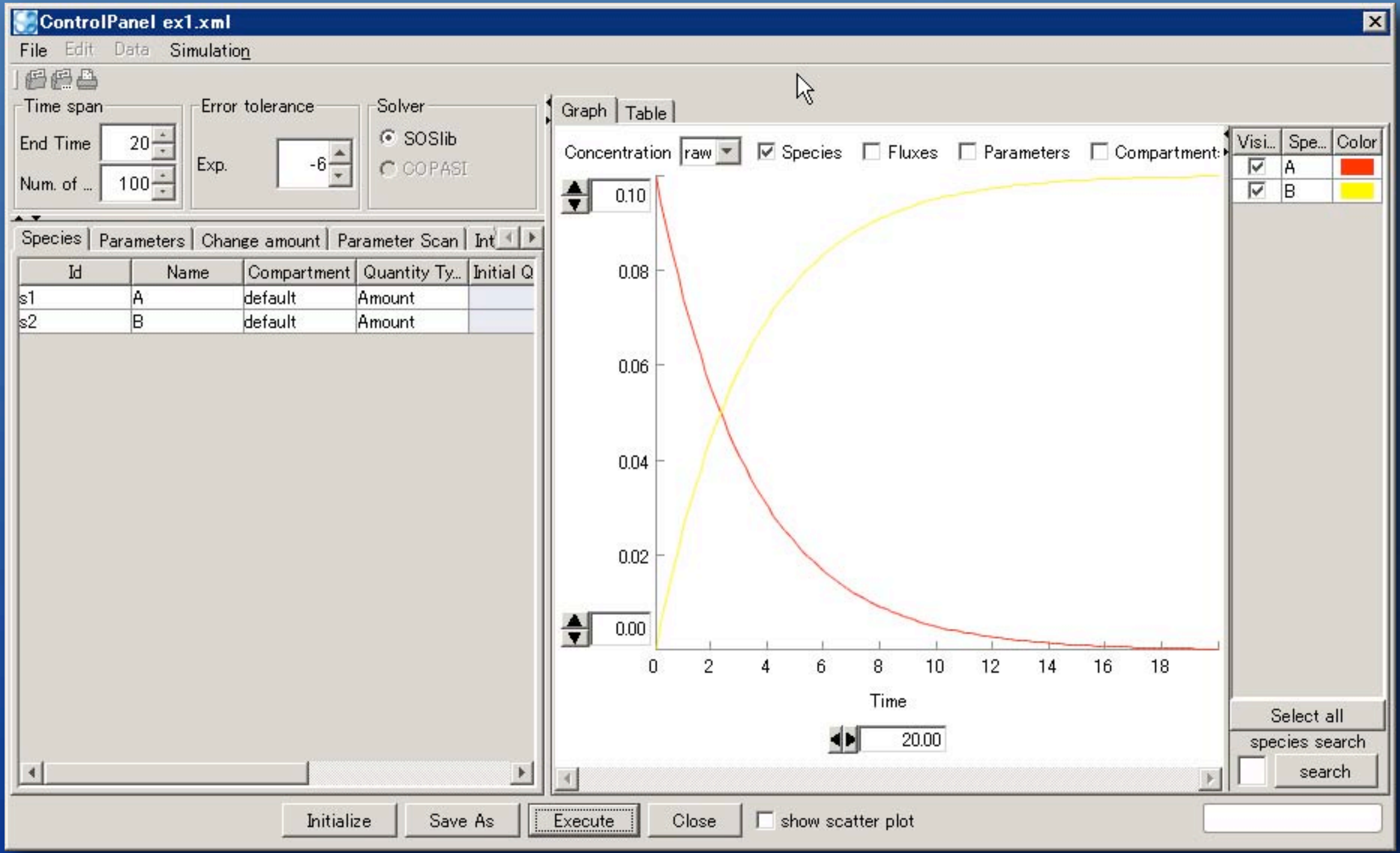
- Click [Simulation] → [ControlPanel]
- Set [End Time] to 20
- Click [Execute] button



Time span

End Time 20

Num. of Points 100



ControlPanel ex1.xml

File Edit Data Simulation

Time span: End Time 20, Num. of ... 100

Error tolerance: Exp. -6

Solver: [✓ SOSlib] COPASI

Species | Parameters | Change amount | Parameter Scan | Int

Id	Name	Compartment	Quantity Ty...	Initial Q
s1	A	default	Amount	
s2	B	default	Amount	

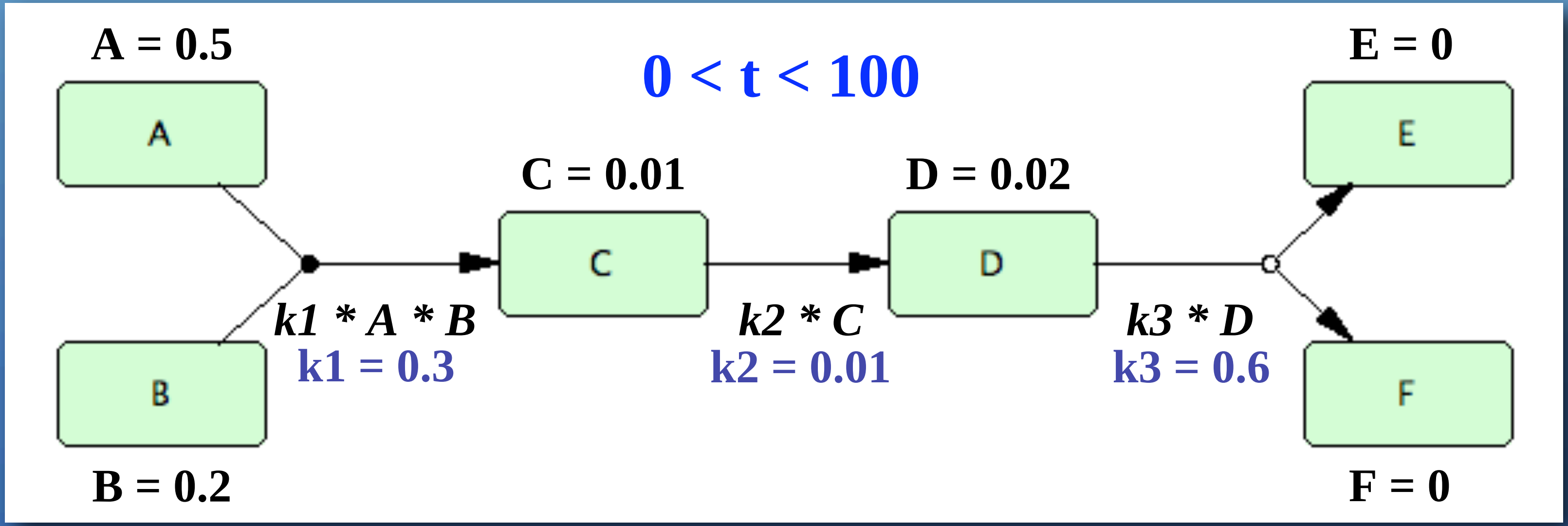
Graph | Table

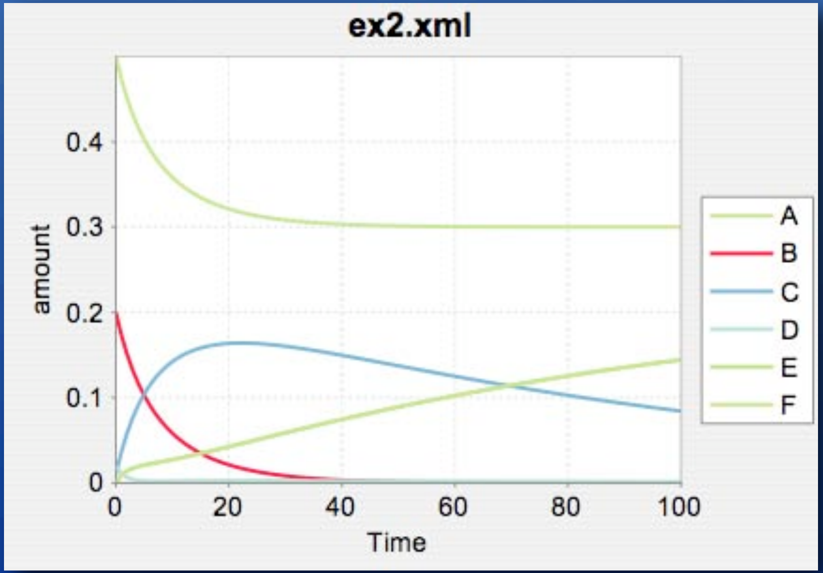
Concentration raw; [✓] Species, [ ] Fluxes, [ ] Parameters, [ ] Compartment

Visi...	Spe...	Color
✓	A	🟥
✓	B	🟨

Simulation (ex2)

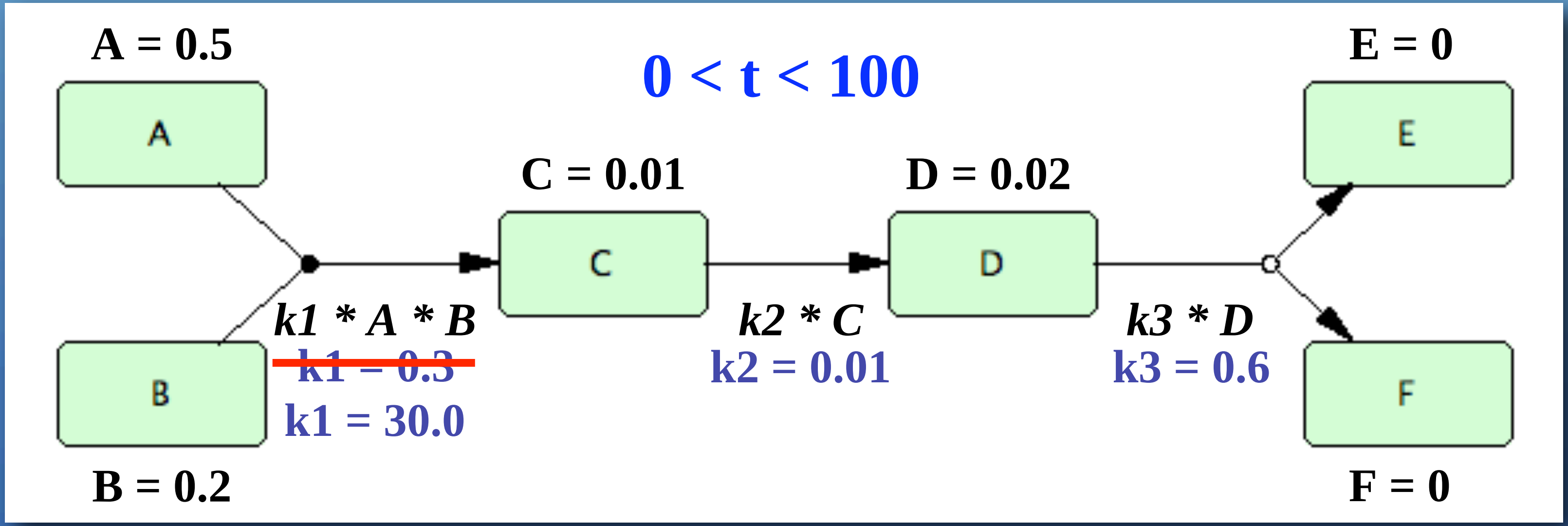
- Create following biochemical reactions
- Execute simulation from [ControlPanel]

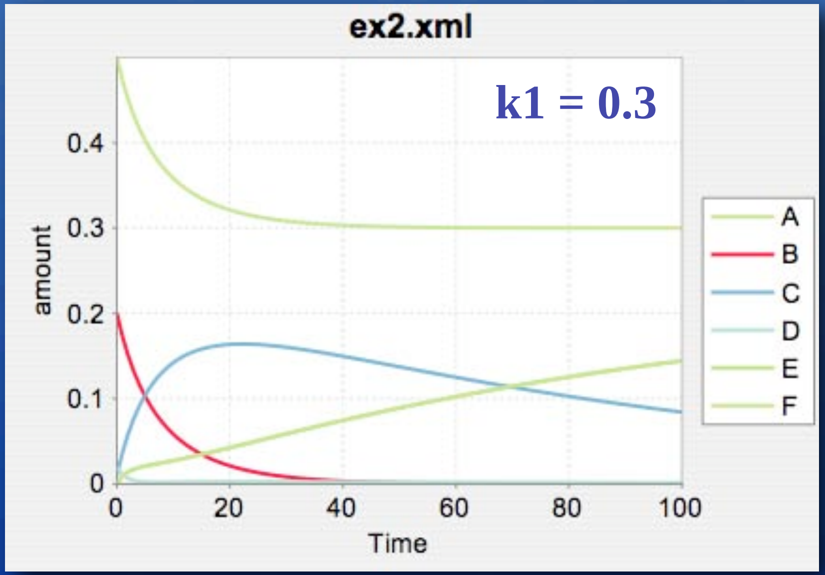


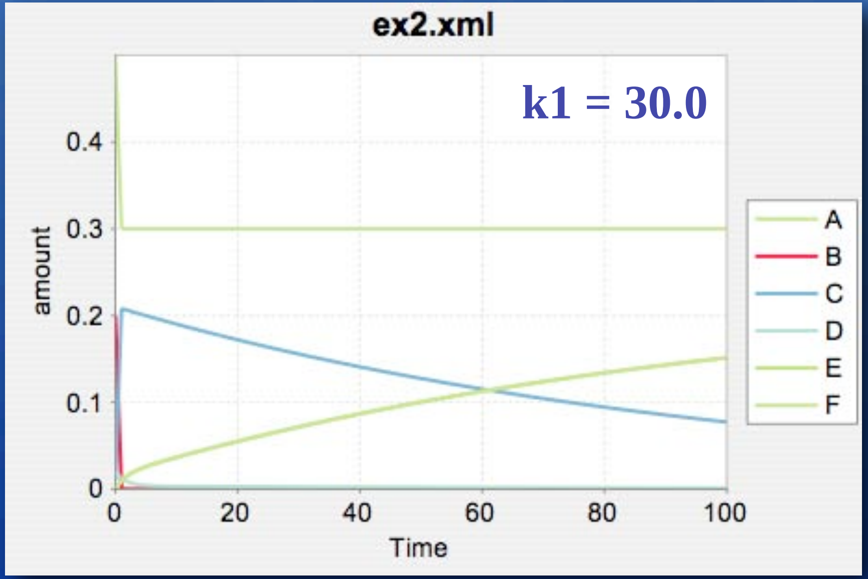


Simulation (ex2)

- Change parameter k1 to 30.0

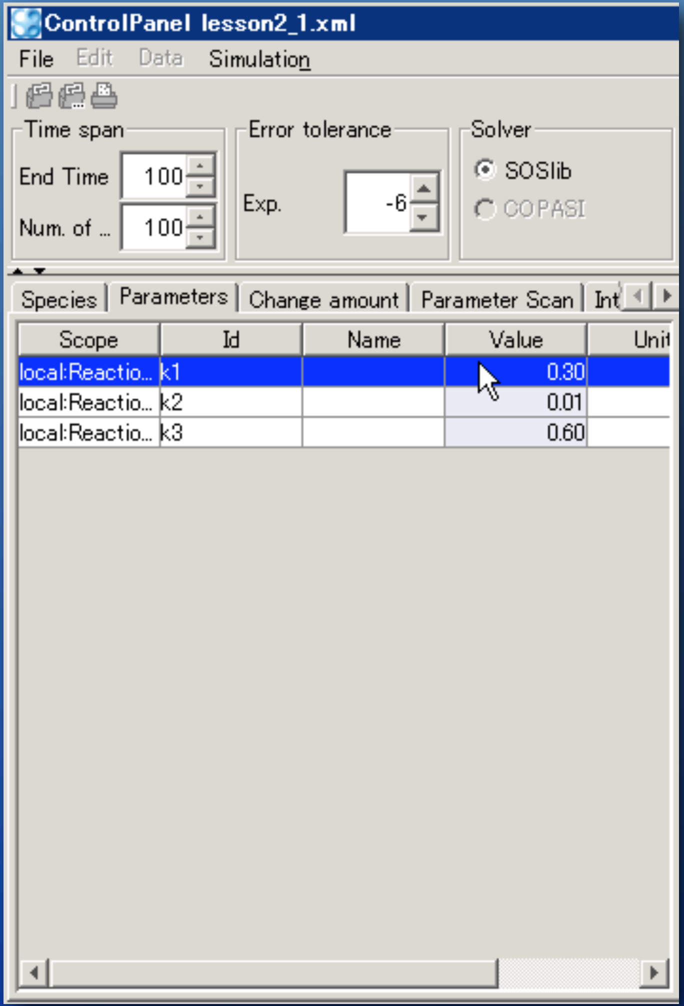






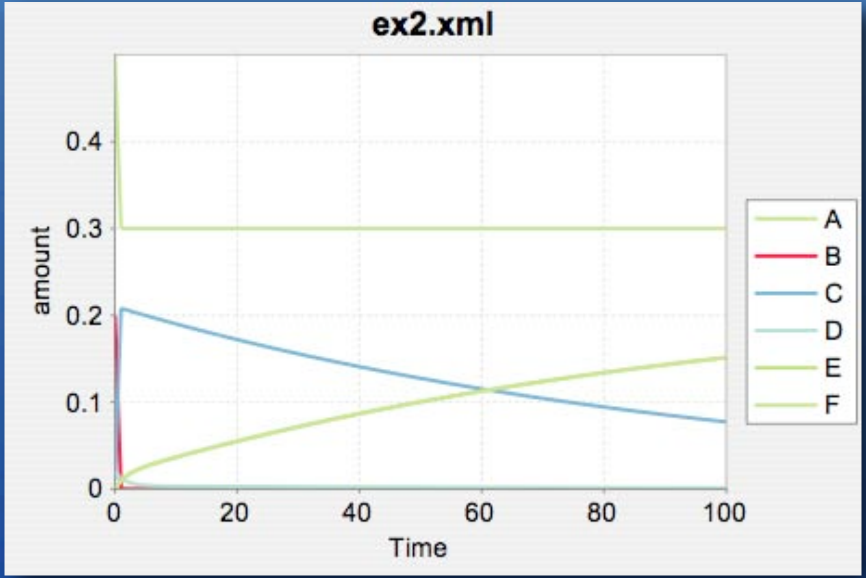
Simulation (ex2)

- Click [Parameters] tab
- Double click [Value] column for k1
- Change parameter k1 to 30.0



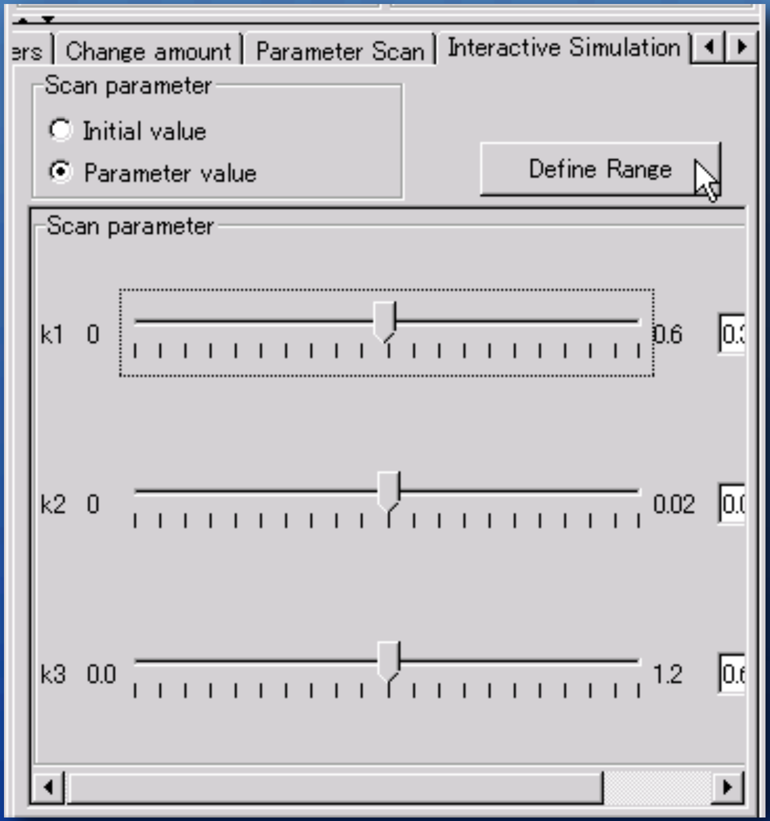
End Time 100
Num. of ... 100
Exp. -6
Solver: [✓ SOSlib] COPASI

Scope	Id	Name	Value	Unit
local:Reactio...	k1		0.30	
local:Reactio...	k2		0.01	
local:Reactio...	k3		0.60	



Simulation (ex2)

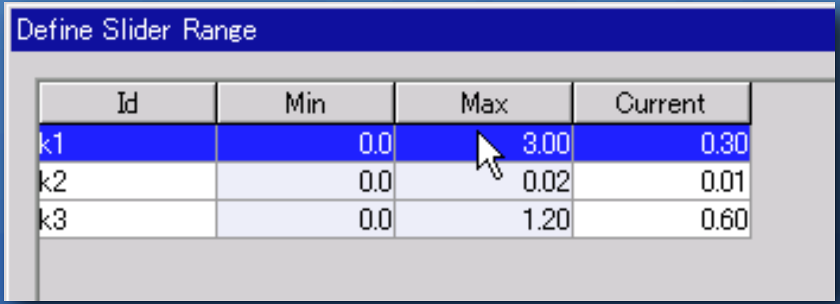
- Click [Interactive Simulation] tab
- Click [Parameter value] radio button
- Click [Define Range] button
- Click [Max] column for k1 and set value as 3.0



Scan parameter

( ) Initial value

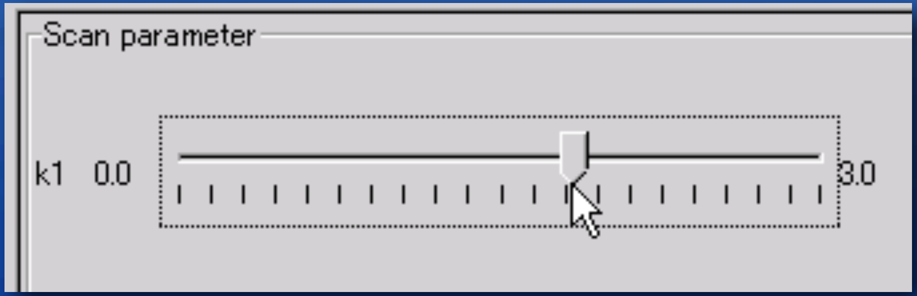
[✓ Parameter value]



Define Slider Range

Id	Min	Max	Current
k1	0.0	3.00	0.30
k2	0.0	0.02	0.01
k3	0.0	1.20	0.60

Drag sliderbar for k1

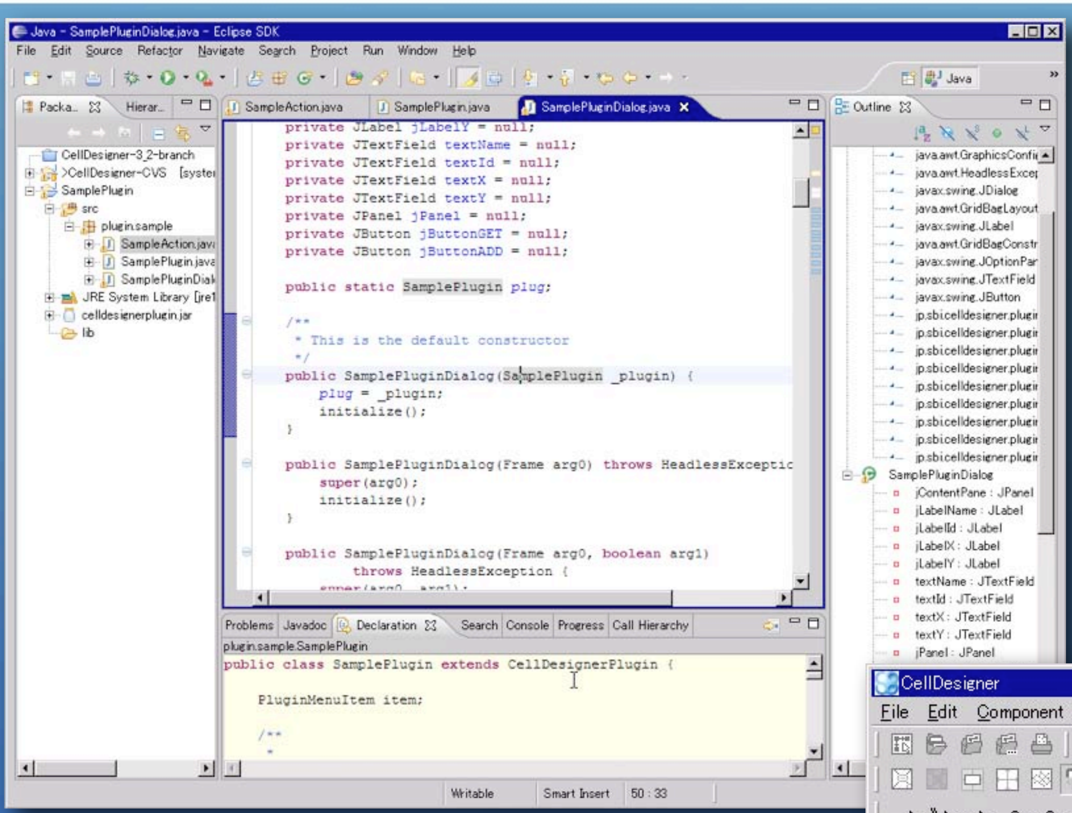


The Systems Biology Institute

Plugin development

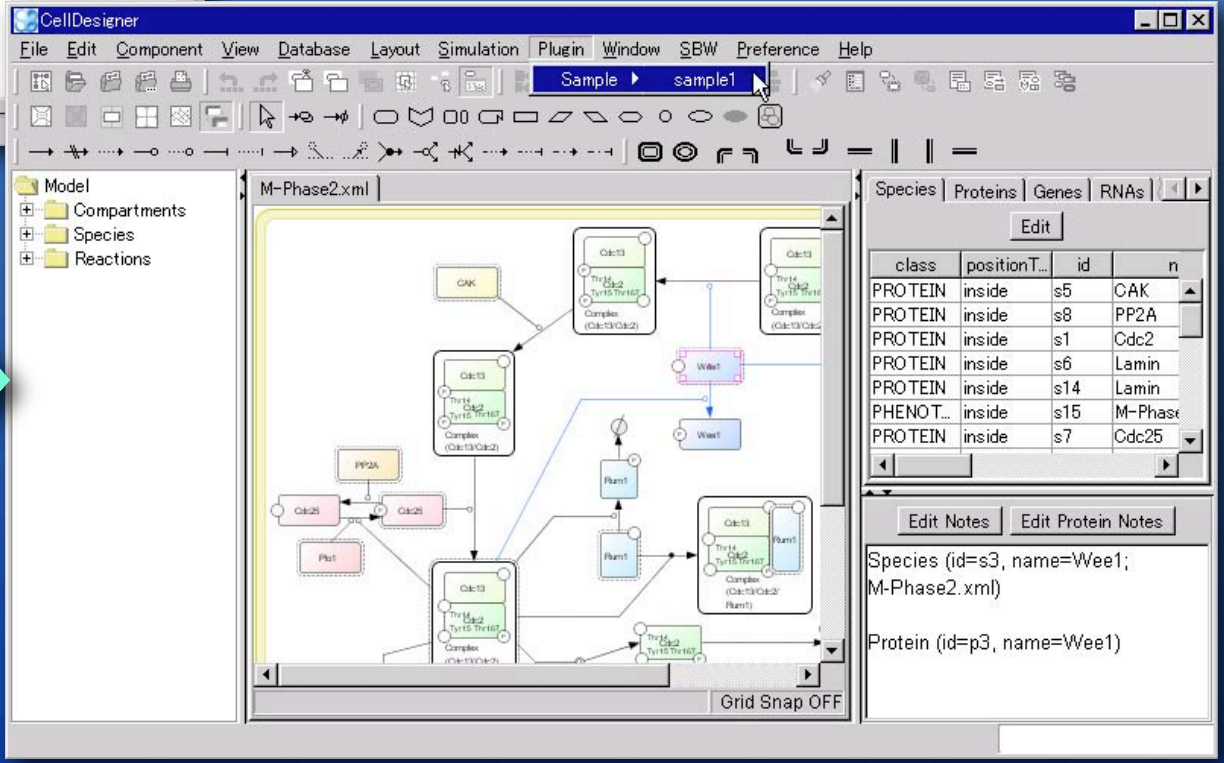
Plugin development

- Develop plugin on Eclipse
- Call plugin from [Plugin] menu on CellDesigner



```
    private JLabel jLabelY = null;
    private JTextField textName = null;
    private JTextField textId = null;
    private JTextField textX = null;
    private JTextField textY = null;
    private JPanel jPanel = null;
    private JButton jButtonGET = null;
    private JButton jButtonADD = null;

    public static SamplePlugin plug;

    /**
     * This is the default constructor
     */
    public SamplePluginDialog(SamplePlugin _plugin) {
        plug = _plugin;
        initialize();
    }

    public SamplePluginDialog(Frame arg0) throws HeadlessExceptic
        super(arg0);
        initialize();
    }

    public SamplePluginDialog(Frame arg0, boolean arg1)
            throws HeadlessException {
```

```
plugin.sample.SamplePlugin
public class SamplePlugin extends CellDesignerPlugin {

    PluginMenuItem item;

    /**
```

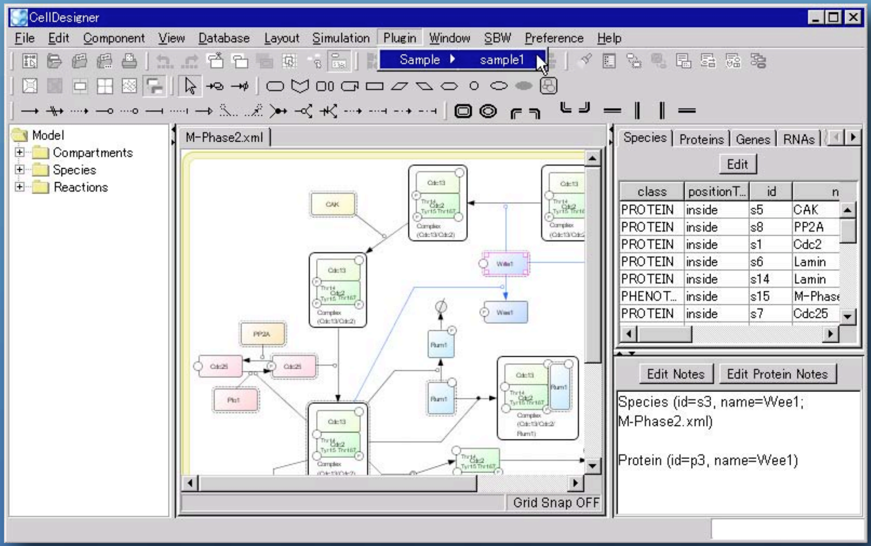


class	positionT...	id	n
PROTEIN	inside	s5	CAK
PROTEIN	inside	s8	PP2A
PROTEIN	inside	s1	Cdc2
PROTEIN	inside	s6	Lamin
PROTEIN	inside	s14	Lamin
PHENOT...	inside	s15	M-Phase
PROTEIN	inside	s7	Cdc25

Species (id=s3, name=Wee1; M-Phase2.xml)

Protein (id=p3, name=Wee1)

Plugin



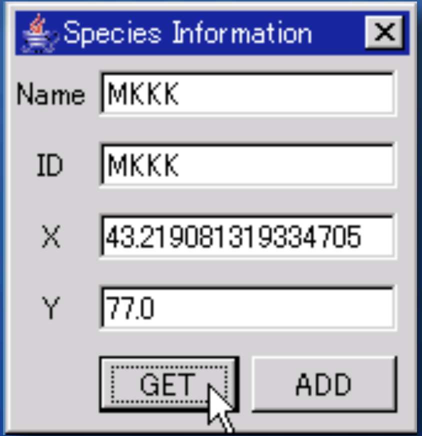
class	positionT...	id	n
PROTEIN	inside	s5	CAK
PROTEIN	inside	s8	PP2A
PROTEIN	inside	s1	Cdc2
PROTEIN	inside	s6	Lamin
PROTEIN	inside	s14	Lamin
PHENOT...	inside	s15	M-Phase
PROTEIN	inside	s7	Cdc25

Species (id=s3, name=Wee1;
M-Phase2.xml)

Protein (id=p3, name=Wee1)

CellDesigner

- Get object (species, reaction, etc.) information
- Add / modify object (species, reaction, etc.)



Name	MKKK
ID	MKKK
X	43.219081319334705
Y	77.0

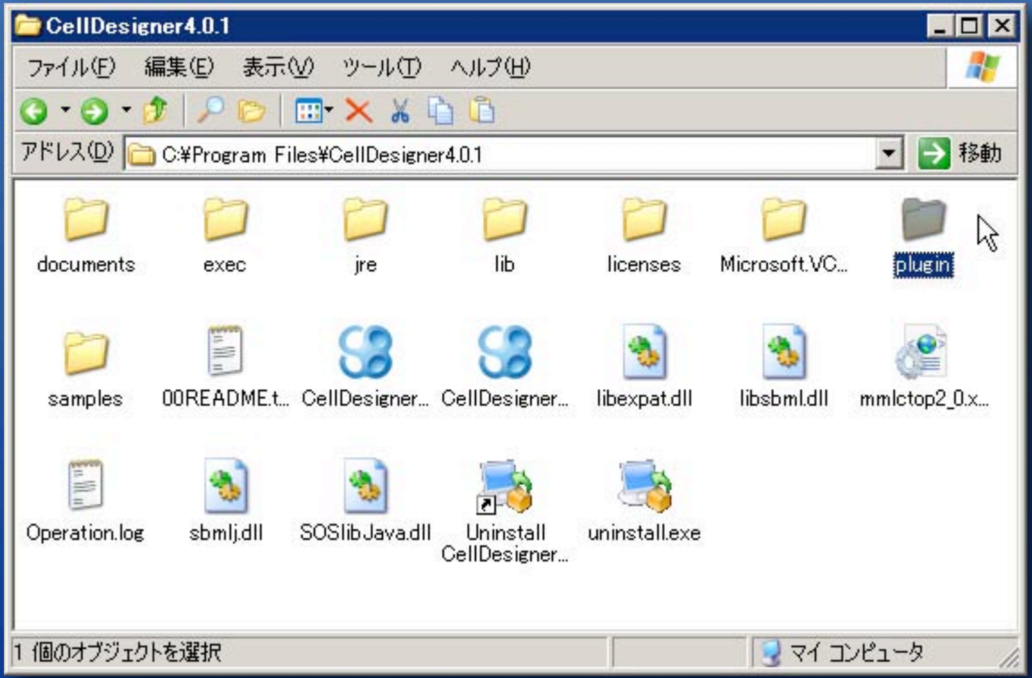
Plugin

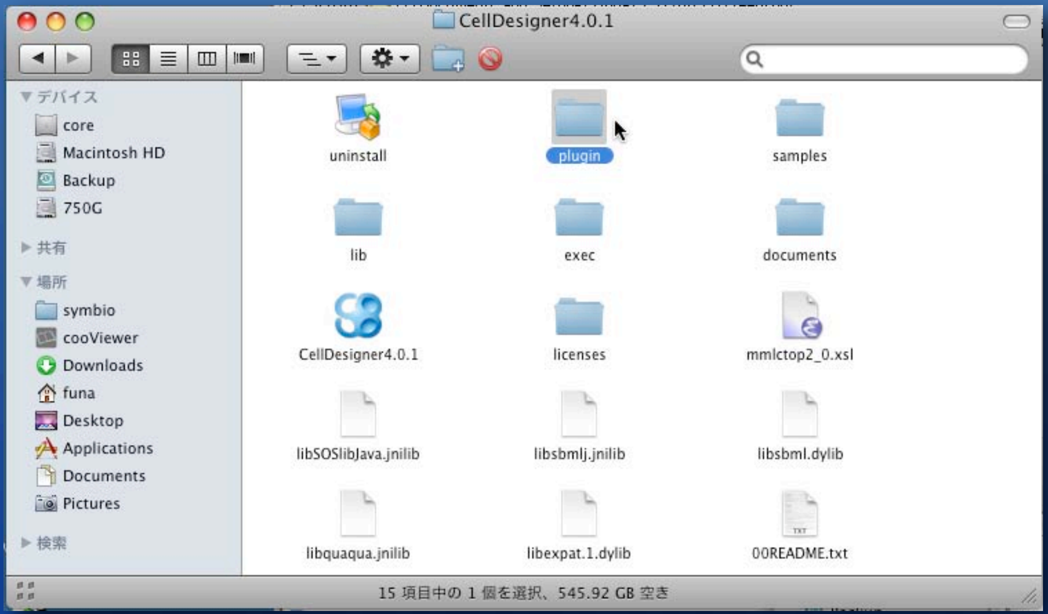
Development environment

- CellDesigner 4.0 or higher
- JDK 1.5.0 or higher
- Eclipse 3.4.0 (may work on earlier version)

How to Install Plugins

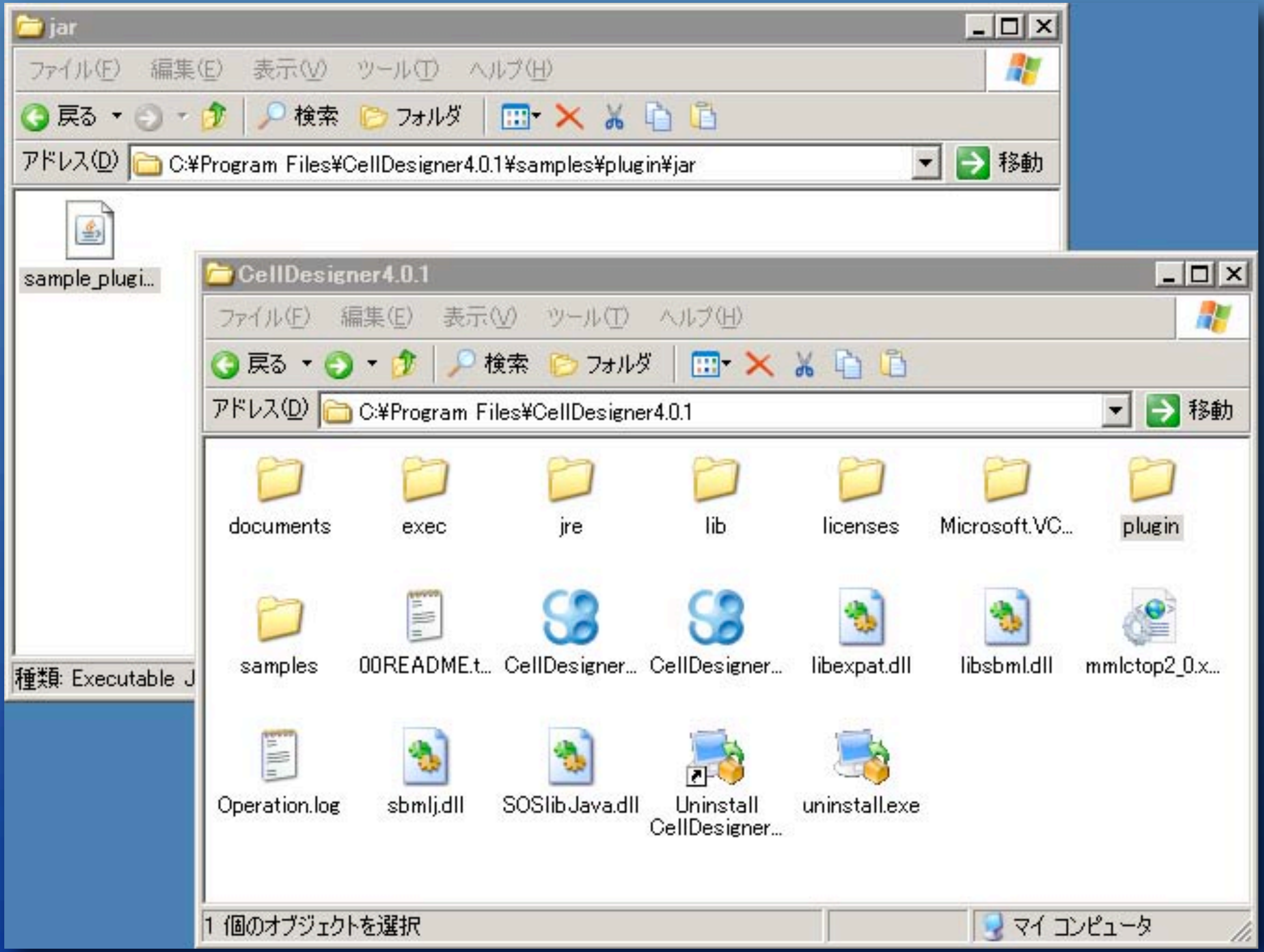
- Copy plugin file (.jar file) to CellDesigner's plugin folder
  - Windows: C:/Program Files/CellDesigner4.0.1/plugin
  - MacOSX: /Applications/CellDesigner4.0.1/plugin





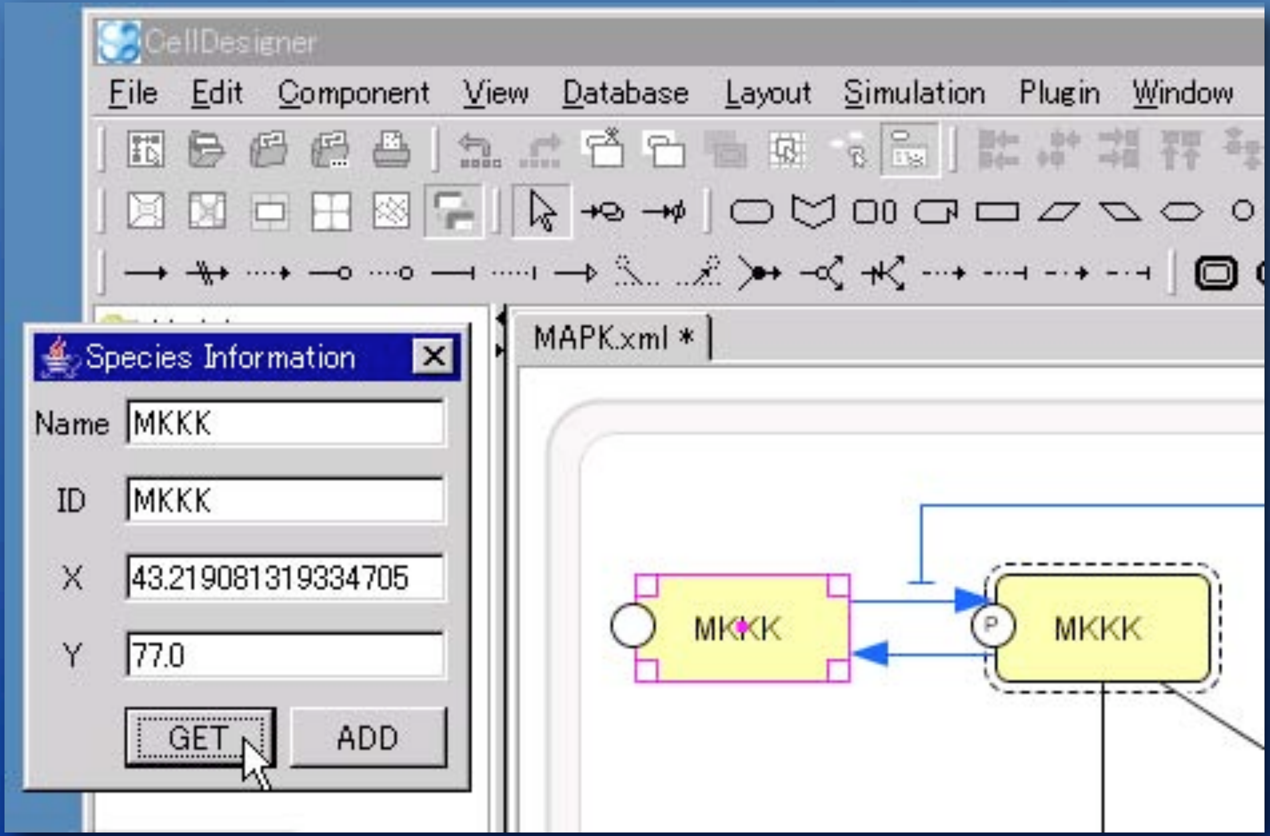
Sample plugin

- Copy **sample_plugin.jar** in samples/plugin/jar folder to **plugin** folder
- Restart CellDesigner



Sample plugin

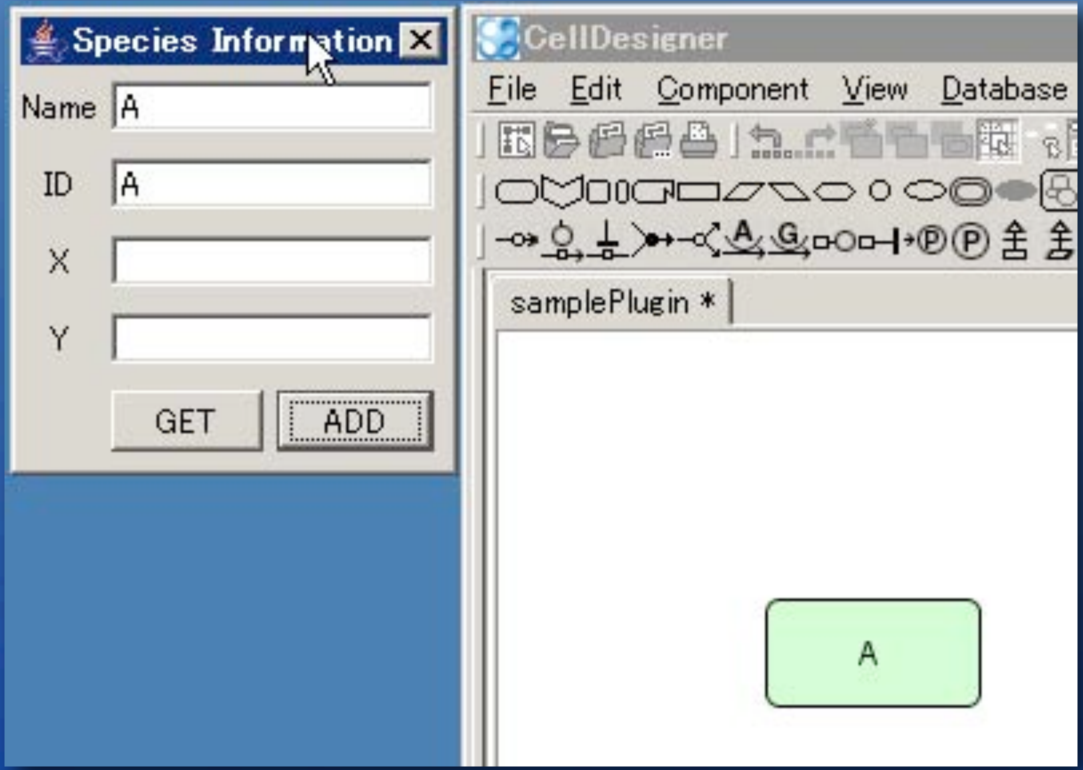
- [File] → [Open] → samples/MAPK.xml
- [Plugin] → [Sample Plugin1] → [Open Sample Plugin1 dialog]
- Select MKKK and click [GET]


CellDesigner
File Edit Component View Database Layout Simulation Plugin Window
Species Information
GET
ADD
MAPK.xml *
MKKK
MKKK
P

Name	MKKK
ID	MKKK
X	43.219081319334705
Y	77.0

Sample plugin

- Create new model
- Input Species Information and click [ADD]



Species Information

Name: A

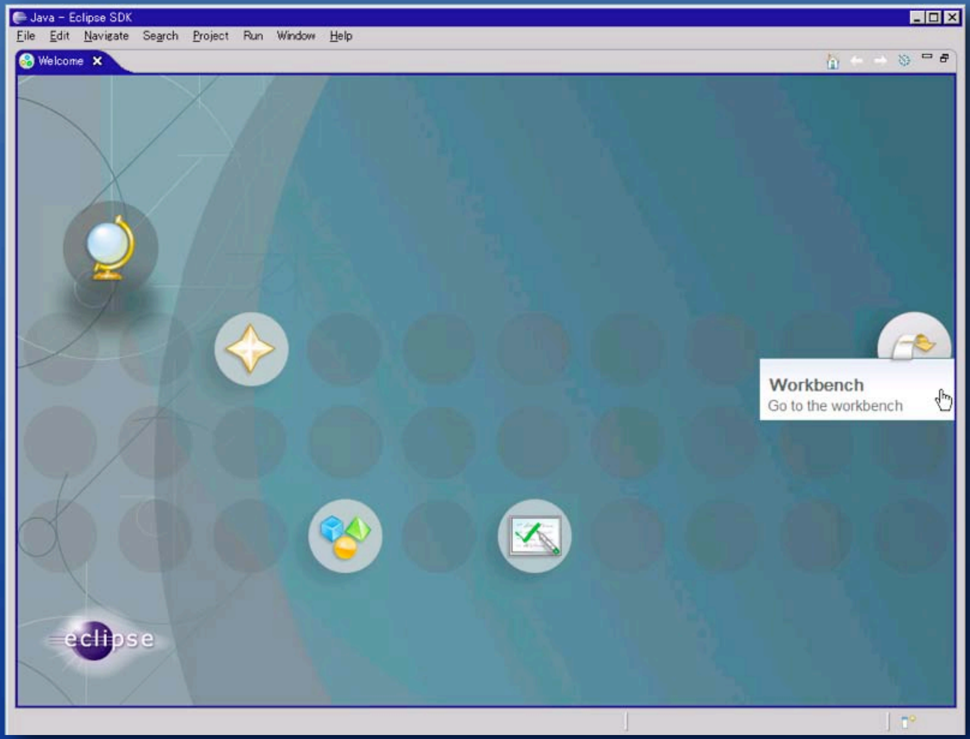
ID: A

X:

Y:

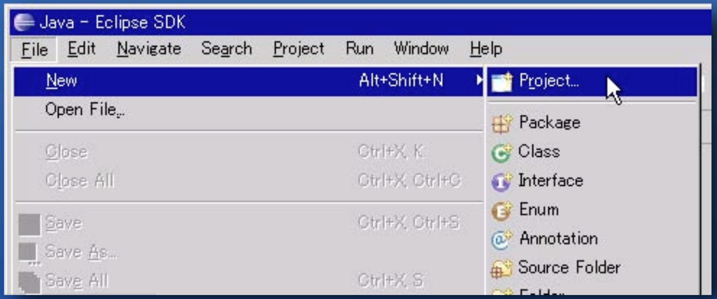
How to build your plugin

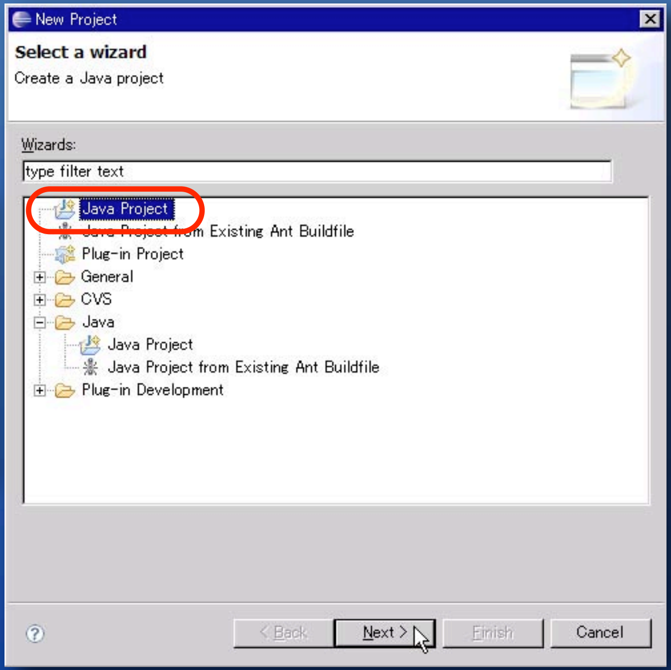
- Download Eclipse 3.4 from
  - http://www.eclipse.org/
- Launch Eclipse and specify your workspace (ex. Desktop/workspace)
- Click [Workbench] icon



Create new project

- [File] → [New] → [Project]
- Select "**Java Project**" and click [Next]
- Input "Project name" (**MyPlugin**) and select [**Create separate source and output folders**]

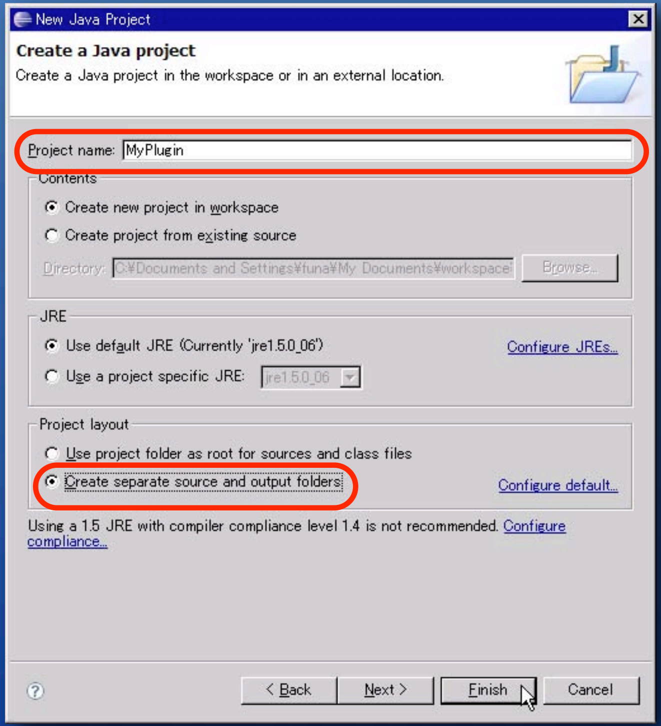



New Project

Select a wizard

Create a Java project

Wizards: type filter text


New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: MyPlugin

Contents
- [✓ Create new project in workspace]
- Create project from existing source

Directory: C:¥Documents and Settings¥funa¥My Documents¥workspace

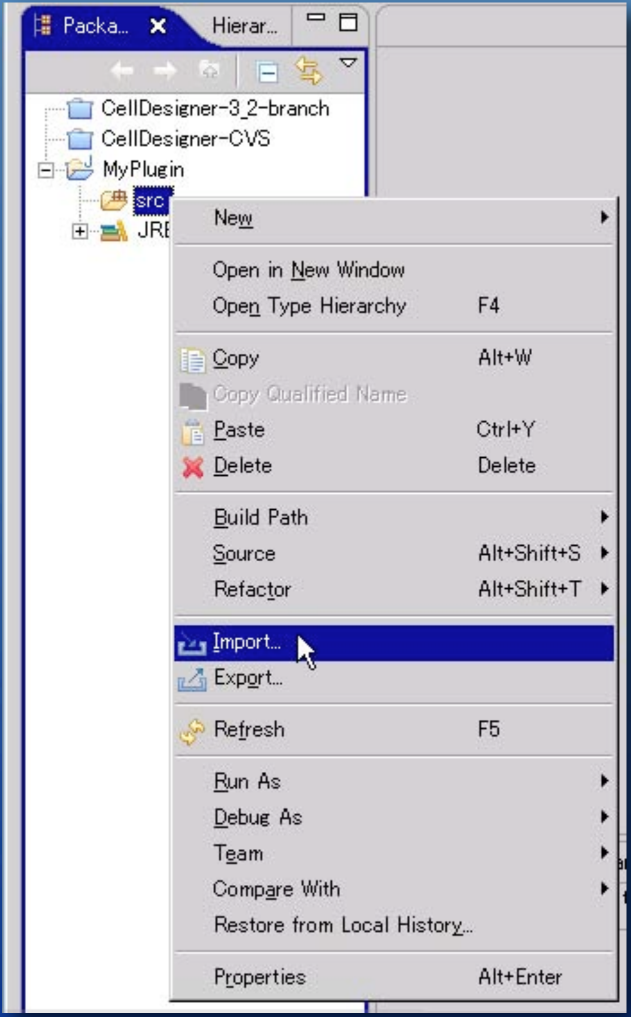
JRE
- [✓ Use default JRE (Currently 'jre1.5.0_06')]
- Use a project specific JRE: jre1.5.0_06

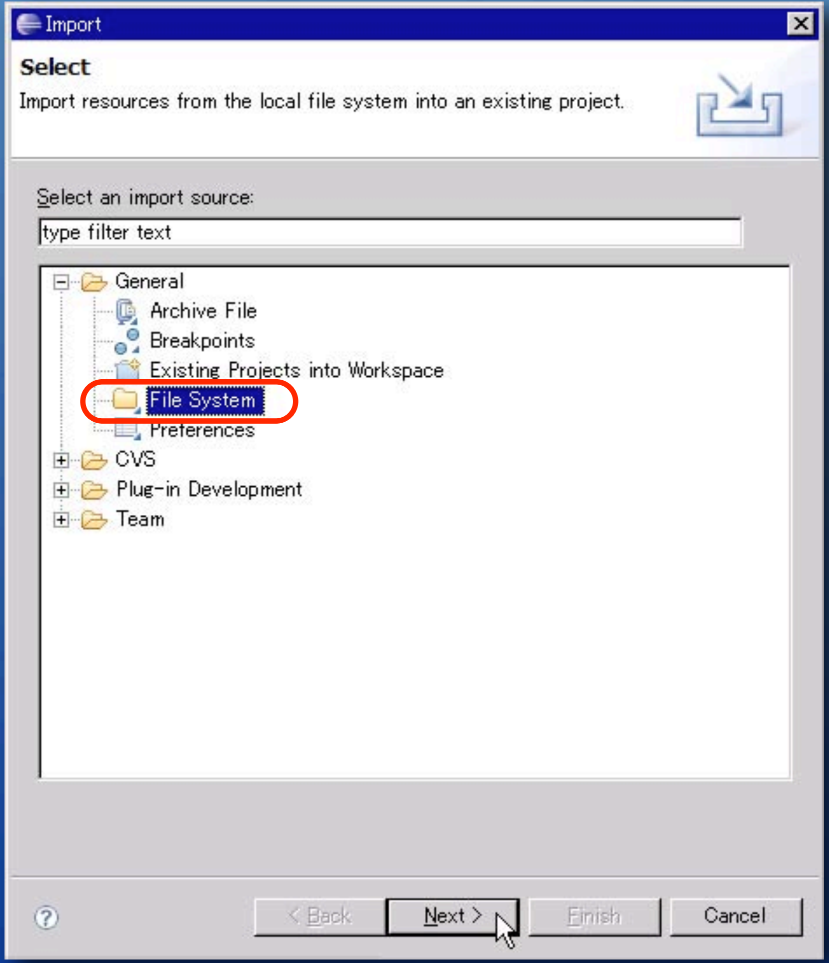
Project layout
- Use project folder as root for sources and class files
- [✓ Create separate source and output folders]

Using a 1.5 JRE with compiler compliance level 1.4 is not recommended. Configure compliance...

Import sample source

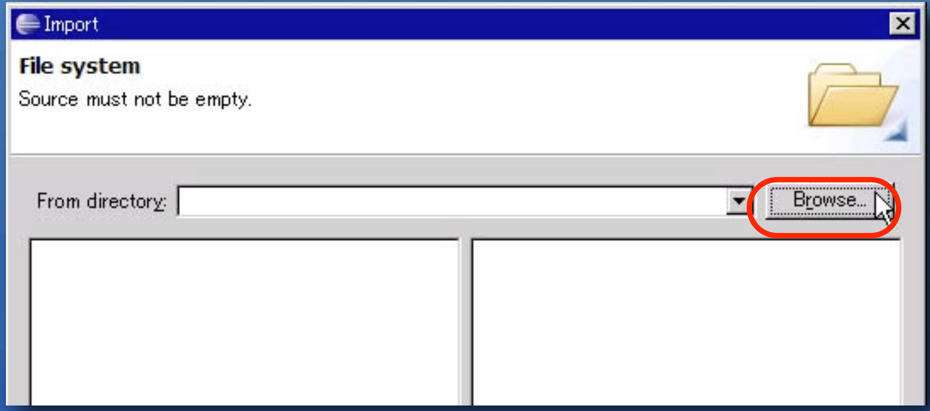
- Click [+] button next to [MyPlugin]
- Right click "src" folder and click [Import]
- Select [File system] and click [Next]





Import source file

- Click [Browse] button next to “From directory”
- Select “C:\Program Files\CellDesigner4.0.1\samples\plugin\src” and click [OK]
- Click check box next to “src” folder

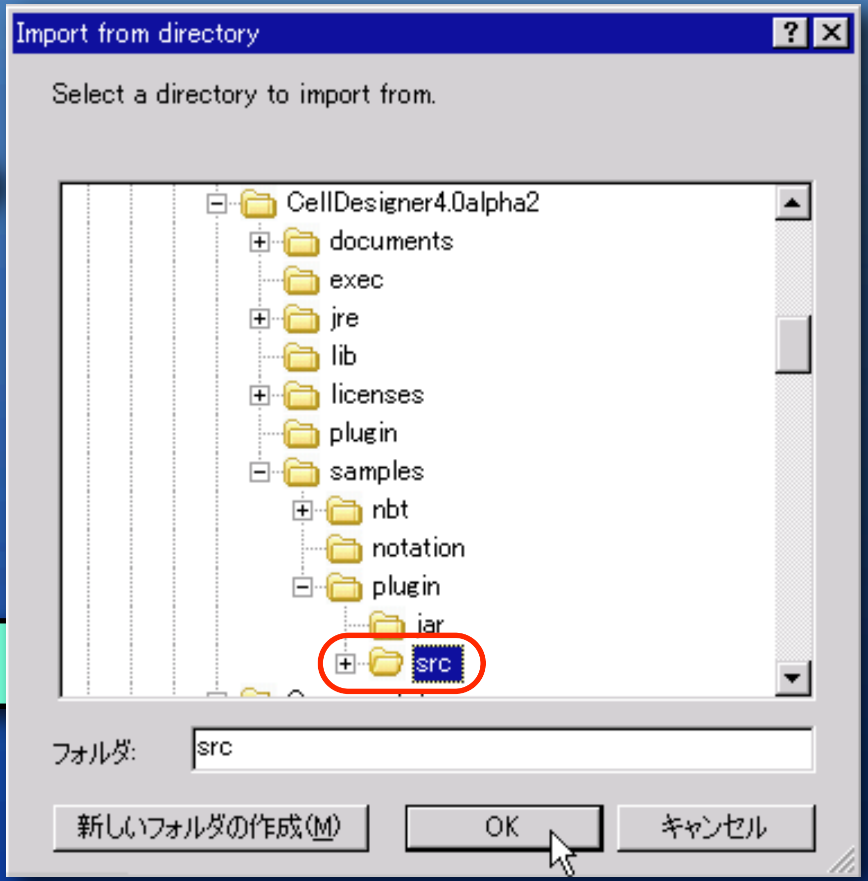


Import

File system

Source must not be empty.

From directory: [ ] Browse...



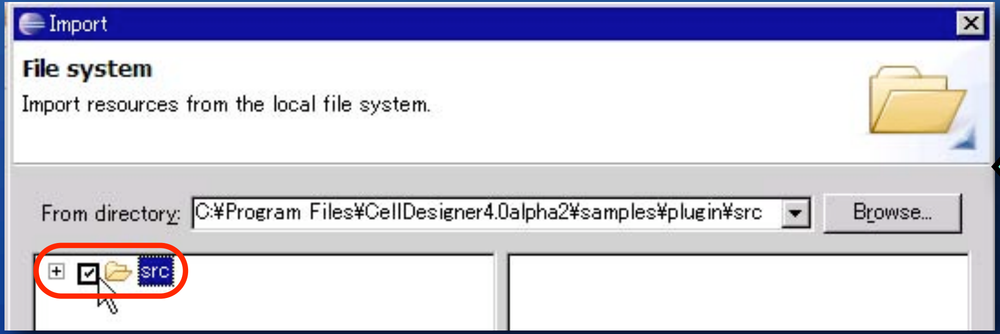
Import from directory

Select a directory to import from.

```
⊟ CellDesigner4.0alpha2
   ⊞ documents
     exec
   ⊞ jre
     lib
   ⊞ licenses
     plugin
   ⊟ samples
      ⊞ nbt
        notation
      ⊟ plugin
          jar
        ⊞ src
```

フォルダ: src

新しいフォルダの作成(M) | OK | キャンセル



Import

File system

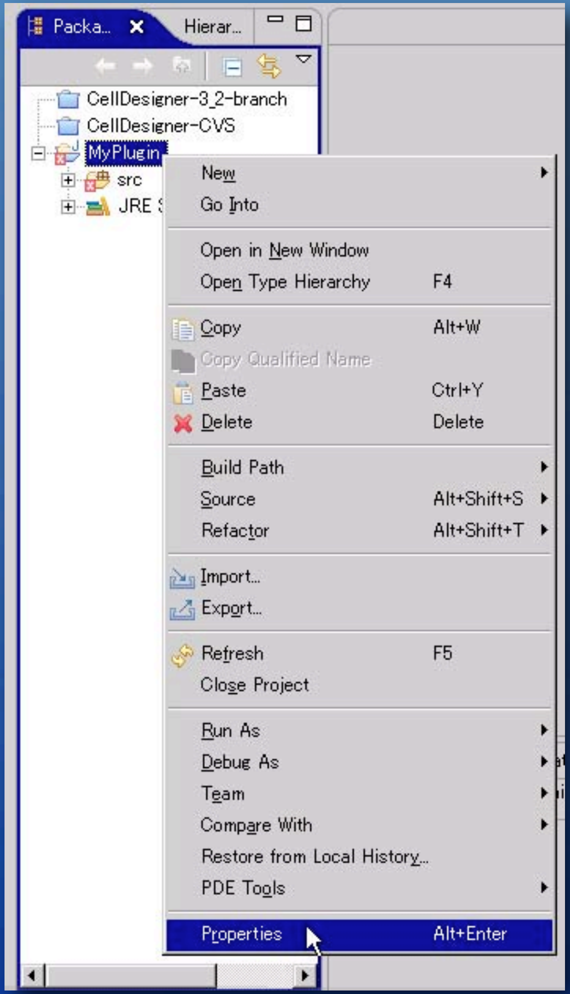
Import resources from the local file system.

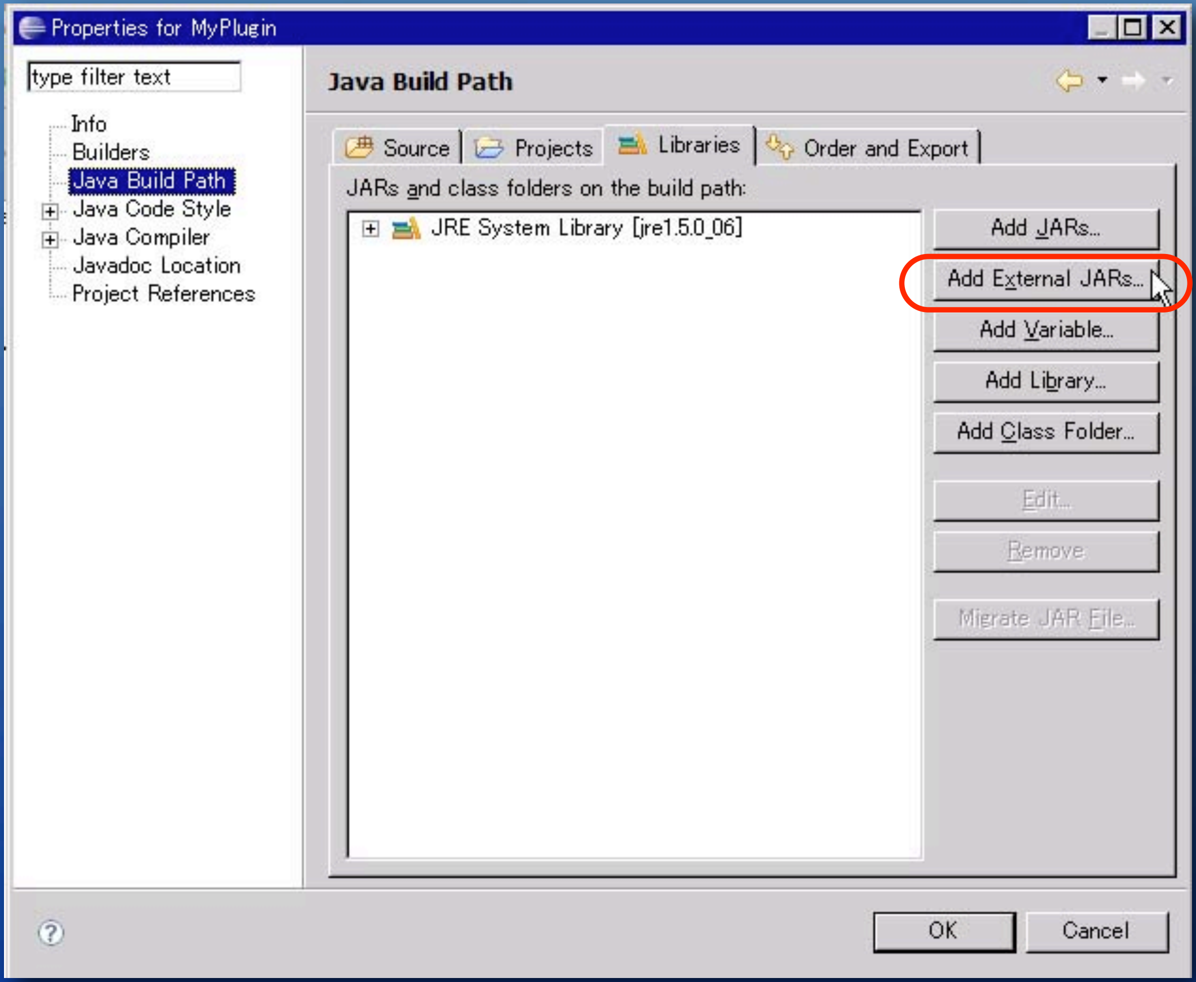
From directory: C:¥Program Files¥CellDesigner4.0alpha2¥samples¥plugin¥src Browse...

⊞ ☑ src

Select Java Build Path

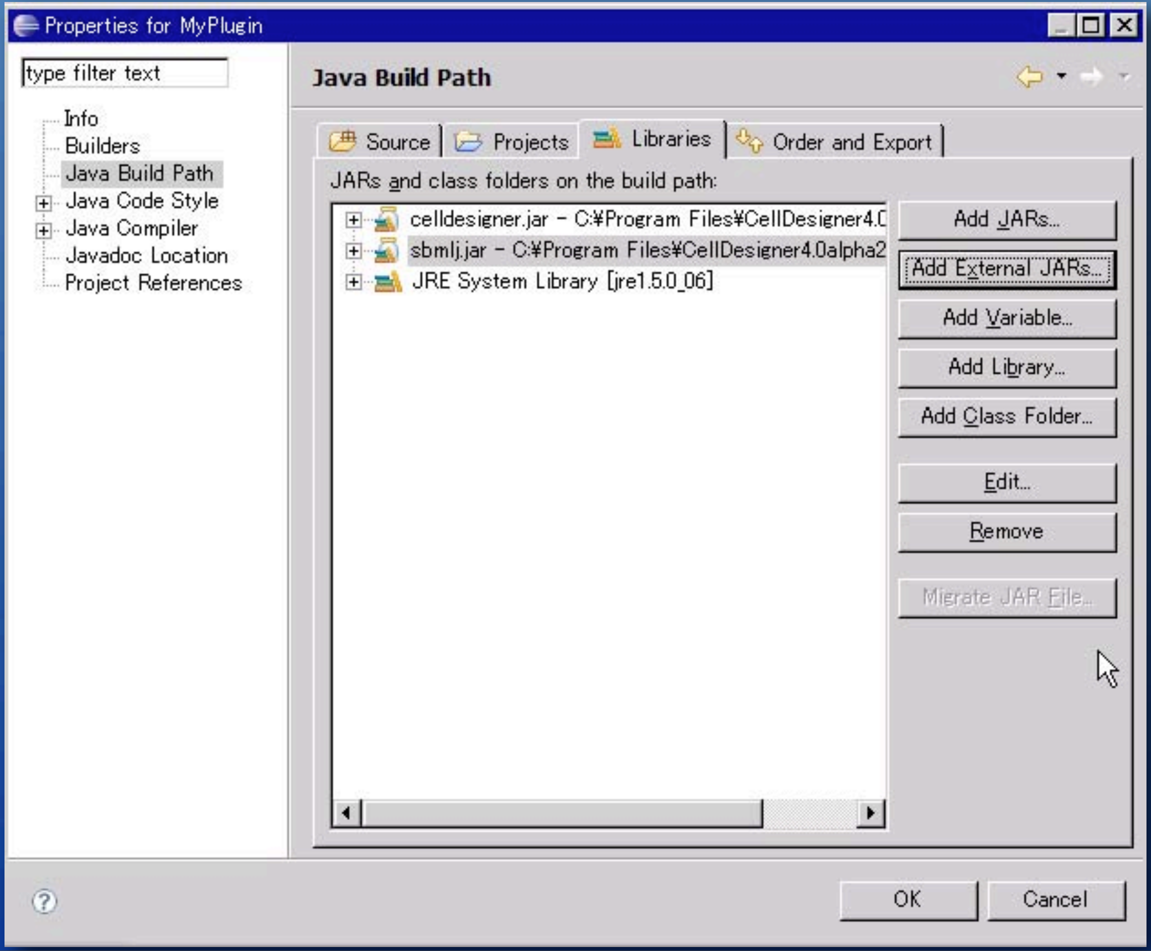
- Right click [MyPlugin] → [Properties]
- Click [Java Build Path] and click [Libraries] tab
- Click [Add External JARs] button





Select Java Build Path

- Select following .jar files
  - C:\Program Files\CellDesigner4.0.1\exec \celldesigner.jar
  - C:\Program Files\CellDesigner4.0.1\lib \sbmlj.jar



Properties for MyPlugin

type filter text

- Info
- Builders
- Java Build Path
- Java Code Style
- Java Compiler
- Javadoc Location
- Project References

Java Build Path

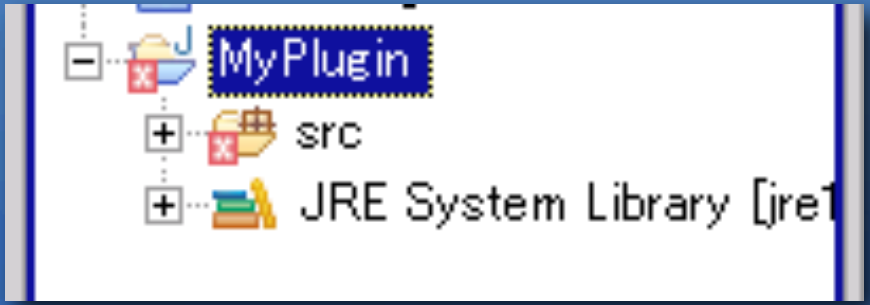
Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

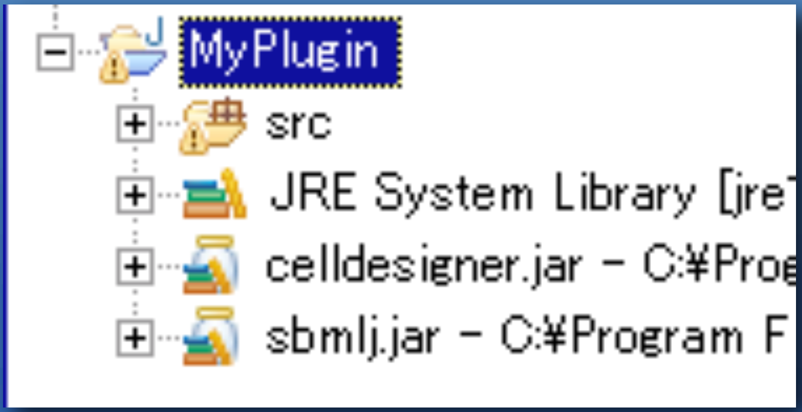
- celldesigner.jar - C:¥Program Files¥CellDesigner4.0
- sbmlj.jar - C:¥Program Files¥CellDesigner4.0alpha2
- JRE System Library [jre1.5.0_06]

Compile

- Imported java source files are automatically compiled and java class files are generated in the "bin" directory of your project directory



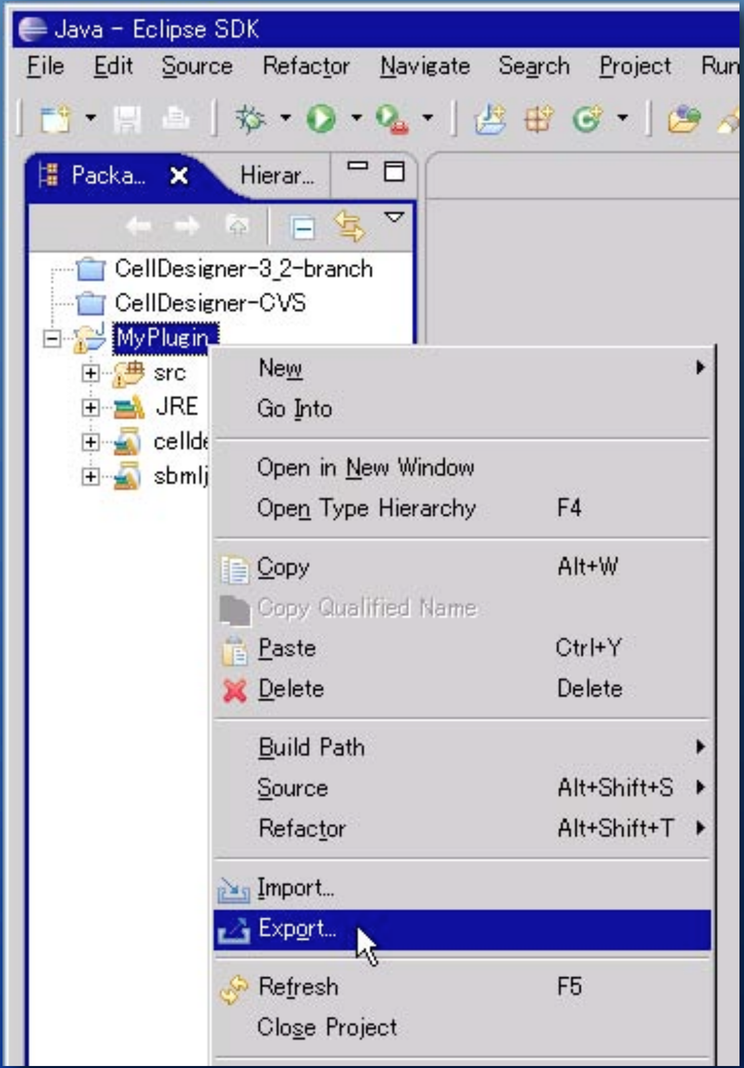
NG

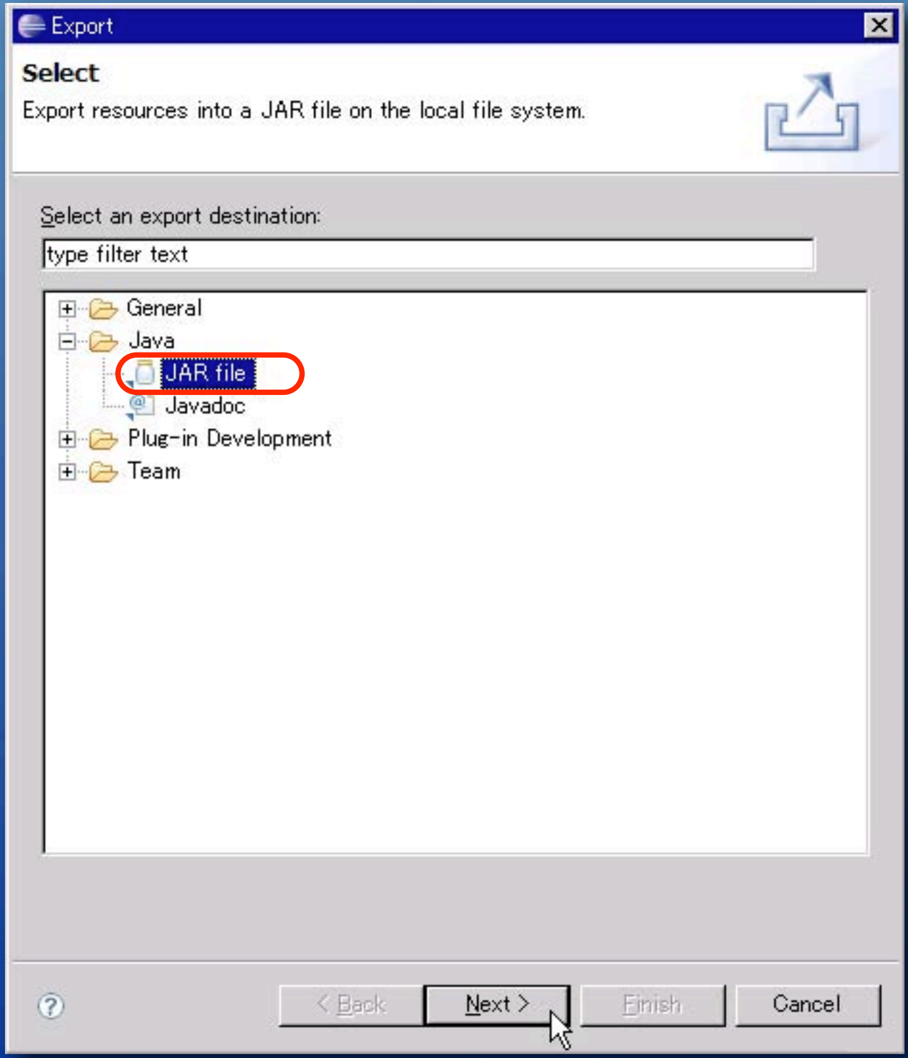


OK

Generate jar files

- Right click [MyPlugin] → [Export]
- Select [JAR file] and click [Next]

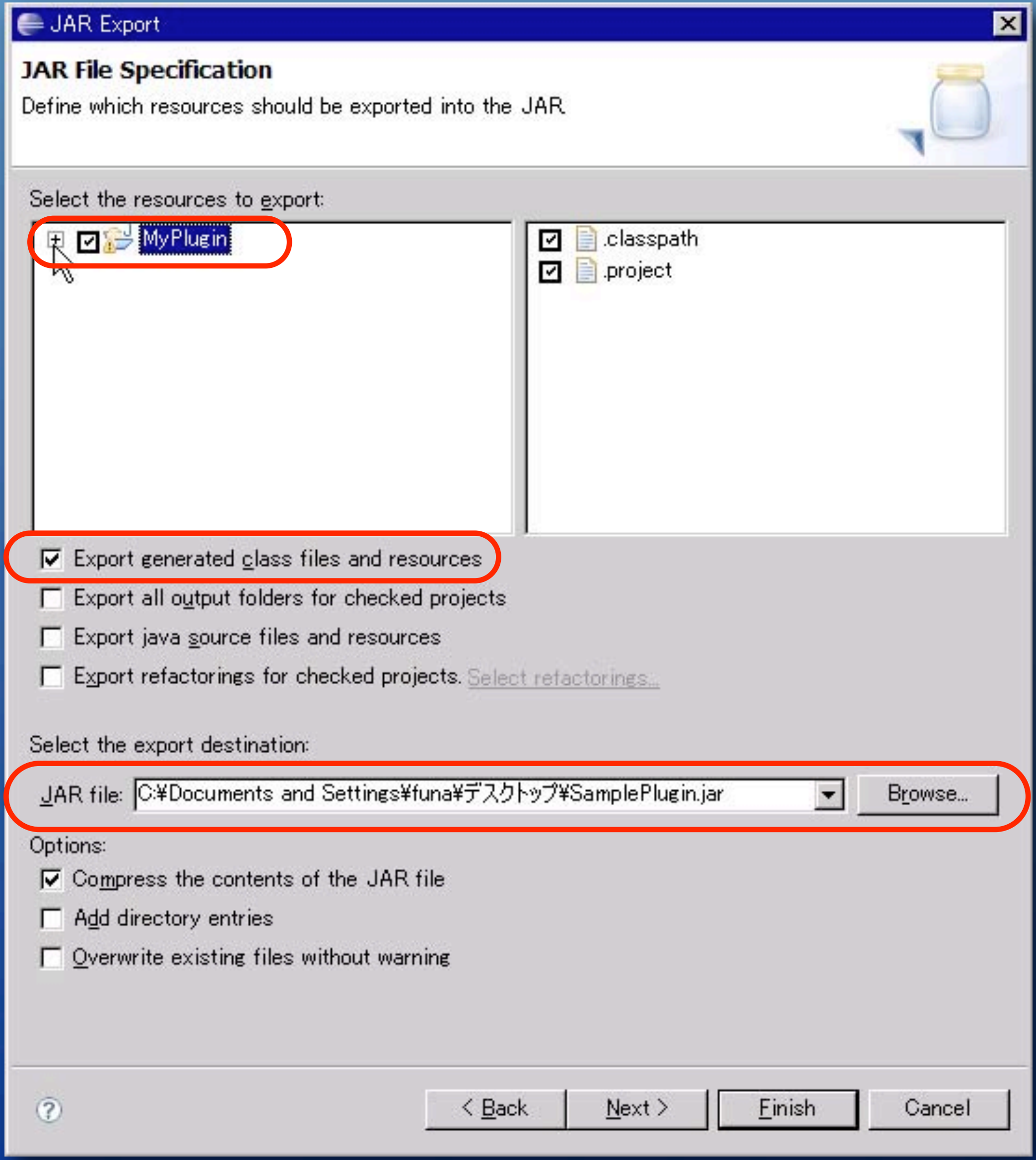




Generate jar files

- Check your project (MyPlugin)
- Select [Export generated class files and resources]
- Specify JAR file

Put jar file to plugin folder


JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

☑ MyPlugin

☑ .classpath
☑ .project

☑ Export generated class files and resources
☐ Export all output folders for checked projects
☐ Export java source files and resources
☐ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: C:¥Documents and Settings¥funa¥デスクトップ¥SamplePlugin.jar

Options:

☑ Compress the contents of the JAR file
☐ Add directory entries
☐ Overwrite existing files without warning

How to implement plugin

- Write your plugin class
  - extend CellDesignerPlugin class
- Write an action class
  - extend PluginAction class
- Create menu and menu item
  - use PluginMenu, PluginMenuItem
- Register PluginMenu to CellDesigner
  - use addCellDesignerPluginMenu()
- Implement some methods to receive events from CellDesigner

1. Write your plugin class

- Your plugin class must extend the CellDesignerPlugin class. CellDesigner will call the constructor of your plugin class to instantiate it.

```
public class SamplePlugin extends CellDesignerPlugin {
   // Constructor
   public SamplePlugin() {

   }
}
```

2. Write action class

- Write an action class which extends the PluginAction class for an action event that would be passed when the plugin menu is selected on CellDesigner.

```
public class SampleAction extends PluginAction {

    public SampleAction(SamplePlugin plugin) {
        // Write your code for constructor
    }

    public void myActionPerformed(ActionEvent e) {
        // Write your code for action event
    }
}
```

SBI The Systems Biology Institute

3. Create menu and item

- Use PluginMenu class and PluginMenuItem class to create menus on CellDesigner. Register the action class to the PluginMenuItem for CellDesigner to invoke the action.

```
public class SamplePlugin extends CellDesignerPlugin {
  // Constructor
  public SamplePlugin() {
    PluginMenu menu = new PluginMenu("Sample");
    SampleAction action = new SampleAction(this);
    PluginMenuItem item = new PluginMenuItem("Sample1",
    action);
    menu.add(item);
    addCellDesignerPluginMenu(menu);
  }
}
```

4. Register PluginMenu

- Use following methods to register PluginMenu to CellDesigner
  - addCellDesignerPluginMenu()
    - Register menu to Plugin menu
  - addSpeciesPopupMenu()
  - addReactionPopupMenu()
  - addCompartmentPopupMenu()
    - Register menu to right-clicked pop-up menu

5. Implement methods

- Implement following methods to receive events from CellDesigner (required).

```
public class SamplePlugin extends CellDesignerPlugin {
  public SamplePlugin() {}                 // Constructor
  public void addPluginMenu() {}    // add PluginMenu

  public void SBaseAdded(PluginSBase sbase) {}
  public void SBaseChanged(PluginSBase sbase) {}
  public void SBaseDeleted(PluginSBase sbase) {}
  public void modelOpened(PluginSBase sbase) {}
  public void modelSelectChanged(PluginSBase sbase) {}
  public void modelClosed(PluginSBase sbase) {}

}
```

Accessible information

- Plugin can get following information
  - Selected model (SBML)
    - PluginModel getSelectedModel()
  - All opened model (SBML)
    - PluginListOf getAllModels()
  - Selected node on model
    - PluginListOf getSelectedAllNode()
  - All nodes on model
    - PluginListOf getAllSpeciesNodes()

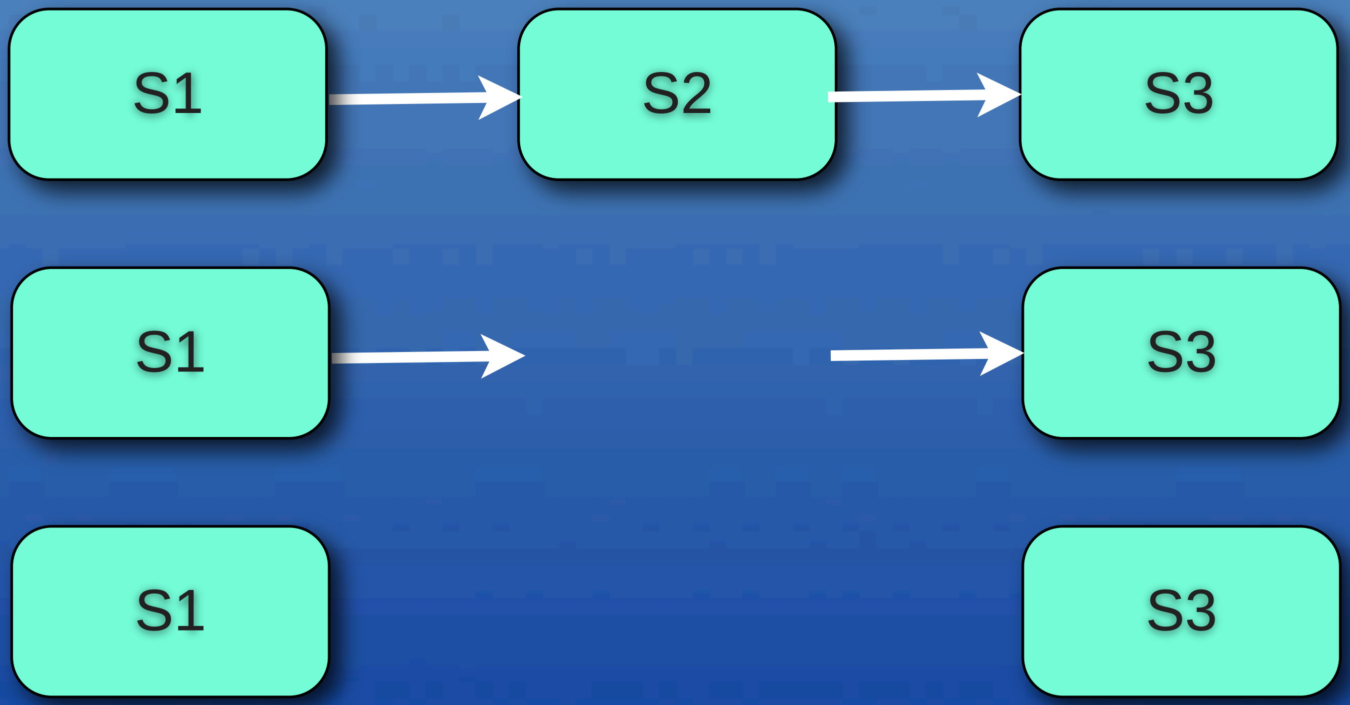
Notification from Plugin

- You can implement functions to add, update and delete PluginSBase in CellDesignerPlugin. The Plugin can notify CellDesigner these changes via CellDesignerPlugin interface.
  - notifySBaseAdded(PluginSBase sbase)
  - notifySBaseChanged(PluginSBase sbase)
  - notifySBaseDeleted(PluginSBase sbase)

Restriction

- Some actions trigger sequential actions. You have to implement the sequential actions in your plugin.

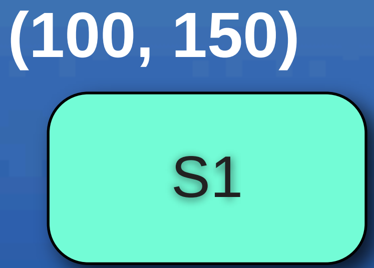
Example: delete species S2



Example code

Get Species properties from CellDesigner

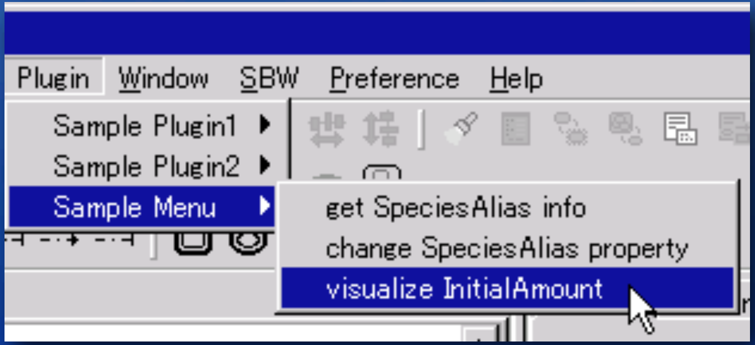
```
private void getSelectedSpecies() {
   PluginListOf lof = plugin.getSelectedSpeciesNode();
   if (lof.size() != 0) {
      // get PluginSpeciesAlias
      PluginSpeciesAlias alias = (PluginSpeciesAlias)lof.get(0);

      // get position
      double x = alias.getX();
      double y = alias.getY();

      // get Species
      PluginSpecies sp  = alias.getSpecies();
      String name = sp.getName();
      String id = sp.getId();
      }
}
```



Example plugin

http://celldesigner.org/~funa/plugintutorial.jar

- Get SpeciesAlias info
  - Print out SpeciesAlias info (for debug)
- Change SpeciesAlias property
  - Change color, size, position of Proteins depend on its name (work with MAPK.xml)
- Visualize InitialAmount
  - Change color of Species when its InitialAmount < 20.0



Acknowledgment

SBML
* SBML community
* Caltech
  Michael Hucka
  Ben Bornstein
  Bruce Shapiro
  Sarah Keating
* Keio Univ.
  Akiya Jouraku

SBGN
* SBGN community
* Nicolas Le Novere (EBI)
* Michael Hucka (Caltech)
* Hiroaki Kitano (SBI)
* Yukiko Matsuoka (SBI)

CellDesigner
* SBI
  Yukiko Matsuoka
  Hiroaki Kitano
* Keio Univ.
  Akiya Jouraku
* MKI
  Norihiro Kikuchi

SBML ODE Solver (Univ. of Vienna)
Rainer Machne
Christoph Flamm

SBW (Univ. of Washington)
Frank Bergmann
Herbert Sauro

COPASI (Univ. of Heidelberg)
Ralph Gauges
Sven Sahle
Ursula Kummer