



BioUML

extensible workbench
for systems biology

Desired SBML extensions

Fedor Kolpakov

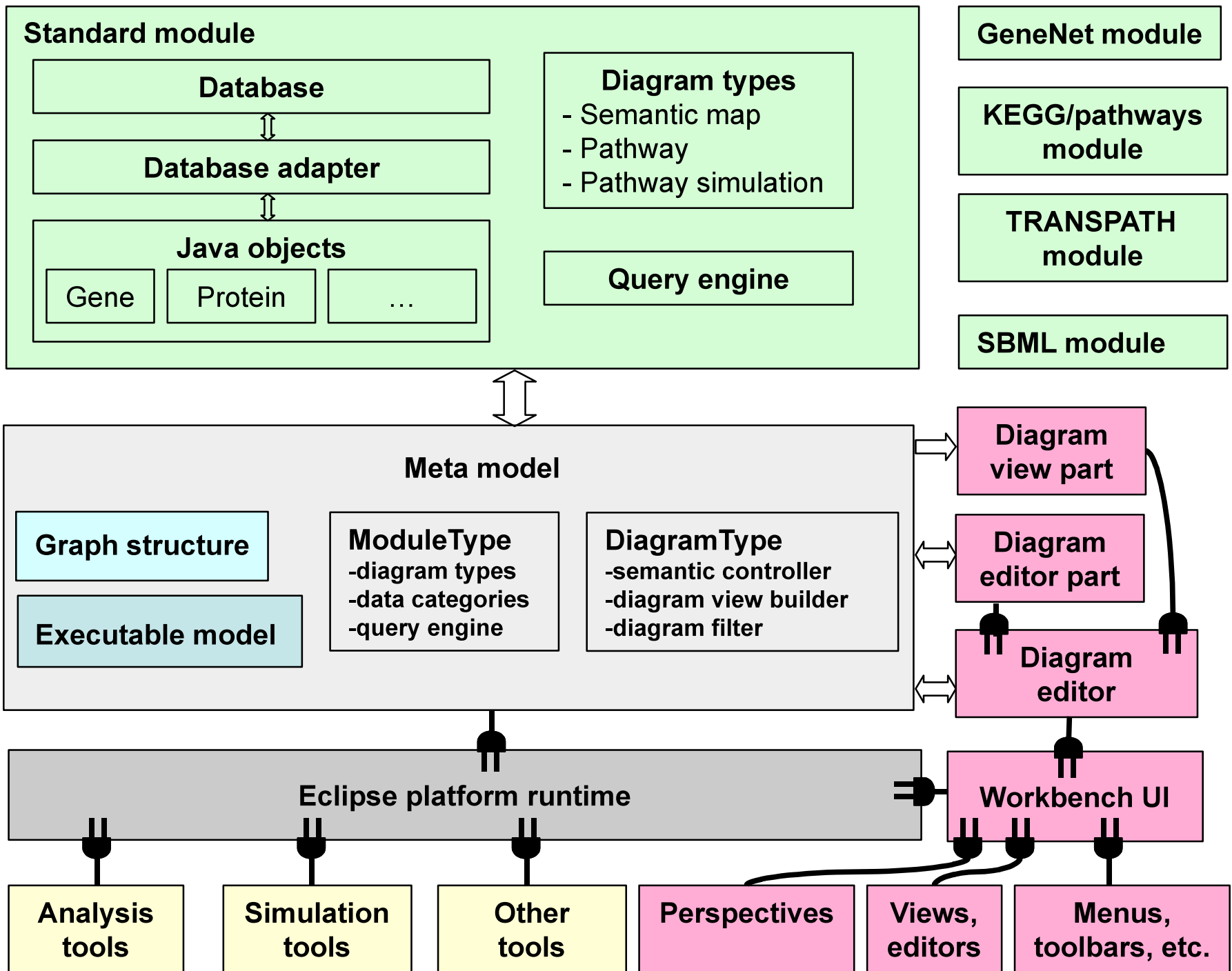
Institute of Systems Biology
(spin-off of DevelopmentOnTheEdge.com)

Laboratory of Bioinformatics,
Design Technological Institute of Digital Techniques

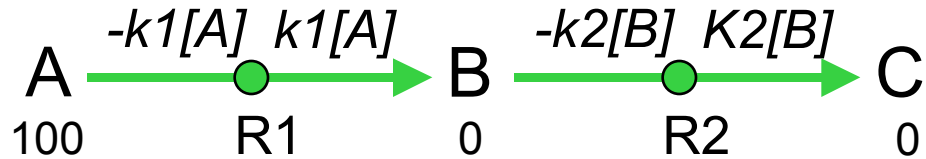
Novosibirsk, Russia

Main BioUML concepts and ideas

- Plug-in based architecture (Eclipse platform runtime from IBM company).
- Visual modeling
 - Meta model – problem domain neutral level of abstraction that describes system as compartmentalized graph
 - Diagram type concept – formally defines graphical notation and provides its incorporation into BioUML workbench.
 - Automated code generation for model simulation.
- Module concept - allows developer to incorporate databases on biological pathways into BioUML workbench taking into account database peculiarities.



Example:
system from two chemical reactions



$k1$ - reaction rate for R1

$k2$ - reaction rate for R2

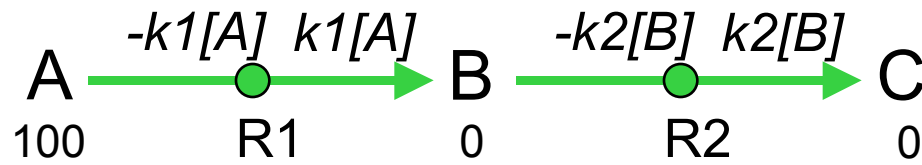
Corresponding mathematical model:

$$\frac{dA}{dt} = -k1[A]$$

$$\frac{dB}{dt} = k1[A] - k2[B]$$

$$\frac{dC}{dt} = k2[B]$$

Meta-model: example of formal description of system from two chemical reactions



ID	A
CC	..
...	
//	

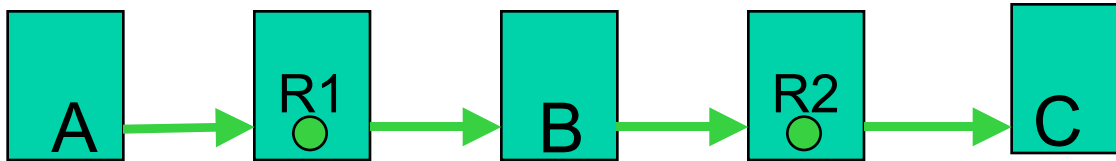
ID	R1
	A->B
...	
//	

ID	B
CC	..
...	
//	

ID	R2
	B->C
...	
//	

ID	C
CC	..
...	
//	

Description of system components in the database



System structure is described as a graph

100	$-k_1[A]$	$k_1[A]$	0	$-k_2[B]$	$k_2[B]$	0
-----	-----------	----------	---	-----------	----------	---

Mathematical model of the system

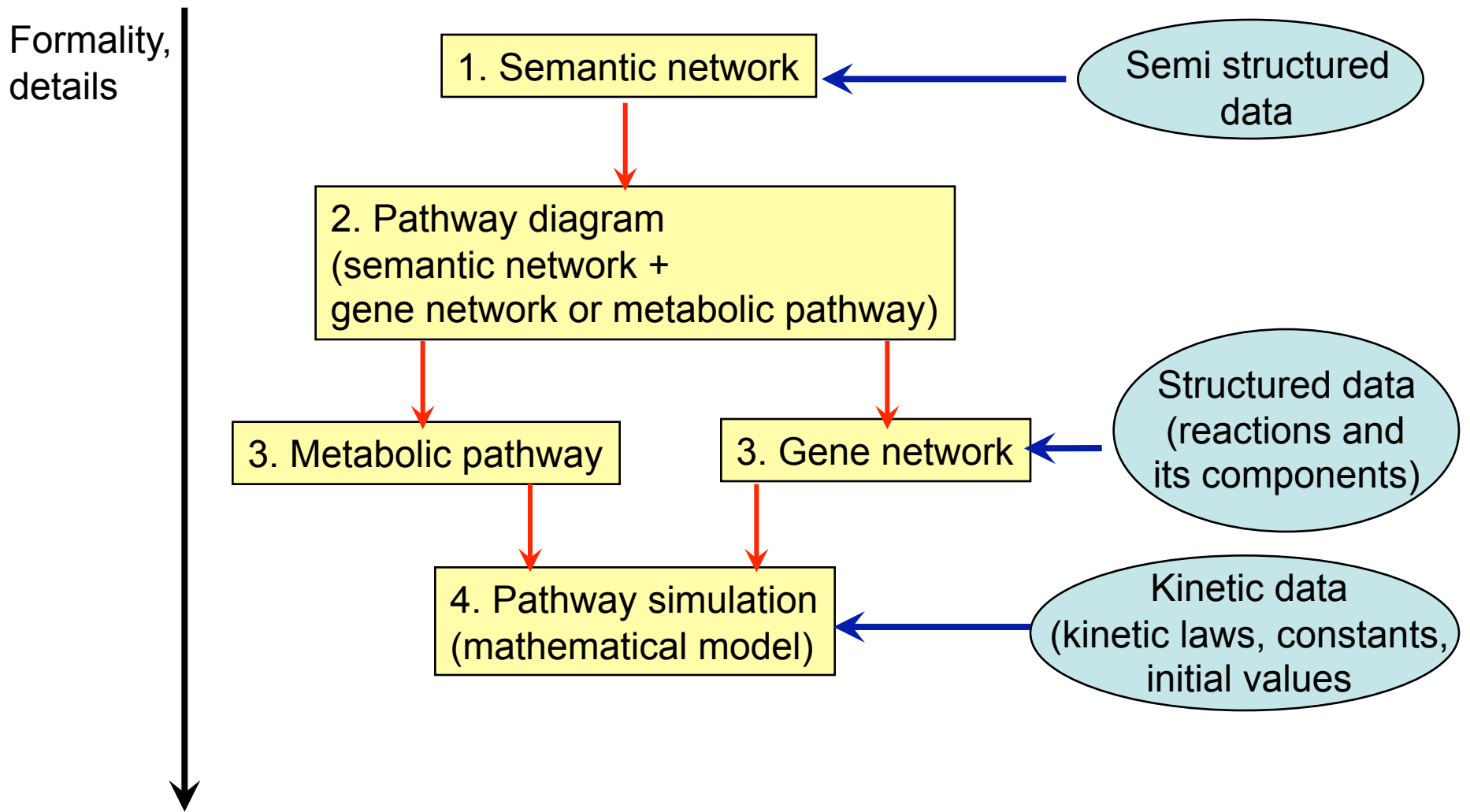
Suggested approach can be applied for modeling biological systems using:

- ✓ – Systems of ordinary differential equations
- ✓ – Systems of algebra-differential equations
- ✓ – State and transition diagrams
- ✓ – Hybrid models
 - Boolean and logical networks
 - Petri nets
 - Markov chains
 - Stochastic models
 - Cellular automates
 - **1D PDE models (blood flow)**
 - ...

Current limitations

- Spatial models
- PDE
- ...

Reconstruction and formal description of biological systems using different diagram types



Biopath (BMOND)

a new approach for formal description and simulation of biological systems



*Fedor Kolpakov^{1,2}, Ruslan Sharipov^{3,1,2}, Ekaterina Kalashnikova³, and Elena Cheremushkina^{1,2}

¹Institute of Systems Biology, Novosibirsk, Russia;
²Design Technological Institute of Digital Techniques SB RAS, Novosibirsk, Russia;
³Institute of Cytology and Genetics SB RAS, Novosibirsk, Russia
 *Contacts: fedor@biouml.org

Summary

Reconstruction of complex biological systems from a huge amount of experimental data requires a formal approach for description and simulation of biological pathways on different logical levels. Biopath/BMOND (Biological Model and Diagrams) database was developed for formalized description and simulation of biological systems using two new technologies: BioUML and BezaBioKer. All Biopath/BMOND data can be divided into 4 major blocks: regulation of eukaryotic cell cycle and cancer, NF- κ B pathway and inflammation, nuclear regulation of gene expression, and human arterial hypertension. Biopath/BMOND contains links to related biological databases and literature references. It is integrated with Cyclonet database (<http://cyclonet.biouml.org>) – a specialized database on cell cycle regulation.

Goals

- >to describe formally structure and functioning of complex biological systems and processes on different logical levels as a set of complementary diagrams;
- >to provide formal description of main properties of biological components (genes, proteins, substances, chemical reactions) that are components of these processes;
- >to simulate behavior of these processes or their parts to validate that we understand main principles of their functioning;
- >to provide user interface for Biopath/BMOND access and editing through the Internet.

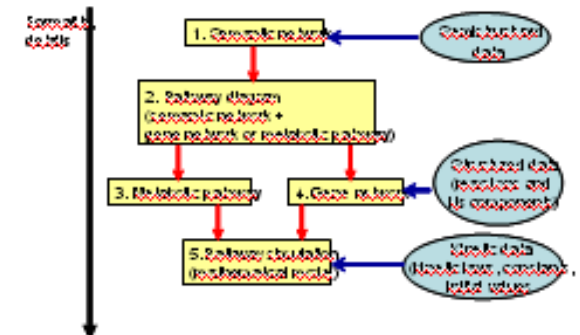
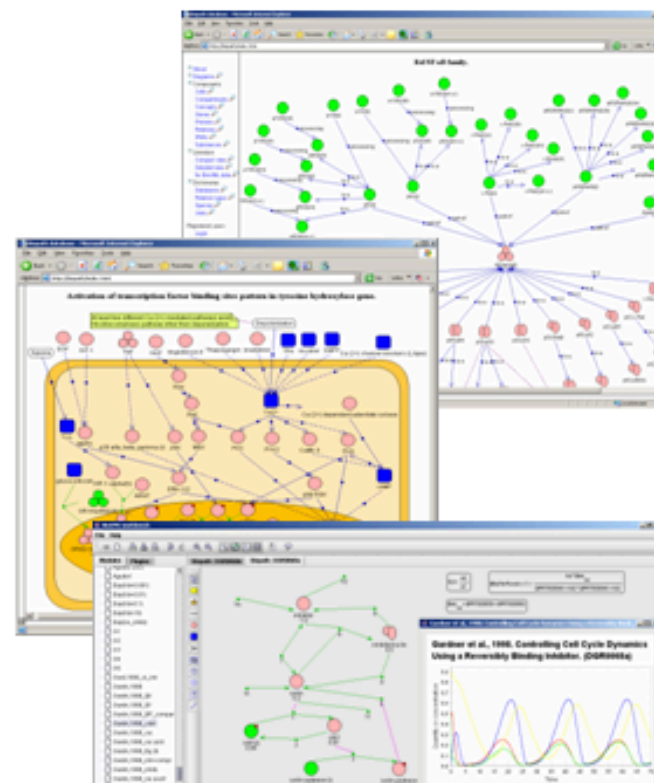
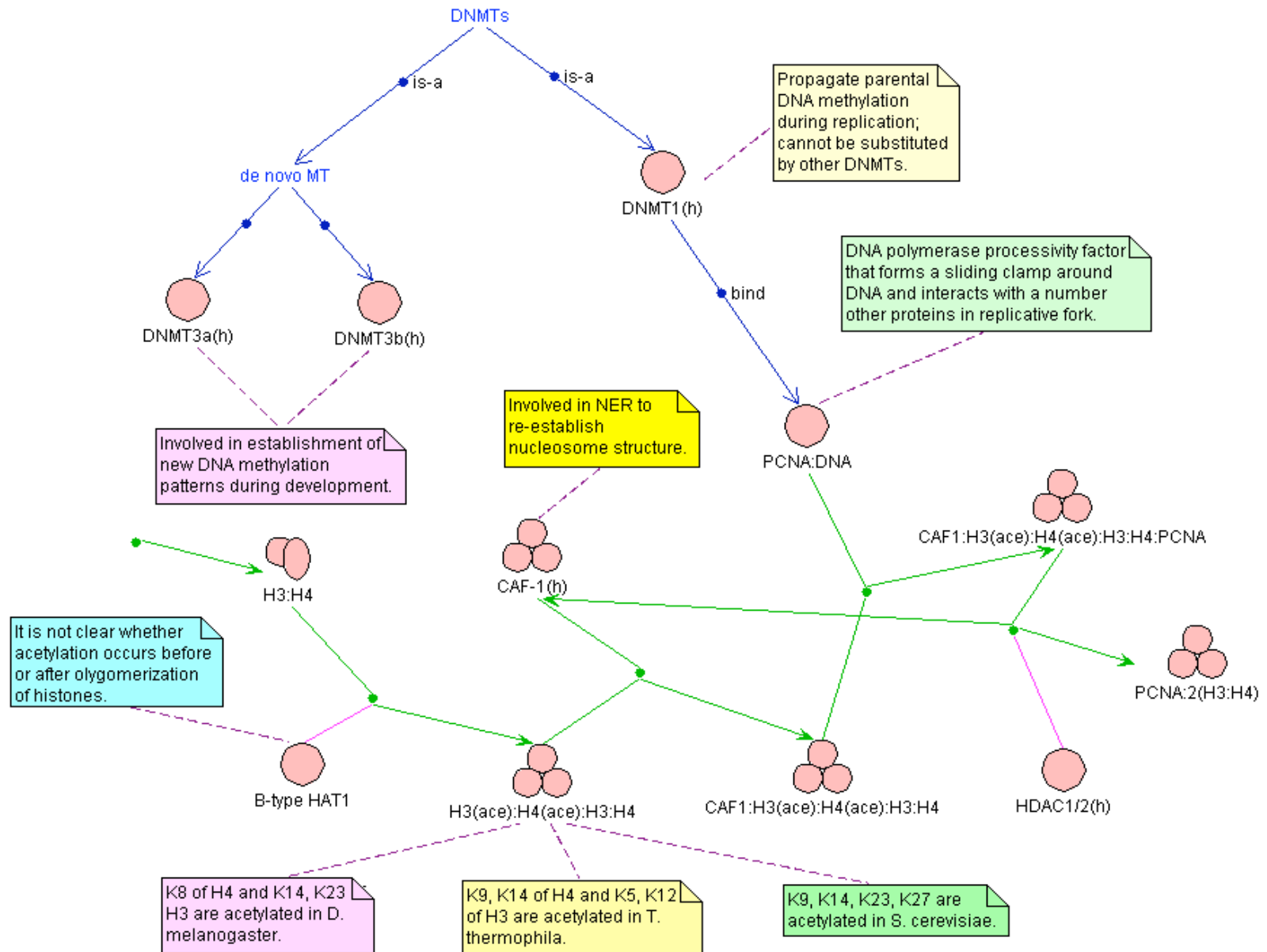


Figure 1. Reconstruction and formal description of biological systems using different diagram types in Biopath/BMOND.

Table 1. Graphical notations for different diagram types ("*" - is used, "-" - is not used in this diagram type)

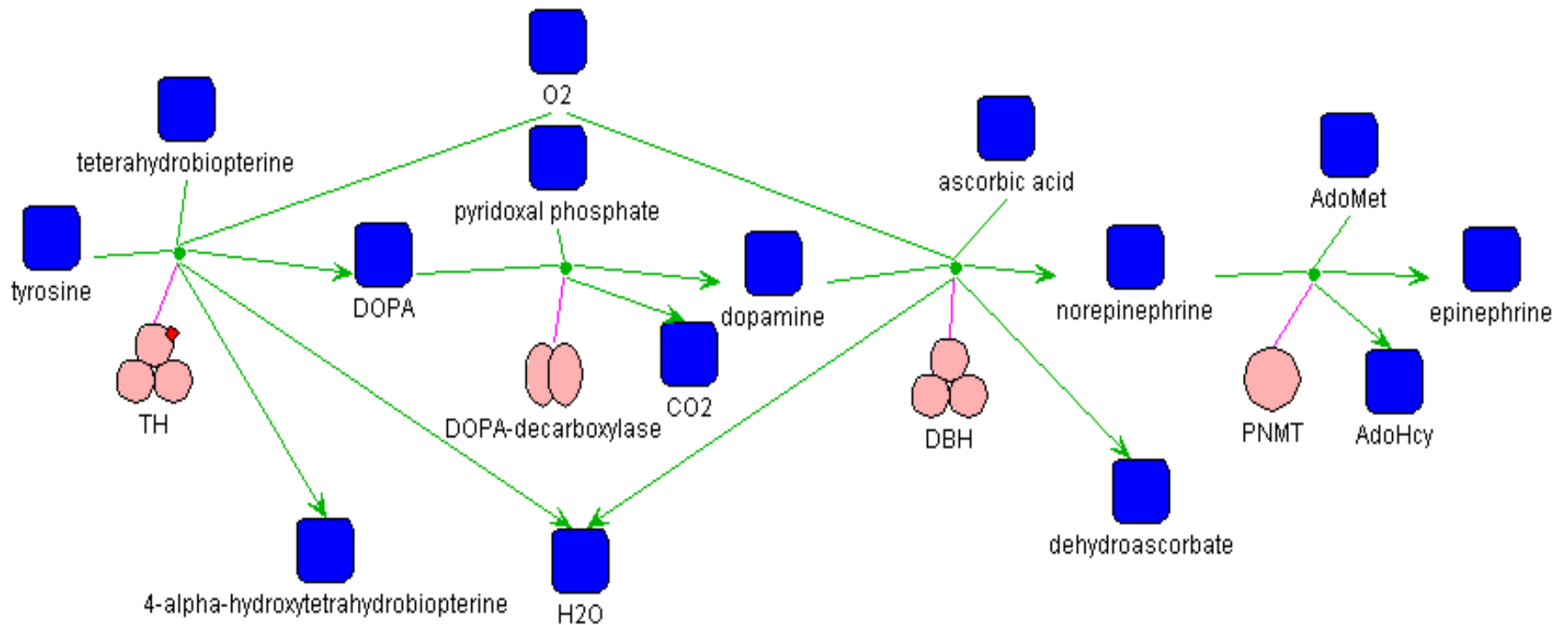
Component type	Description	Diagram types				
		Genetic network	Pathway diagram	Metabolic pathway	Gene network	Pathway simulation
Proteins						
	Active substance					
	Active substance					
	Active substance					
	Active substance	*	*	*	*	*
	Active substance					
	Transition substance					
Other components						

Function of human DNA methyltransferases (pathway diagram)

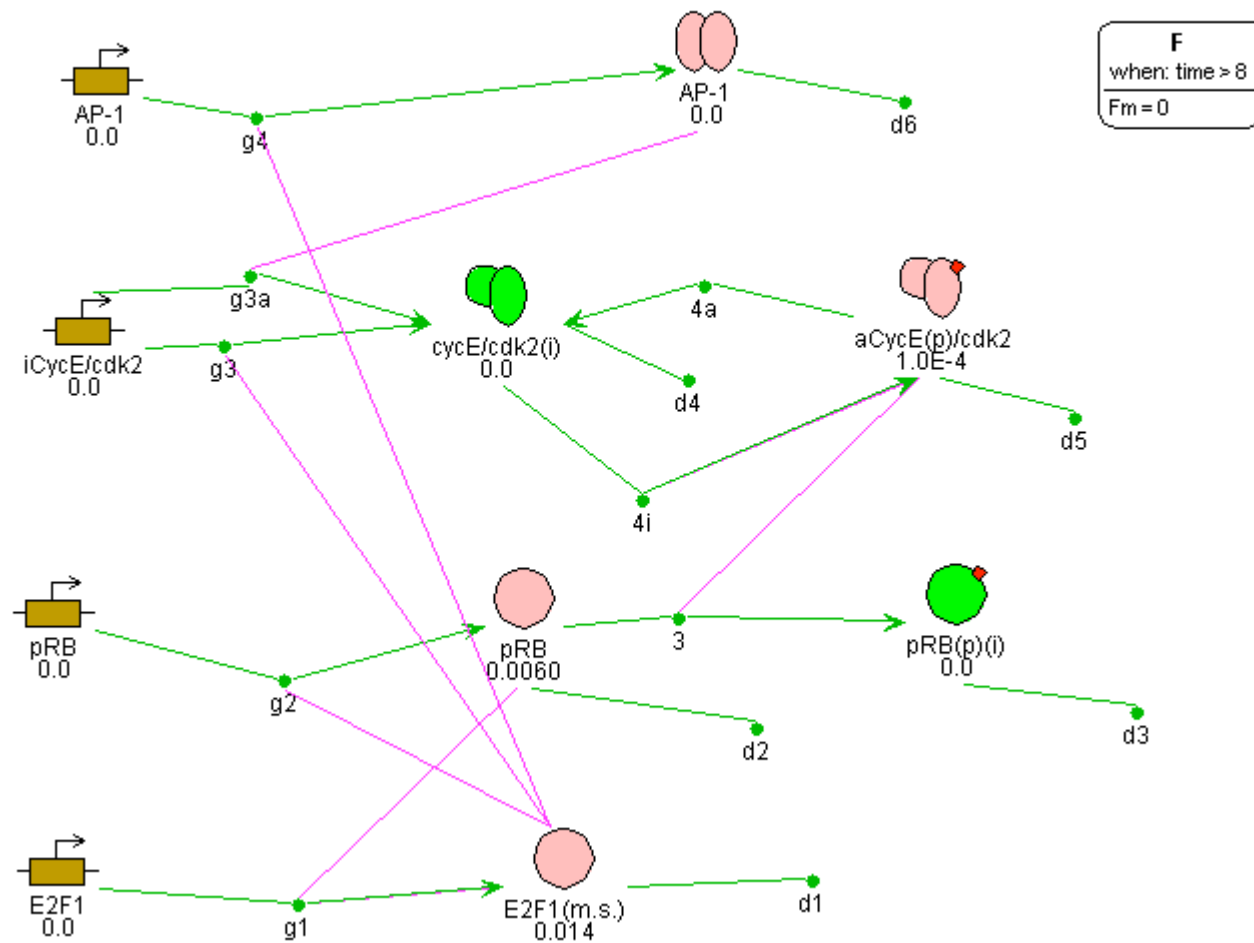


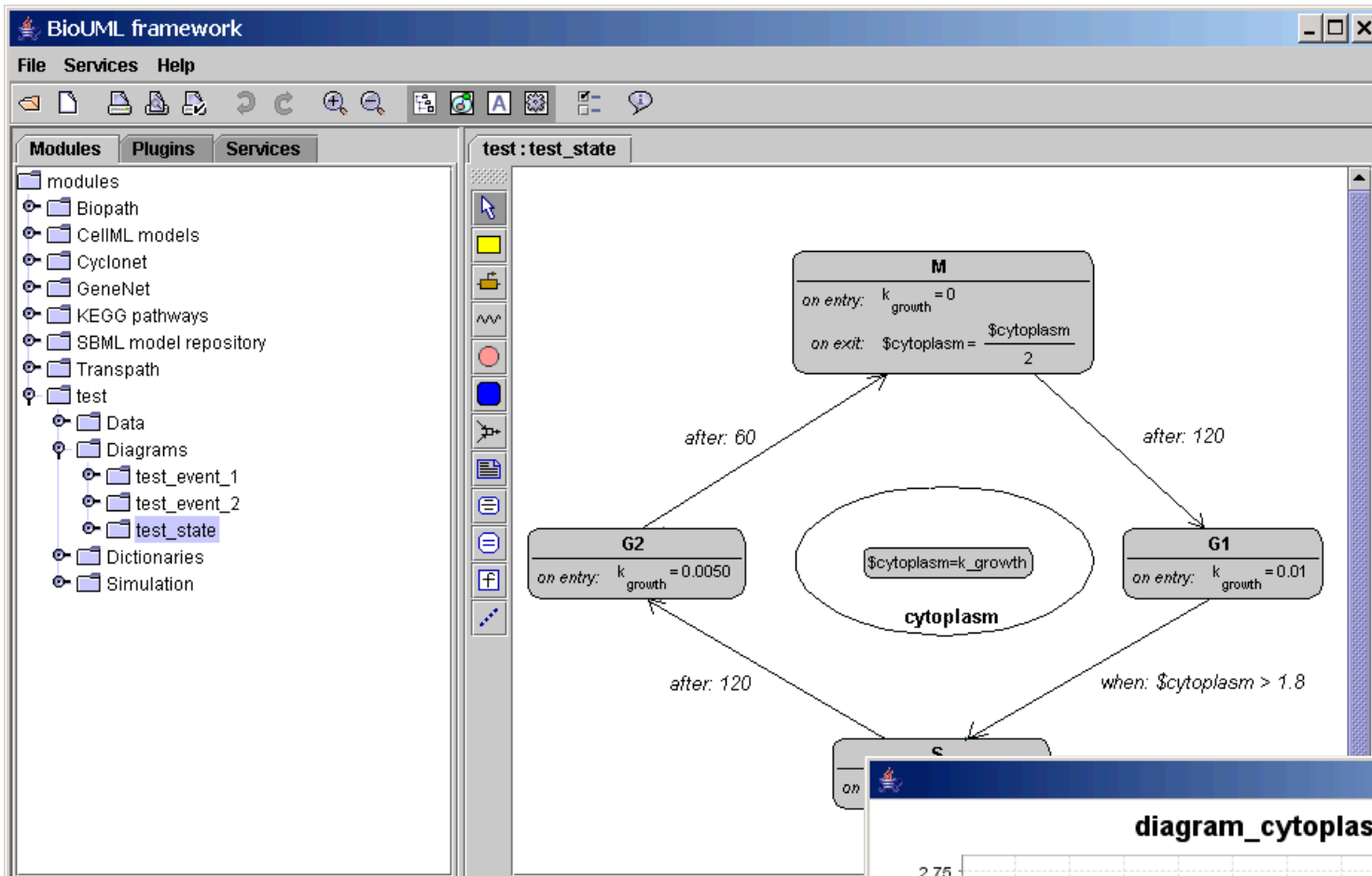
The biosynthesis of catecholamines

(metabolic pathway)



Cell cycle model of mammalian G1/S transition control with E2F feedback loops (pathway simulation diagram)





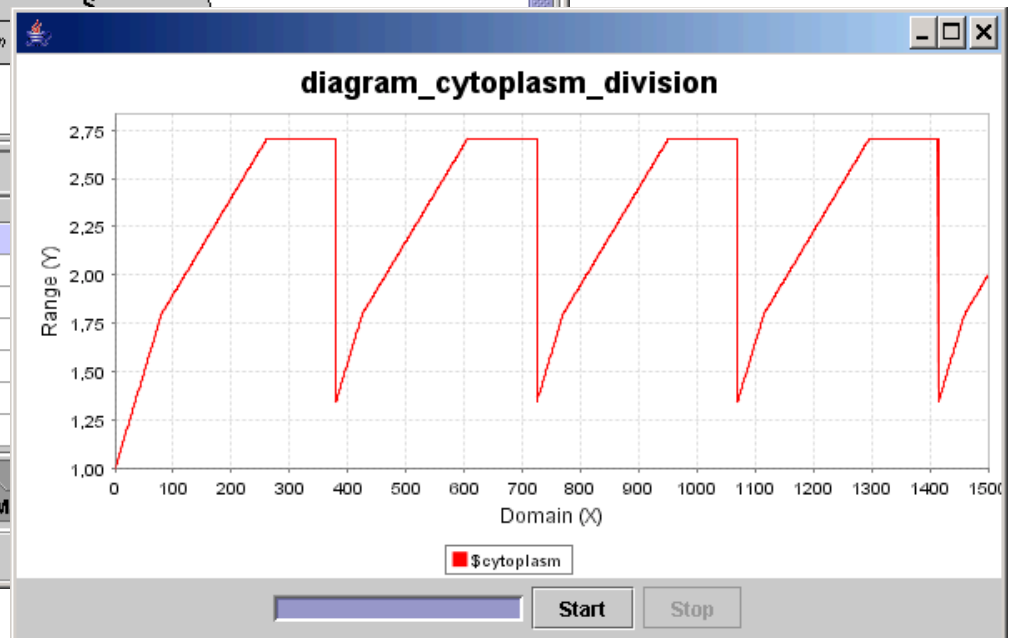
Property	Value
Node	
Title	growth rule
Comment	
Size	104.0 28.0
Role	
Equation	k_growth
Comment	

Property	Value
Java simulation engine	
ODE solver name	
Output dir	
Initial time	
Completion time	
Time increment	
Needed Plot Simulation	

View Edit Editors

Description Legend

Parameters Variables JAVA M





The SBML Wiki

Semantic Test Suite

You can edit these Wiki pages from within your browser. [Click here to learn how.](#)

The semantic test suite is set of test models with given time series behaviour data. The aims of the suite are described in more detail in the [Test Suite Introduction](#).

The test suite is downloadable with documentation and test automation scripts in 2 versions one for [windows cygwin](#) and the other for [unix](#). Documentation on the general structure of the test suite, file formats and automation scripts is available separately [here](#)

The suite is under development. A test is either:

- complete - it has times series behaviour data and supporting documentation
- under development - the model exists but no times series data is available
- planned - a model to cover more of the specification is required

The suite is currently divided into the following categories:

[Basic Reaction Semantic Tests](#) 19 complete tests

[Complex Reaction Semantic Tests](#) 6 complete tests

[Parameter Scope Semantic Tests](#) 4 complete tests

[Discontinuity and Delays Semantic Tests](#) 14 complete tests + 2 planned tests

[Compartment and Transport Reaction Semantic Tests](#) 3 tests under development

[Stoichiometry Semantic Tests](#) 2 completed Tests

[Basic Rule Semantic Tests](#) 16 complete tests

[Algebraic Rule Semantic Tests](#) 2 complete tests

Results of SBML semantic tests

Simulation engine	Details	Tests	Successful	Failed	Errors	Needs tuning	CSV error	Result differs	Success rate	Time (s)
<i>SBML level 1</i>										
Java	details	53	49	0	0	0	4	0	100.00 %	21
MATLAB	details	53	49	0	0	0	4	0	100.00 %	444
<i>SBML level 2</i>										
Java	details	141	120	0	0	0	21	0	100.00 %	41
MATLAB	details	141	120	0	0	0	21	0	100.00 %	1598

Results for Java simulation engine, SBML level 2 semantic tests

Summary statistics - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: http://biouml.org/sbml_tests/SemanticTests-Java-l2.html Переход Ссылки >>

Name	Tests (s)	Successful	Failed	Errors	Needs tuning	CSV error	Result differs	Time (s)
BasicReactions	19	19	0	0	0	0	0	4
ParameterNamespace	4	4	0	0	0	0	0	1
ComplexReactions	6	6	0	0	0	0	0	3
MathML	43	43	0	0	0	0	0	12
BasicRules	16	16	0	0	0	0	0	4
Stoichiometry	2	2	0	0	0	0	0	0
Discontinuity, TimeAndDelays	15	14	0	0	0	1	0	4
AlgebraicRules	2	2	0	0	0	0	0	0
Functions	6	5	0	0	0	1	0	1
Events	5	4	0	0	0	1	0	1
Compartments	3	3	0	0	0	0	0	2
rulesForParametersAndCompartments	20	2	0	0	0	18	0	5

Готово Интернет

Test: functions/algebraic-rule/functions-algebraic-rule.test - ok

```

test           : functions/algebraic-rule/functions-algebraic-rule.test
model          : xml, diagram, description
simulation engine : Java
generated code  : functions\_algebraic\_rule\_l2.java
results        : table with results
plots          : normal, log
status         : ok
simulation time : 0.469 s.

```

Parameters:

```

TIME 40
STEPS 100
SPECIES X0 X1 T S1 S2
ATOL 1e-10
RTOL 1e-7
ZERO 1e-13
URL Function_Definition_used_in_Algebraic_Rule_Semantic_Test
REM tests the use of a function in an algebraic rule

```

SBML model

Diagram



Function Definitions

ID	NAME	FUNCTION DEFINITION
f	...	Function[x, y, x + y]

Model description

Compartments

ID	NAME	DIMENSION	SIZE	UNITS	DERIVED UNITS	OUTSIDE	CONSTANT
----	------	-----------	------	-------	---------------	---------	----------

Simulation results **(legend)**

time	X0	X1	T	S1	S2
0.0	1.0 1.0	-1.965279617781012E-38 0.0	1.0349855076499675E-27 0.0	0.0 -2.3592239273284576E-16	0.0 -5.898059818321144E-16
0.8	0.923116346390738 0.9231163463556705	0.003244942591247183 0.003244942451956452	0.07363871101801513 0.07363871119237303	0.021039631719432895 0.02103963176924928	0.052599079298582235 0.052599079423123196
1.6	0.8521437889687263 0.8521437889127818	0.012289348132997127 0.012289347901178058	0.13556686289827696 0.13556686318604017	0.03873338939950771 0.03873338948172593	0.09683347349876927 0.09683347370431483
2.4000000000000004	0.7866278610677425 0.7866278610050365	0.026190450755669 0.026190450491197725	0.18718168817658895 0.18718168850376582	0.053480482336168284 0.053480482429647445	0.13370120584042067 0.13370120607411862
3.2	0.7261490370739397 0.7261490371125509	0.0441187835308001 0.04411878375374514	0.2297321793952605 0.22973217913370397	0.065637765541503 0.0656377654667729	0.16409441385375745 0.16409441366693228
4.0	0.6703200460353722 0.6703200459850173	0.06534611488889602 0.0653461147583564	0.264333839075732 0.2643338392566264	0.07552395402163768 0.07552395407332085	0.1888098850540942 0.18880988518330213
4.8000000000000001	0.6187833918057666 0.6187833915689761	0.08923458793830828 0.08923458707454503	0.29198202025592535 0.2919820213564789	0.08342343435883581 0.08342343467327981	0.20855858589708953 0.20855858668319951
5.6000000000000005	0.5712090638486887 0.5712090636578717	0.11522694757633319 0.11522694698970624	0.31356398857497847 0.313563989352422	0.08958971102142238 0.08958971124354924	0.22397427755355595 0.2239742781088731
6.4	0.5272924240434489 0.5272924241884959	0.1428377509945876 0.1428377518526681	0.3298698249619641 0.32986982395883596	0.09424852141770403 0.09424852113109607	0.2356213035442601 0.23562130282774016
7.2	0.48675225596108657 0.48675225627220386	0.17164546691286223 0.17164546847960835	0.3416022771260515 0.3416022752481878	0.09760065060744327 0.09760065007091077	0.24400162651860804 0.24400162517727692
8.0	0.4493289641191446 0.449328964245015	0.20128537771946772 0.20128537851101796	0.3493856581613881 0.349385657243967	0.09982447376039656 0.09982447349827626	0.24956118440099145 0.24956118374569067
8.8	0.41478291167675835 0.4147829106072333	0.23144320671469854 0.23144320307127406	0.35377388160854345 0.3537738863214926	0.10107825188815521 0.10107825323471216	0.2526956297203876 0.2526956330867804

BioUML modules

BioUML standard module

Databases

- Biopath/BMOND (<http://biopath.biouml.org>)
- GeneNet (<http://wwwmgs.bionet.nsc.ru>)
- KEGG/Ligand (<http://www.kegg.com>)
- TRANSPATH (<http://www.biobase.de>)

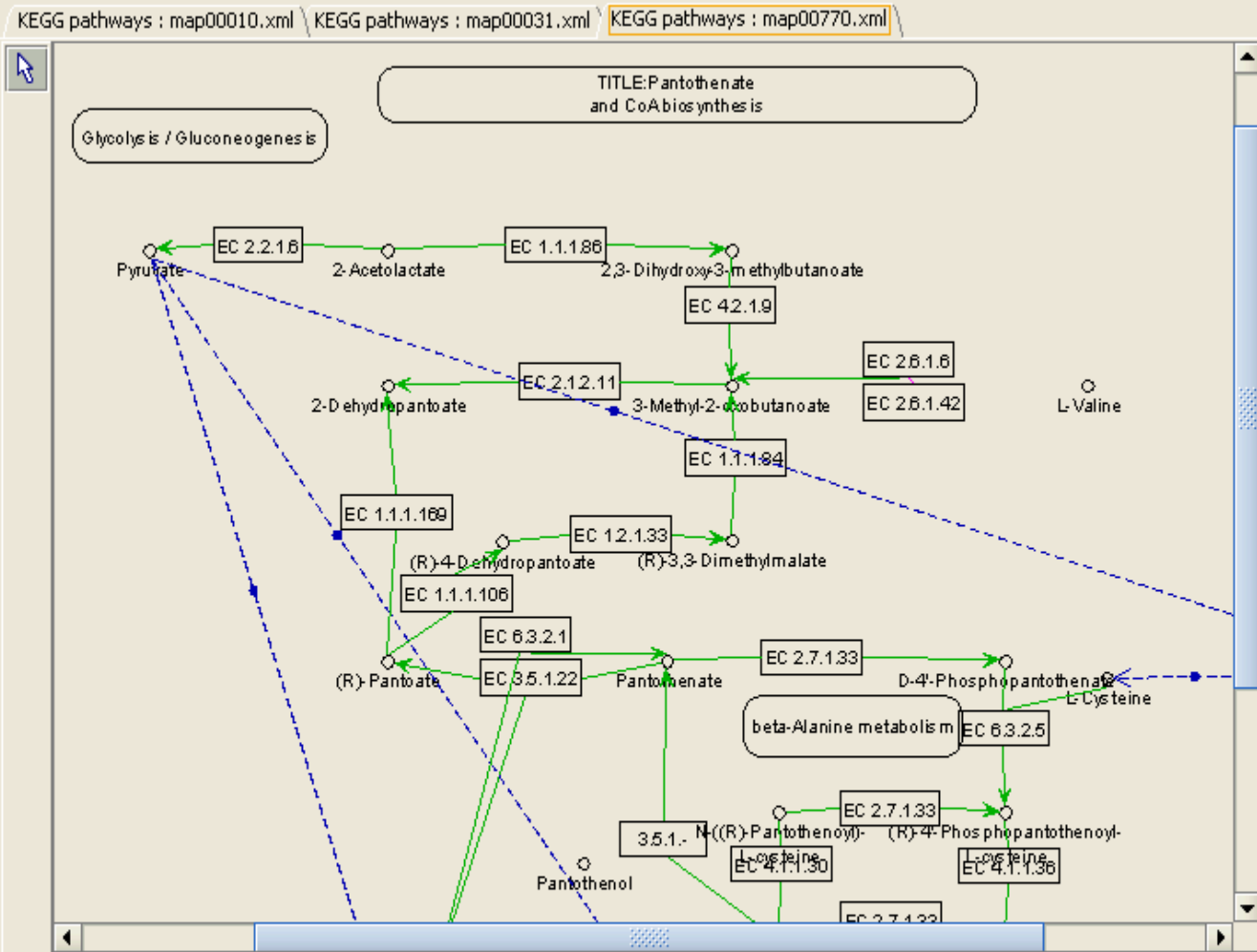
Formats

- SBML – Systems Biology Markup Language, level 1, 2 ([http:// www.sbml.org](http://www.sbml.org))
- CellML – Cell Markup Language (<http://www.cellml.org>)
- BioPax – Biological Pathways Exchange (<http://www.biopax.org>)
- GXL - Graph eXchange Language (<http://www.gupro.de/GXL>)
- GinML – extension of GXL for description regulatory networks (<http://gin.univ-mrs.fr/GINsim>)
- BioNetGen - mathematical models of biological systems from user-specified rules for biomolecular interactions (<http://cellsignaling.lanl.gov/bionetgen>)

KEGG pathway

Modules Plugins

- modules
 - BioNetGen
 - Biopath (local)
 - Cell cycle
 - CellML models
 - GinSim repository
 - KEGG pathways
 - Data
 - Diagrams
 - map00010.xml
 - map00031.xml
 - map00770.xml
 - map02030bsu.xml
 - ec:2.1.1.80: protein-glutamate O-me
 - ec:2.7.3.-
 - ec:3.1.1.61: protein-glutamate meth
 - ko:AER
 - ko:CHEC
 - ko:CHED
 - ko:CHEV ko:HD.CHEV.R
 - ko:CHEW
 - ko:CHEX
 - ko:CHEY ko:HD.CHEY.R
 - ko:CHEZ
 - ko:FLIG
 - ko:FLIM
 - ko:FLIN



Navigation and editing tools:

- Navigation: Left and right arrows.
- Text formatting: Bold (b), Italic (i), Underline (u).
- View options: Normal, Html, Preview.
- Editors: Description, ApplicationLog, JavaScript.

SBML model

BioUML workbench

File Help

Modules \ Plugins \

- modules
- BioNetGen
- Biopath (local)
- Cell cycle
- CellML models
- GinSim repository
- KEGG pathways
- MMD
- SBML model repository
 - Diagrams
 - 100Yeast.xml
 - Agrobacterium-tumefaciens-C58.xml
 - CellCycle-1991Gol.xml
 - CellCycle-1991Tys-2.xml
 - CellCycle-1991Tys.xml**
 - CellCycle-1997Nov.xml
 - CellCycle-1998Gar.xml
 - CircClock-1999Lel_periodic.xml
 - CircClock-2001Ued.xml
 - CircClock-2002Vil.xml
 - ElectroPhys-1952Hod.xml
 - Genetic-2000Elo.xml
 - Genetic-2003Mar.xml
 - Heliobacter-pylori-26685.xml
 - MAPKcasc-2000Kho.xml
 - MAPKcasc-2000Lev-2.xml

SBML model repository : CellCycle-1991Tys.xml

```
graph TD; C2[C2 0.0] <-->|Reaction3| CP[CP 1.0]; CP --> EmptySet[EmptySet 0.0]; EmptySet --> YP[YP 0.0]; YP -->|Reaction8| EmptySet; YP -->|Reaction8| M[M 0.0]; M -->|Reaction1| CP; M -->|Reaction4| Y[Y 0.0]; Y -->|Reaction1| CP; Y -->|Reaction4| M; pM[pM 0.3] -->|Reaction5| M; M -->|Reaction5| pM;
```

cell

Tyson1991CellModel (CellCycle-1991Tys.xml)

Tyson1991CellModel (CellCycle-1991Tys.xml)

Time	\$cell.M	\$cell.Y	\$cell.CP	\$cell.EmptySet	\$cell.C2	\$cell.pM	\$cell.YP
0	0.00	0.00	1.00	0.00	0.00	0.30	0.00
1	0.15	0.10	1.10	0.00	0.00	0.15	0.05
2	0.10	0.05	1.15	0.00	0.00	0.10	0.05
3	0.05	0.02	1.18	0.00	0.00	0.05	0.05
4	0.02	0.01	1.20	0.00	0.00	0.02	0.05
5	0.01	0.00	1.20	0.00	0.00	0.01	0.05
6	0.00	0.00	1.20	0.00	0.00	0.00	0.05
7	0.00	0.00	1.20	0.00	0.00	0.00	0.05
8	0.00	0.00	1.20	0.00	0.00	0.00	0.05
9	0.00	0.00	1.20	0.00	0.00	0.00	0.05
10	0.00	0.00	1.20	0.00	0.00	0.00	0.05

Quantity or concentration

Time

Legend: \$cell.M (red), \$cell.Y (blue), \$cell.CP (green), \$cell.EmptySet (yellow), \$cell.C2 (orange), \$cell.pM (magenta), \$cell.YP (cyan)

Stop

Diagram

Title: Tyson1991CellModel

View Edit Editors

Property

- Java simulation engine
- Output dir: ./out

Description ApplicationLog Parameters Variables

Пуск

10:29



Motivation

For our routine work we need to extend SBML format to support:

- Broader range of biological models, including physiological models
- Models composition
- Multi-scale composite models (for example, model of bacterial chemotaxis)
- Tight integration with experimental data:
 - model variation – to reproduce experimental condition we need to modify model structure (for example for mutations), remove or add new events or reactions (for example to simulate experimental conditions)
 - parameters set, fitted (optimized) parameters
 - experimental data and observation – we need some format to store these data
- Storage of simulation results



Required SBML extensions

- Experiment
 - Experiment condition
 - Model variation
 - Parameter set
 - Observed values
 - Simulated values
 - Optimized parameter set
- Model composition
- Multi-scale model composition
- Graphic notation
- Layout information



Required SBML extensions

- Experiment
 - Experiment condition
 - Model variation
 - **Parameter set**
 - Observed values
 - Simulated values
 - **Optimized parameter set**
- **Model composition**
- Multi-scale model composition
- **Graphic notation**
- **Layout information**



Suggestion: agent based approach for multi-scale model composition

- Agent-based computational approaches have a natural modular architecture, which reflects the modular organization of biological systems.
- Bacterial chemotaxis model was simulated by Thierry Emonet and other using agent based approach.
- Now we try to apply agent based approach for simulation arterial hypertension.



AgentCell : A Multi-scale Agent-based Platform for Bacterial Chemotaxis

Thierry Emonet^{1}, Charles M. Macal², Michael J. North², Charles E. Wickersham¹, Philippe Cluzel¹*

¹ The Institute for Biophysical Dynamics and the James Franck Institute, The University of Chicago, 5640 S. Ellis Av., Chicago, IL 60637.

² Center for Complex Adaptive Agent Systems Simulation, Decision and Information Sciences Division, Argonne National Laboratory, 9700 S. Cass Ave., Argonne IL 60439.

ABSTRACT

Motivation: As experimental approaches in biology shift from molecular descriptions to systems analysis, it is necessary to develop a computational framework able to describe simultaneously cellular processes throughout various biological scales. Our goal is to relate intracellular molecular events to the phenotype of a single cell and finally to population behavior. Agent-based computational approaches have a natural modular architecture, which reflects the modular organization of biological systems. We used bacterial chemotaxis, one of the best-characterized biological systems, as a test-bed for the development of our computational framework.

Results: We have developed AGENTCELL, a multi-scale agent-based simulation platform to model the relationship between intracellular processes in individual cells and the behavior of a cellular population. We used the chemotaxis system of the *Escherichia coli* bacterium to simulate individual molecular interactions within single swimming cells. In-

ular signaling pathways. Consequently, even genetically identical cells can exhibit different behaviors. To investigate the relationship between population behavior and the particular phenotype of each individual cell, we need a numerical approach that combines intracellular biochemistry, single cell behavior and population dynamics, all within one framework. A simulation platform with these features would allow us to incorporate time- and context-dependency of signaling pathways (Zhu et al. 2003). These features could be used to study phenomena such as quorum sensing where the dynamic coupling between intracellular events and population dynamics is important (Shapiro 1998).

Most computational models characterize biological systems at one specific scale of interest: e.g. molecular, cellular, or inter-cellular. Computational tools such as GEPASI (Mendes 1993), DBSOLVE (Goryanin et al., 1999) and STOCHSIM (Morton-Firth 1998, Le Novère and Shimizu 2001) model intracellular biochemical reactions within one cell. At the whole cell level, there exist more integrative



Agents are “problem solving entities with well-defined boundaries and interfaces”.

Agents have goals and can determine if their situation becomes better or worse relative to the fulfillment of these goals.

Agents act on locally available information; global or system-wide information is not accessible.

The main difference between agents and objects is that agents are autonomous: they have the ability to control their internal state and behavior without the direction of a central authority (Wooldridge 1997).

Autonomy decentralizes decision making and therefore greatly simplifies the implementation of the over whole system’s control. Because each agent decides by itself when to act and what action to perform, there is no need for a complex centralized decision making entity.

Agents follow protocols to interact with their environment and to interact with each other.

Because of their autonomy, agents are free to make run-time decisions about the scope (with whom to interact) and nature of the interactions.

Flexibility in the timing, scope and nature of the interactions is one of the advantages of agent-based.



Simulation of arterial hypertension

1. Haemodynamic block – graph (vessels as edges, 1D PDE for blood flow in each vessel).
2. Kidney block - molecular pathways of kidney processes involved in regulation of arterial blood pressure and water salt balance.
3. Endocrine regulation block – pathways effects of hypothalamus, adrenal gland, and atrium hormones on cardio-vascular system.
4. Nerve activity block - simulating action of sympathetic and parasympathetic nerve systems affecting renal processes, blood vessel tone, etc.

INTEGRATED APPROACH FOR SIMULATION OF PHYSIOLOGICAL, BIOMECHANICAL, AND MOLECULAR-GENETICAL ASPECTS OF ARTERIAL HYPERTENSION



¹Fedor A. Kolpakov^{1,2}, Ruslan N. Sharipov^{1,2,3}, Elina A. Biberdorf⁴, Yuriy L. Trakhinin⁴, Mikhail V. Puzanov^{1,2}, Alexander V. Koshukov^{1,2}, Alexander M. Blokhin⁴, Arkadiy L. Markel³, Ludmila N. Ivanova³

¹*Institute of Systems Biology, Novosibirsk, Russia;*

²*Design Technological Institute of Digital Techniques SB RAS, Novosibirsk, Russia;*

³*Institute of Cytology and Genetics SB RAS, Novosibirsk, Russia;*

⁴*Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia*

[†]Contacts: fedor@biouml.org

Summary

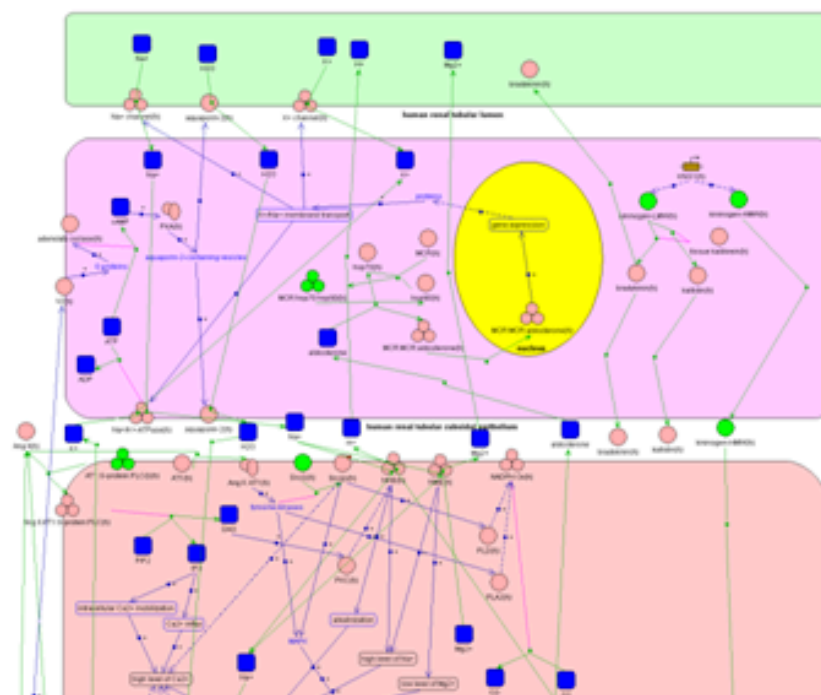
Arterial hypertension (AH) is an important risk factor for development of cardiovascular diseases, which cause a significant part of death cases in developed countries. A number of models simulating parts of cardiovascular system (CVS) have been created of different degree of complexity and precision. We have been designing a new long-term model of this system, which is a result of physico-mathematical and biological approaches.

Tasks and Goals

Creation of database on arterial hypertension containing both theoretical and experimental data and its integration with international specialized databases.

2. Improvement of theoretical and computational tools for modeling of human hemodynamics.

3. Design of the modified mathematical model of vascular tree describing human cardiovascular system (CVS) and also models of gene networks of separate biological processes which take part in arterial blood pressure regulation in normal and hypertensive state.



Results

•Arterial hypertension is a particular case of CVS infringement. Physiology, biophysics (biomechanics) and molecular biology (genetics) are tightly related in its regulation. Simultaneously we are developing both physico-mathematical and biological components of our CVS model. We make an accent on modeling of the kidney processes because it plays a very important part mediating control of arterial blood pressure.

•Four main blocks of the model have been outlined: haemodynamics, regulation of water-salt balance by the kidney, endocrine regulation and nerve activity (sympathetic and parasympathetic).

1. The haemodynamic block has been designing on the basis of several mathematical models. Combination of good tested and effective methods of line and orthogonal sweep (S.K. Godunov) have also been used for calculations.



Suggestion

BioUML team can implement drafts of SBML specifications as BioUML plug-ins to test suggested approaches from practical view point during routine work.

Acknowledgements

Part of this work was partially supported by following grants:

- Volkswagen-Stiftung (I/75941),
- INTAS Nr. 03-51-5218
- RFBR Nr. 04-04-49826-a
- Siberian Branch of Russian Academy of Sciences (interdisciplinary projects № 46).

Author is grateful to for useful comments, discussions and technical support

*Alexander Kel
Sergey Zhatchenko*

Software developers

*Mikhail Puzanov Alexandr Koshukov
Vasiliy Hudyakov Vlad Zhvaleev
Oleg Onegov Igor Tyazhev
Artem Shaidukov*

Annotators

*Ruslan Sharipov
Elena Cheremushkina
Ekaterina Kalashnikova*



Refined definition of graphic notation

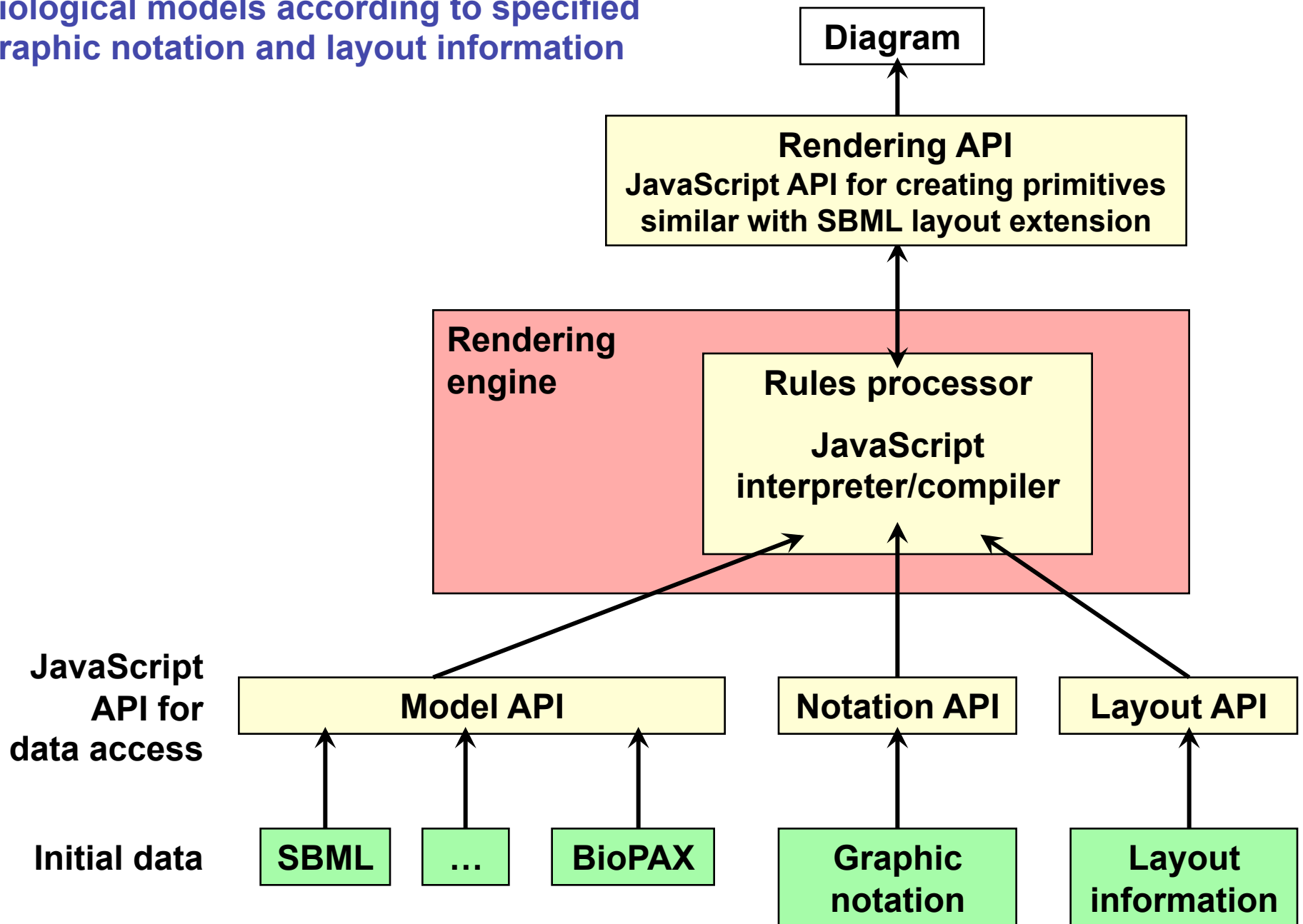
- Object types
(for example, protein, gene, reaction)
- Object properties
(for example, phosphorylated)
- User defined object properties - layout information
(for example, size, color, location)
- Rules for objects and their properties visualization
- Rules for semantic control of diagram integrity
- Selection, highlighting, filtering



Formal definition of graphic notation as XML document and integration with SBML format

Graphic notation components	Defined as	SBML
Object types	XML	<annotation>
Object properties	XML	<annotation>
User defined properties	XML	<layout>
Rules for visualization	JavaScript	
Rules for semantic control	JavaScript	
Selection, highlighting	JavaScript	

Basic software architecture for rendering of biological models according to specified graphic notation and layout information





Process Diagrams as a test case

We plan to test our approach on Process Diagrams. This work will include:

- Formal definition of graphic notation as XML document.
- Integration with SBML format.
- Developing of plug-in for BioUML workbench that will implement suggested architecture
 - BioUML meta-model will be used as **Model API** and **Layout API**;
 - BioUML DiagramType will be used as **Graphic notation API**;
 - BioUML graphics library will be used as **Rendering API**;
 - BioUML DiagramViewBuilder will be used as prototype for Rendering Engine.