BIOUML – FRAMEWORK FOR VISUAL MODELING AND SIMULATION BIOLOGICAL SYSTEMS

Fedor A. Kolpakov

Biosoft.Ru, Novosibirsk, Russia.
Digital Design Technolgy Institute SB RAS, Novosibirsk, Russia.
e-mail: fedor@biosoft.ru

Keywords: systems biology, graphic notation, simulation, ODE, MATLAB, Java, GeneNet.

Resume

Motivation: With the completion of several genomics initiatives, including the Human Genome Project, researchers are poised to begin the next phase of elucidating how living systems function. Systems biology, a synergistic application of experiment, theory and modeling towards understanding biological processes as whole systems, requires integrated software environment that spans the comprehensive range of capabilities including access to databases with experimental data, tools for formalized description of biological systems structure and functioning, as well as tools for their visualization and simulations.
Results: Here we describe architecture and structure of BioUML framework designed for formalized graphic notation of biological systems structure and functioning, their simulations and access to databases on biological pathways. BioUML meta model provides an abstract layer to present structure of any biological system as a clustered graph. BioUML viewer and editor provide visualization of these graphs as diagrams and their editing. To incorporate any databases on biological pathways into BioUML framework we introduce a module concept and demonstrate it by creating module for GeneNet database. BioUML modeler allows a user to create and modify visual diagrams of biological systems and provides automatic generation of their executable models as MATLAB M-files. Using MATLAB these models can be simulated and investigated.
Availability: http://www.biouml.net; http://groups.yahoo.com/group/biouml.

Introduction

BioUML is designed as common purpose framework for systems biology providing formalized graphic notation of biological systems structure and functioning, their visualization and simulations as well as access to databases with relevant experimental data. BioUML is mostly oriented towards representing biochemical networks including cell signaling pathways, metabolic pathways, gene networks and molecular genetics systems.
BioUML framework is Java application consisting from following parts:

- *meta model* – provides an abstract layer to present structure of any biological system as a clustered graph.
- *BioUML viewer* – a universal viewer to visualize graphs of biological systems structure as diagrams.
- *BioUML editor* – universal diagram editor.
- *BioUML search engine* - it allows a user to create graphs of related biological entities. It provides similar functionality with TRANSPATH search engine

(Schacherer et al., 2001); however the resulted graph can be edited and customized by a user using BioUML editor. Currently it is under construction.

- *BioUML modeler* - allows a user to model/simulate dynamics of biological systems using block diagrams.
- *Database modules* – provides incorporation of different databases on biological pathways into BioUML framework
- *Standard diagram and data types* – an attempt to standardize data types and graphic notations for biological pathways.

Meta model

The core of BioUML framework is meta model (fig.1) providing an abstract layer to present structure of any biological system as a clustered graph that further can be visualized as a diagram (fig. 2) or stored as XML file (http://www.biouml.net/xml.shtml).

Class `DiagramElement` defines common attributes for all graph elements. Any instance of this class contains reference to corresponding object from a concrete database. By this way we wrap arbitrary database object to be element of diagram. To provide automatic executable model generations we can associate a role (variable or right side of equation) with any diagram element.

All graph edges are directed and are instances of `Edge` class. Simple graph nodes are instances of `Node` class. `Compartment` class is used to group several nodes in one compartment. `EquivalenceNodeGroup` is a special case of compartment to group nodes equivalent in a given context, for example homologous genes or proteins. All diagrams are instances of the same class `Diagram`. To take into account different diagram types, `Diagram` contains `type` attribute that is instance of `DiagramType` class. `DiagramType` is responsible for visualization and semantic control of specific diagram. For this purpose it uses specialized subclasses of `DiagramViewBuilder` (to generate diagram view) and `SemanticController` to check the diagram semantics. `DiagramType` also defines what classes (database objects) can be used as nodes and edges.

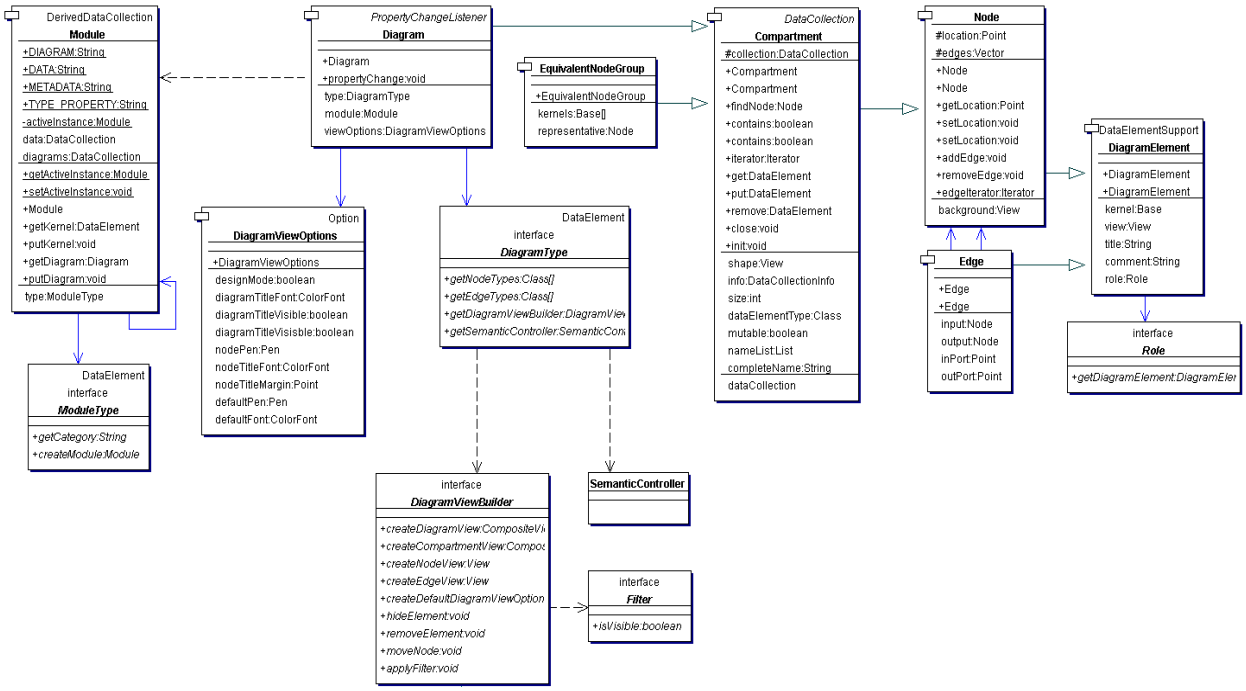

Figure 1. BioUML meta model package.

Module concept

We introduce a concept of modules that serve for incorporation of different databases on biological pathways into BioUML framework. If we use a metaphor, then BioUML can be imagined as operating system (for example Windows), and modules are concrete programs (for example MS Word).
Concrete BioUML modules defines mapping of database content into diagram elements and diagram types that can be used with the database. As was described above, diagram type also specifies concrete classes of `DiagramViewBuilder` to generate database specific diagram view and `SemanticController` to check the diagram semantics.
Currently we have developed module for GeneNet database (Kolpakov et al., 1998) and now we are working on modules for TRANSPATH (Schacherer et al., 2001) and KEGG (Kanehisa et al., 2002) databases. We provide all sources of GeneNet module (http://www.biouml.net/genenet.shtml) to allow other developers to use them as an example to incorporate they databases into BioUML framework.

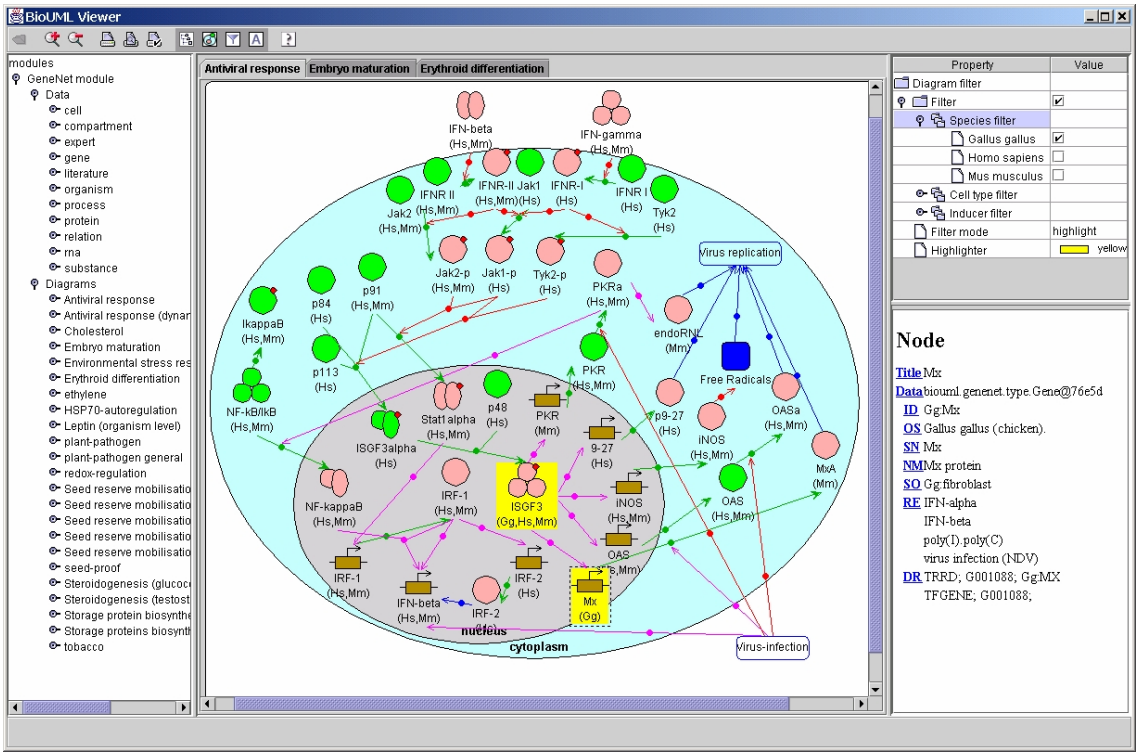

Figure 2. BioUML viewer screen short. Left pane shows GeneNet database content that was incorporated as GeneNet module; central pane shows "Antiviral response" diagram; right top pane shows filters that are applied to the diagram, right bottom pane shows properties of "Mx" graph node. The data are taken from the publicly available version of GeneNet database (http://wwwmgs.bionet.nsc.ru/mgs/systems/genenet/).

Standard diagram and data types

We try to standardize data types and graphic notation for description biological pathways structure and they simulation using approach suggested in GeneNet system (Kolpakov et al., 1998) as a start point. Currently we specify data three diagram types (http://www.biouml.net/standard.shtml):

1) Pathway structure diagram – the diagram type to present metabolic and signal transduction pathways structure.
2) Pathway simulation diagram – extension of pathway structure diagram where variables are associated with graph nodes and differential equations – with graph edges.

3) Generalized pathway diagram – a pathway structure diagram generalized by different species, cell lines or experimental conditions.

BioUML modeler

BioUML modeler allows a user to model continuous dynamics systems that can be represented by system of ordinary differential equations (ODEs). A simulated biological system is presented as pathway simulation diagram (Fig. 3). To specify right side of differential equation MATLAB language is used, additionally some conventions is used for variable names. When such diagram is build, BioUML modeler allows user to automatically generate executable models as MATLAB M-files (Fig. 4) and start powerful MATLAB ODE suite (Shampine, Reichelt, 1997) for model simulations.

Example below demonstrates application of BioUML modeler for simulations simple pharmacokinetic model. Here 100 units of some drug A were injected intravenously. This drug can be break up by some enzyme E in liver giving the metabolite B. We suggest that drug flow from blood to liver is proportional to drag amount of in the blood. The same is true for drug flow from liver to blood, however the constant is other (k_1 in first case and k_2 in second case). We also assume that enzyme concentration in liver is E_0 and the dynamics of break up reaction can be described using Michaelis-Menten equation. Figure 4 shows the model diagram and result of its simulation, figure 5 demonstrates M-files that were automatically generated by BioUML modeler.

To simplify creation of complex diagram we are developing a set of standard reactions. It will be similar with standard blocks in MATLAB/Simulink – user should only specify some reaction constants while the needed differential equations will be generated automatically. The other direction of future current work is implementation of hybrid models to allow us joint modeling of a continuous subsystem with logical components.

Conclusion

We described the current status of BioUML framework. However our ultimate goal is to create of visual language for systems biology similar to UML. We would like to involve scientific community in this process and we provide special forum http://groups.yahoo.com/group/biouml/ for this purpose.

Acknowledgements

Part of this work was supported by the grant of Volkswagen-Stiftung (I/75941) and company DevelopmentOnTheEdge.com that provides its product BeanExplorer (www.beanexplorer.com) for development user interfaces for BioUML framework. Author is grateful to Sergey Zhatchenko and Alexander Kel for useful comments and discussions, as well as to Igor Tyazhev, Vlad Zhvaleev and Oleg Onegov for technical support.

References

1. Kanehisa M., Goto S., Kawashima S., and Nakaya, A. (2002) *The KEGG databases at GenomeNet*. Nucleic Acids Res., **30**, 42-46.
2. Kolpakov F.A., Ananko E.A., Kolesov G.B. and Kolchanov N.A. (1998) *GeneNet: a database for gene networks and its automated visualization*. Bioinformatics, **14(6)**, 529-537.
3. Schacherer F., Choi C., Gotze U., Krull M., Pistor S., Wingender E. (2001) *The TRANSPATH signal transduction database: a knowledge base on signal transduction networks*. Bioinformatics, **17(11)**, 1053-1057

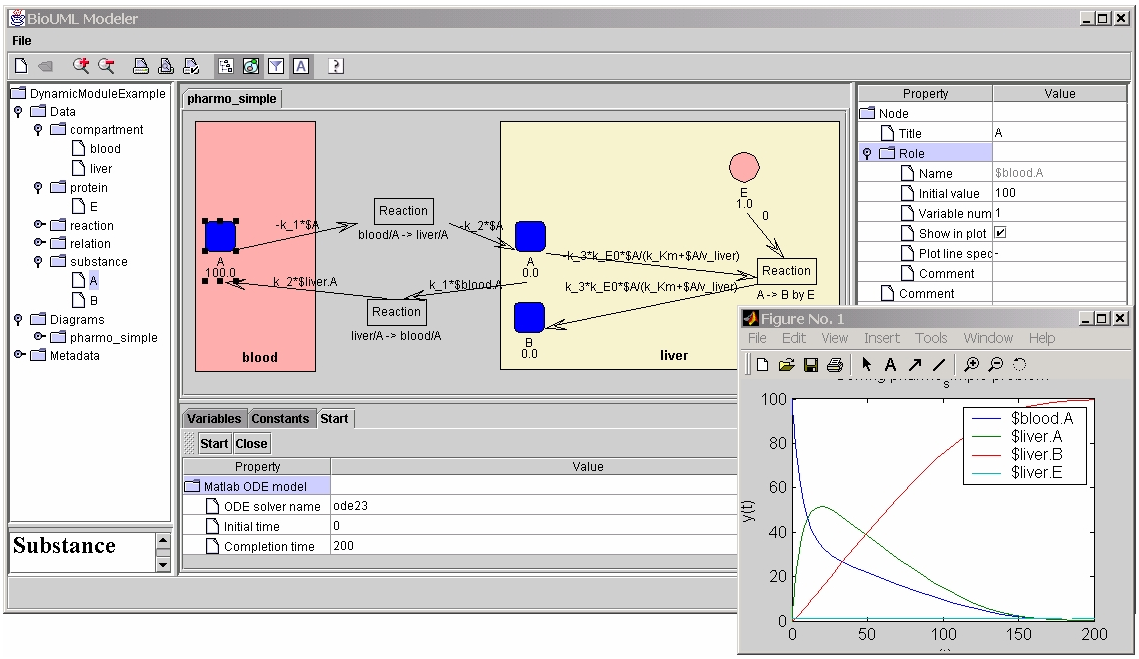

Figure 3. BioUML modeler graphic user interface. Strings above arrows are right sides of differential equations associated with graph edges; numbers – are initial values of corresponding variables associated with graph nodes. Bottom right – result of model simulation using MATLAB.

```
%script for 'pharmo_simple' model simulation
%constants declaration
global k_1 k_2 k_3 k_E0 k_Km v_blood v_liver
k_1 = 0.1
k_2 = 0.05
k_3 = 0.01
k_E0 = 1.0
k_Km = 0.1
v_blood = 100.0
v_liver = 100.0

%Model variables and their initial values
y = []
y(1) = 100.0                            % y(1) - $blood.A
y(2) = 0.0                              % y(2) - $liver.A
y(3) = 0.0                              % y(3) - $liver.B
y(4) = 1.0                              % y(4) - $liver.E

%numeric equation solving
[t,y] = ode23('pharmo_simple_dy',[0 200],y)

%plot the solver output
plot(t, y(:,1),'-',t, y(:,2),'-',t, y(:,3),'-',t, y(:,4),'-')
title ('Solving pharmo_simple problem')
ylabel ('y(t)')
xlabel ('x(t)')
legend('$blood.A','$liver.A','$liver.B','$liver.E');

--------------------------------------------------------------------------------
function dy = pharmo_simple_dy(t, y)
% Calculates dy/dt for 'pharmo_simple' model.

%constants declaration
global k_1 k_2 k_3 k_E0 k_Km v_blood v_liver

% calculates dy/dt for 'pharmo_simple' model
dy = [ -k_1*y(1)+k_2*y(2)
       -k_3*k_E0*y(2)/(k_Km+y(2)/v_liver)-k_2*y(2)+k_1*y(1)
       k_3*k_E0*y(2)/(k_Km+y(2)/v_liver)
       0]
```

Figure 4. Generated by BioUML modeler M-files to simulate 'pharmo_simple' model. Top - script file for model simulation and graphic result presentations; bottom - function to calculate dy/dt for the model.