



0100100010011101

ISB

Encoding and exchanging graphical representation: architecture and formats

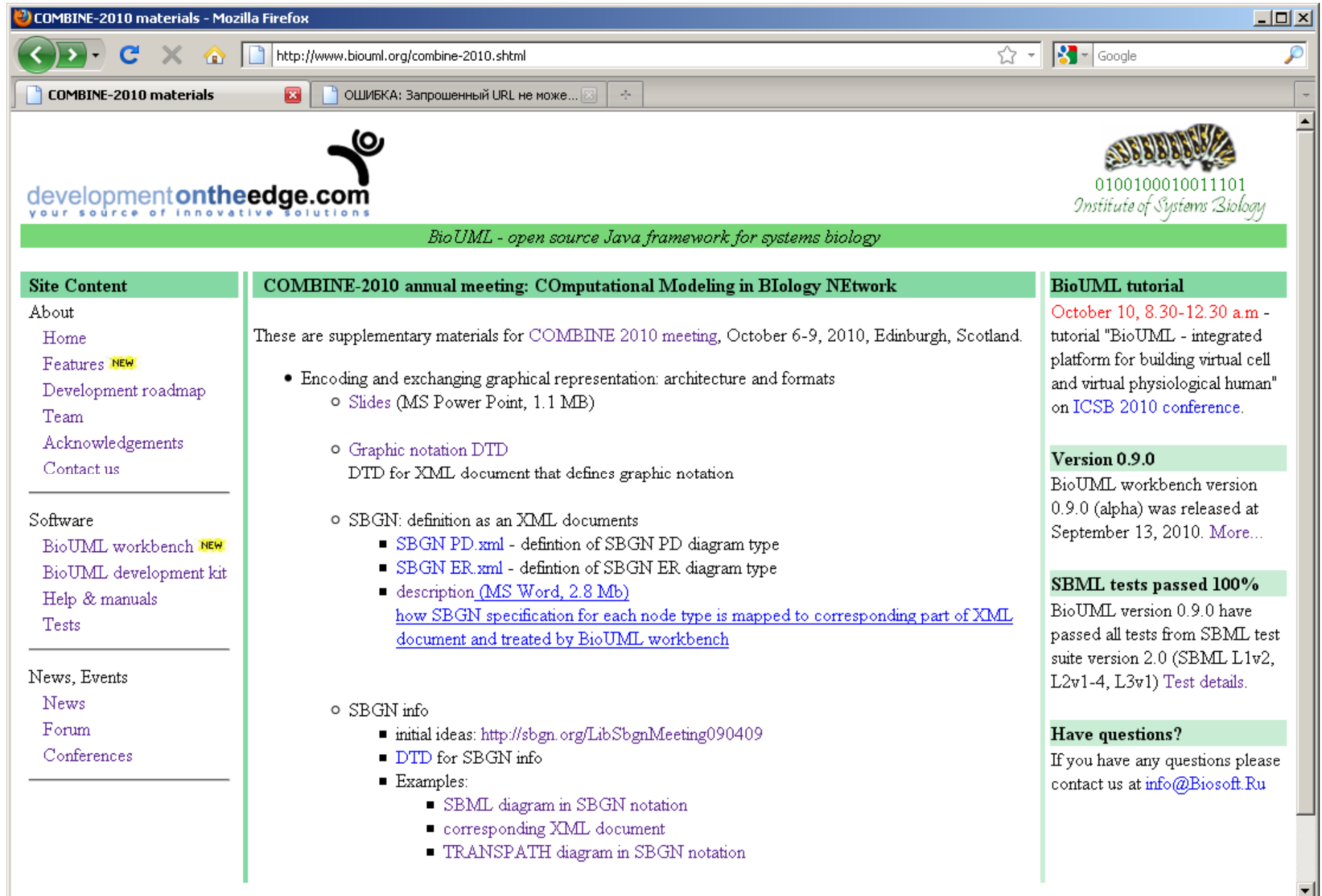
Fedor Kolpakov

Institute of Systems Biology

Novosibirsk, Russia

COMBINE-2010, October 7, 2010

Supplementary materials: <http://www.biouml.org/combine-2010.shtml>



The screenshot shows a Mozilla Firefox browser window with the address bar containing <http://www.biouml.org/combine-2010.shtml>. The page content is organized into three main columns:

- Left Column (Site Content):**
 - About
 - Home
 - Features **NEW**
 - Development roadmap
 - Team
 - Acknowledgements
 - Contact us
 - Software
 - BioUML workbench **NEW**
 - BioUML development kit
 - Help & manuals
 - Tests
 - News, Events
 - News
 - Forum
 - Conferences
- Middle Column (COMBINE-2010 annual meeting: CComputational Modeling in BIology NETWORK):**

These are supplementary materials for [COMBINE 2010 meeting](#), October 6-9, 2010, Edinburgh, Scotland.

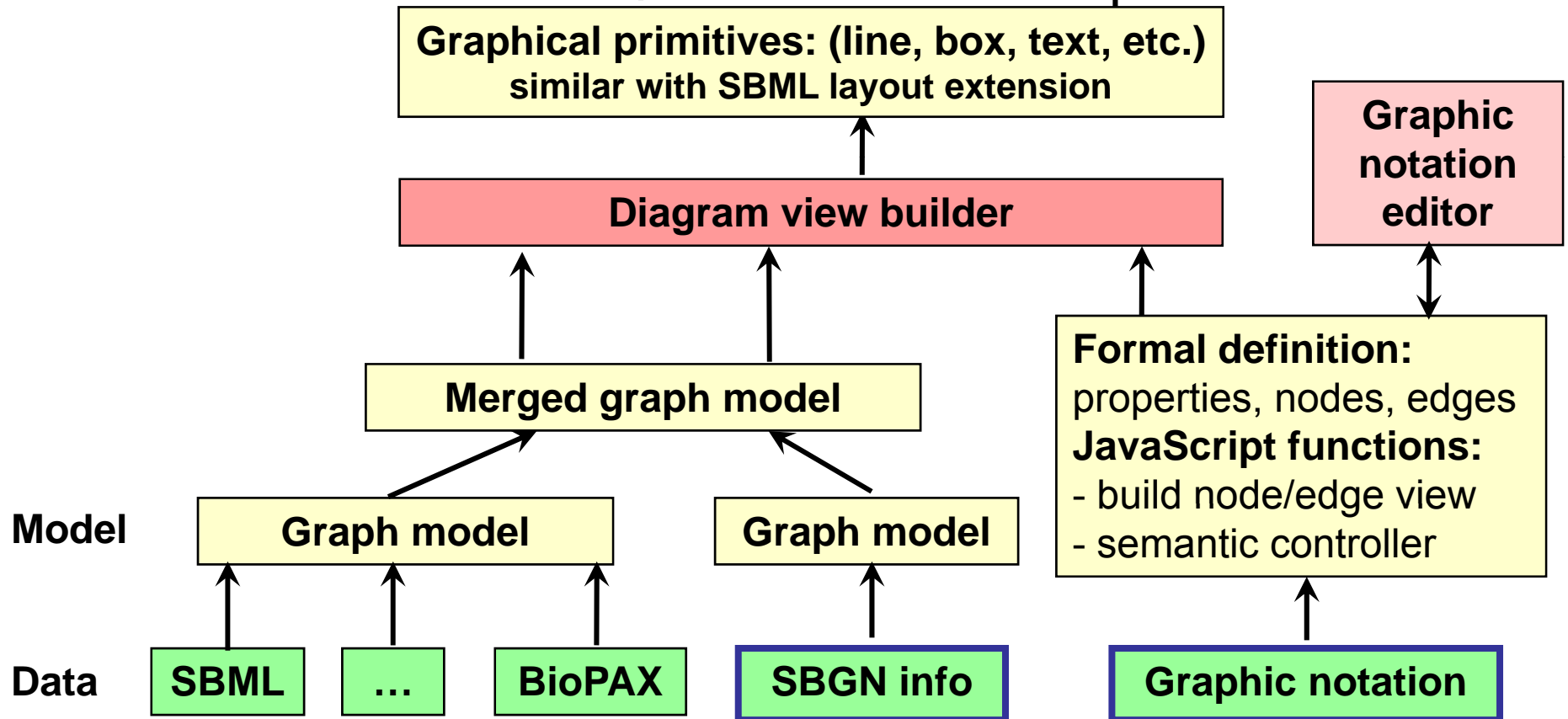
 - Encoding and exchanging graphical representation: architecture and formats
 - Slides (MS Power Point, 1.1 MB)
 - Graphic notation DTD
 - DTD for XML document that defines graphic notation
 - SBGN: definition as an XML documents
 - [SBGN PD.xml](#) - definition of SBGN PD diagram type
 - [SBGN ER.xml](#) - definition of SBGN ER diagram type
 - description ([MS Word, 2.8 Mb](#))
 - [how SBGN specification for each node type is mapped to corresponding part of XML document and treated by BioUML workbench](#)
 - SBGN info
 - initial ideas: <http://sbgn.org/LibSbgnMeeting090409>
 - DTD for SBGN info
 - Examples:
 - SBML diagram in SBGN notation
 - corresponding XML document
 - TRANSPATH diagram in SBGN notation
- Right Column (BioUML tutorial):**
 - BioUML tutorial**
 - October 10, 8.30-12.30 a.m - tutorial "BioUML - integrated platform for building virtual cell and virtual physiological human" on [ICSB 2010 conference](#).
 - Version 0.9.0**
 - BioUML workbench version 0.9.0 (alpha) was released at September 13, 2010. [More...](#)
 - SBML tests passed 100%**
 - BioUML version 0.9.0 have passed all tests from SBML test suite version 2.0 (SBML L1v2, L2v1-4, L3v1) [Test details](#).
 - Have questions?**
 - If you have any questions please contact us at info@Biosoft.Ru

Rendering,
editing

Java client
(BioUML workbench)

Web browser using
HTML5 <canvas> element
(BioUML web edition)

Architecture



XML formats

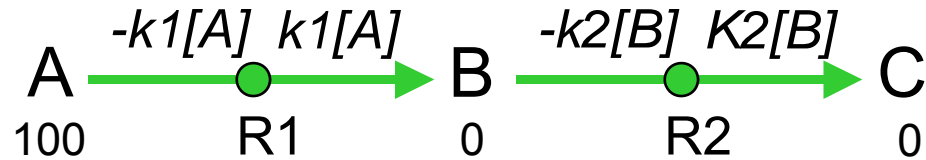
<http://79.125.109.165/bioumlweb/#de=databases/Biopath/Diagrams/DGR0240r>

ID: DGR0240r
Title: Integral diagram of proapoptotic pathway.
Complete name: databases/Biopath/Diagrams/DGR0240r

**Rendering of large diagram by web browser
 (please try: move, zoom, edit)**

Graph model

Example:
system from two chemical reactions



k_1 - reaction rate for R1

k_2 - reaction rate for R2

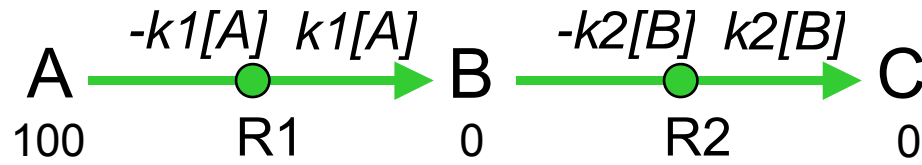
Corresponding mathematical model:

$$\frac{dA}{dt} = -k_1[A]$$

$$\frac{dB}{dt} = k_1[A] - k_2[B]$$

$$\frac{dC}{dt} = k_2[B]$$

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



```

ID  A
CC  ..
...
//
    
```

```

ID  R1
    A->B
...
//
    
```

```

ID  B
CC  ..
...
//
    
```

```

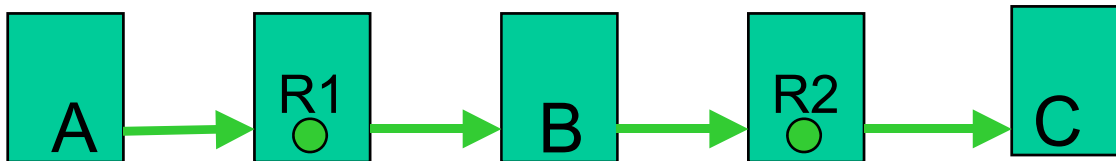
ID  R2
    B->C
...
//
    
```

```

ID  C
CC  ..
...
//
    
```

Description of system components in the database:

predefined standard types + any attributes



System structure is described as a

compartmentalised attributed graph

```

100  -k1[A]  k1[A]  0  -k2[B]  k2[B]  0
    
```

Mathematical model of the system

Graphic notation

Graphic notation editor main concepts

- graphic notation is defined formally as XML document
- graphic notation editor provides user friendly interface for XML document editing
- SBGN graphic notation (prototype) is implemented
- BioUML workbench allows to create and edit diagrams using graphic notation defined as XML document

Graphic notation versus graph layout

- allows edit diagram
- allows to create new diagram
- allows formally define SBGN and use it in diagrams from different sources (SBML, BioPAX, TRANSPATH, Reactome, etc.)
- allows to reuse graphic notation by many tools

Graphic notation – main components

- properties – formal definition of properties that can be used as properties of nodes and edges (for example, title, multimer, etc.). Definition of property includes:
 - name, type
 - short description
 - controlled vocabulary (optional)
- node types – definition of node includes:
 - name
 - icon
 - properties
 - view function (JavaScript)
 - short description
- edge types – definition of edge includes:
 - name
 - icon
 - properties
 - view function (JavaScript)
 - short description
- semantic controller – defines rules for semantic control of diagram integrity. For this purpose it defines following functions:
 - canAccept (JavaScript)
 - isResizable (JavaScript)
 - move (JavaScript)

```
<!ELEMENT graphicNotation
  ( properties,
    nodes,
    edges?,
    viewOptions?,
    semanticController?,
    viewBuilder? )>
<!--
<!ATTLIST graphicNotation
  version          CDATA          "0.7.9"
  appVersion       CDATA          "0.7.9"
>
```

```
<!ELEMENT properties (property*)>
```

```
<!ELEMENT property (tags?)>
```

```
<!ATTLIST property
```

```
  name          CDATA          #REQUIRED
```

```
  type          CDATA          #REQUIRED
```

```
  short-description CDATA          #IMPLIED
```

```
  value         CDATA
```

```
>
```

```
<!ELEMENT tags (tag+)>
```

```
<!ELEMENT tag>
```

```
<!ATTLIST tag
```

```
  name          CDATA          #REQUIRED
```

```
  value         CDATA          #IMPLIED
```

```
>
```

```
<!-- Possible property types: -->
```

```
<!-- - simple types: boolean, int, double, String -->
```

```
<!-- - special types: pen, brush, size -->
```

```
<!-- - array -->
```

```
<!-- - composite -->
```

<!-- Nodes and edges

-->

<!ELEMENT propertyRef>

<!ATTLIST propertyRef

name CDATA #REQUIRED

value CDATA

>

<!ELEMENT nodes (node+)>

<!ELEMENT node (propertyRef*)>

<!ATTLIST node

type CDATA #REQUIRED

icon CDATA #IMPLIED

isCompartment CDATA #IMPLIED

>

<!ELEMENT edges (edge+)>

<!ELEMENT edge (propertyRef*)>

<!ATTLIST edge

type CDATA #REQUIRED

icon CDATA #IMPLIED

>

```
<!-- ViewOptions, ViewBuilder -->
```

```
<!ELEMENT viewOptions (property*)>  
<!ATTLIST viewOptions  
>
```

```
<!ELEMENT viewBuilder (nodeView*), (edgeView*) >  
<!ATTLIST viewBuilder  
  prototype          CDATA          #REQUIRED  
>
```

```
<-- JavaScript code to create node/edge view -->
```

```
<!ELEMENT nodeView>  
<!ATTLIST nodeview  
  type              CDATA          #REQUIRED  
>
```

```
<!ELEMENT edgeView>  
<!ATTLIST edgeView  
  type              CDATA          #REQUIRED  
>
```

```
<!-- Semantic controller                                -->

<!ELEMENT semanticController (canAccept?, isResizable?, move?)>
<!ATTLIST semanticController
  prototype          CDATA          #REQUIRED
>

<-- JavaScript code for semantic controller            -->
<!ELEMENT canAccept>
<!ELEMENT isResizable>
<!ELEMENT move>

<!-- Icons                                            -->
<!ELEMENT icons (icon*)>

<!ELEMENT icon>
<!ATTLIST icon
  name                CDATA          #REQUIRED
>
```

Why JavaScript?

- can be integrated and executed by programs on different languages, including:
 - Java – includes JavaScript support from version 1.6, Rhino JavaScript library
 - C++ – SpiderMonkey library
- can be executed by browser
HTML 5 standard defines `<canvas>` element for drawing. Now diagrams drawing and editing can be done by web browser.
(see BioUML web edition as an example)

Access to graph model from JavaScript functions in graphic notation

```
var multimer = node.getValue(  
    "sbgn:multimer",    // property name  
    0);                // default value
```

```
var entityType = node.getValue(  
    "sbgn:entityType", // property name  
    "unspecified");    // default value
```

Graphic Notation Editor

user interface

Graphic Notations Editor

File Help

sbgn(ST).xml \

- sbgn(ST).xml
 - Properties
 - multimer**
 - text
 - value
 - Nodes
 - Edges
 - Reactions
 - Examples
 - ViewOptions
 - Semantic controller
 - Values set

List of specific properties that are used by graphic notation

Properties editor

Property	Value
Property	
Name	multimer
Type	int
Short description	multimer

Javascript Editor

Javascript Compilation Log

View Edit Editors

Graphic Notations Editor

File Help

sbgn(ST).xml \

- sbgn(ST).xml
 - Properties
 - multimer
 - text
 - value
 - Nodes
 - source-sink
 - tag
 - unspecified entity
 - simple chemical
 - multimer
 - macromolecule
 - complex
 - compartment
 - unit of information
 - variable
 - and
 - or
 - xor
 - not
 - transition

For each node type user can define:

- name
- properties
- icon
- view function (JavaScript)

```
function f(container, node, options, g)
{
    var d, pen, brush;
    d = new Dimension(50,50);
    brush = new Brush(new Color(1.0, 0.9, 1.0));
    pen = options.getDefaultPen();
    var multimer = node.getValue("multimer", 0);
    if(multimer>1)
    {
        var ellipse = new EllipseView(pen, brush, 5, 5, d.width, d.height);
    }
}
```

Property	Value
Properties	
Name	simple chemical
Property count	1
Icon	<input type="radio"/> Select...

Javascript Editor

Javascript Compilation Log

Graphic Notations Editor

File Help

sbgn(ST).xml \

- sbgn(ST).xml
 - Properties
 - Nodes
 - Edges
 - Reactions
 - Examples
 - complex
 - modulation
 - test
 - simple
 - transition
 - transition2
 - test4
 - test2
 - test3
 - ViewOptions
 - Semantic controller
 - Values set

sbgn(ST).xml : complex \

“Examples” node contains a set of diagrams that demonstrates usage of graphic notation.

The diagram illustrates a complex structure within a light blue rounded rectangle. It contains several nested and connected elements:

- An outer light blue rounded rectangle containing:
 - A white rounded rectangle at the bottom left labeled "INFO".
 - A white rounded rectangle at the bottom right labeled "varY".
 - A white rounded rectangle at the top left labeled "INFO".
 - A white rounded rectangle at the top right labeled "var2".
 - A light green rounded rectangle in the center labeled "label3".
 - A pink circle in the lower right labeled "label4".
 - A white rounded rectangle at the bottom center labeled "varX".
- A dashed black rounded rectangle inside the top-left area of the outer rectangle, containing:
 - A white rounded rectangle labeled "INFO".
 - A white rounded rectangle labeled "var1".
 - A light green rounded rectangle labeled "label1".

Property	Value
Compartment	
Title	label1
Comment	
ShapeColor	<input type="text" value="white"/>
Size	
Attributes	

```
function f(container, node, options, g)
{
  var d, pen, brush;
```

Javascript Editor

Javascript Compilation Log

Graphic Notations Editor

File Help

sbgn(ST).xml \

- sbgn(ST).xml
 - Properties
 - Nodes
 - Edges
 - Reactions
 - Examples
 - complex
 - modulation
 - test
 - simple
 - transition
 - transition2
 - test4
 - test2
 - test3
 - ViewOptions
 - Semantic controller
 - Values set

When user selects some element on the diagram he can edit:

- object properties
- JavaScript that builds a view for selected diagram element

```
function f(container, node, options, g)
{
    var d, pen, brush;
    d = new Dimension(50,50);
    brush = new Brush(new Color(1.0, 0.9, 1.0));
    pen = options.getDefaultPen();
    var multimer = node.getValue("multimer", 0);
    f(multimer);
}
```

Property	Value
Node	
Title	label4
Comment	
Size	
Attributes	

Property Value

Node

Title label4

Comment

Size

Attributes

View Edit Editors

Javascript Editor

Javascript Compilation Log

Graphic Notations Editor

File Help

sbgn(ST).xml \

- sbgn(ST).xml
- Properties
- Nodes
- Edges
- Reactions
- Examples
- ViewOptions
- Semantic controller ←
- Can accept
- Is resizable
- Move
- Values set

“Semantic controller” node contains list of JavaScript functions that provide semantic constraints and semantic integrity of the diagram.

Property	Value
Data Collection	
Size	0

```
function f(container, de)
{
    var deType = de.getValue('kernel', '').getType();
    if(deType == "unspecified entity" || deType == "xor" || deType == "or" || deType == '
    || deType == "transition" || deType == "uncertain process" || deType == "omitted pro
    {
        var type = container.getValue('kernel', '').getType();
        if(type == "xor" || deType == "or" || deType == "omitted pro
```

Javascript Editor

Javascript Compilation Log

View Edit Editors

SBGN info

SBGN information: main principles

- initial ideas: <http://sbgn.org/LibSbgnMeeting090409>
- SBGN info:
 - compartmentalized attributed graph (SBGN graph)
 - references to biological object (SBML, BioPAX, TRANSPATH, etc. elements)
 - biological statements (complex structure, modifications, etc.)
 - graph layout
 - style for rendering
 - styles info (predefined pens, brush, colors, etc.)
- SBGN info + graphic notation contains all necessary information to generate and edit diagram
- SBGN info is stored in XML format
 - as XML fragment (inserted into SBML <annotation> element)
 - as XML document (for example, for BioPAX and TRANSPATH)
- Information from SBML, BioPAX, Transpath, and other sources is represented as compartmentalized attributed graph and can be merged with SBGN graph using graph element identifiers.

```
<!ELEMENT sbgn (nodes, edges?, styles?)>
```

```
<!ELEMENT nodes (node)*>
```

```
<!ELEMENT node (label, nodeLayout, nodeRendering?)>
```

```
<!ATTLIST node
```

```
  id          CDATA          #REQUIRED
```

```
  type       CDATA          #REQUIRED
```

```
  parent     CDATA          #REQUIRED
```

```
  ref        CDATA
```

```
  clone      CDATA
```

```
>
```

```
<!ELEMENT label (html?)>
```

```
<!ATTLIST label
```

```
  value      CDATA
```

```
  style      CDATA
```

```
  colorFont  CDATA
```

```
>
```

```
<!ELEMENT nodeLayout>
<!ATTLIST nodeLayout
  x          CDATA          #REQUIRED
  y          CDATA          #REQUIRED
  width      CDATA          #REQUIRED
  height     CDATA
>
```

```
<!ELEMENT nodeRendering>
<!ATTLIST nodeRendering
  style      CDATA          #REQUIRED
  pen        CDATA
  brush      CDATA
>
```

```
<!ELEMENT edges (edge)*>
```

```
<!ELEMENT edge (path?, edgeRendering?)>
```

```
<!ATTLIST edge
```

```
  id          CDATA          #REQUIRED
```

```
  type        CDATA          #REQUIRED
```

```
  from        CDATA          #REQUIRED
```

```
  to          CDATA          #REQUIRED
```

```
  ref         CDATA
```

```
>
```

```
<!--  
path concept is based on Java java.awt.geom.GeneralPath  
concept. Path represents a geometric path constructed  
from straight lines, and quadratic and cubic (B?zier)  
curves.  
-->
```

```
<!ENTITY % segmentType  
" (moveTo | lineTo | quadTo | cubicTo | close) ">
```

```
<!ELEMENT path (segment)+>
```

```
<!ELEMENT segment>
```

```
<!ATTLIST segment
```

type	segmentType	#REQUIRED
x0	CDATA	
y0	CDATA	
x1	CDATA	
y1	CDATA	
x2	CDATA	
y2	CDATA	

```
>
```

```
<!ELEMENT styles (style)*>
```

```
<!ELEMENT style (pen?, brush?, font?)>
```

```
<!ATTLIST style
```

```
  id                CDATA                #REQUIRED
```

```
  comment           CDATA                #REQUIRED
```

```
>
```

```
<!ELEMENT pen>
```

```
<!ATTLIST pen
```

```
  id                CDATA                #REQUIRED
```

```
  width            CDATA                #REQUIRED
```

```
  color            CDATA                #REQUIRED
```

```
  stroke           CDATA
```

```
>
```

```
<!ELEMENT brush>
```

```
<!ATTLIST brush
```

```
  id                CDATA                #REQUIRED
```

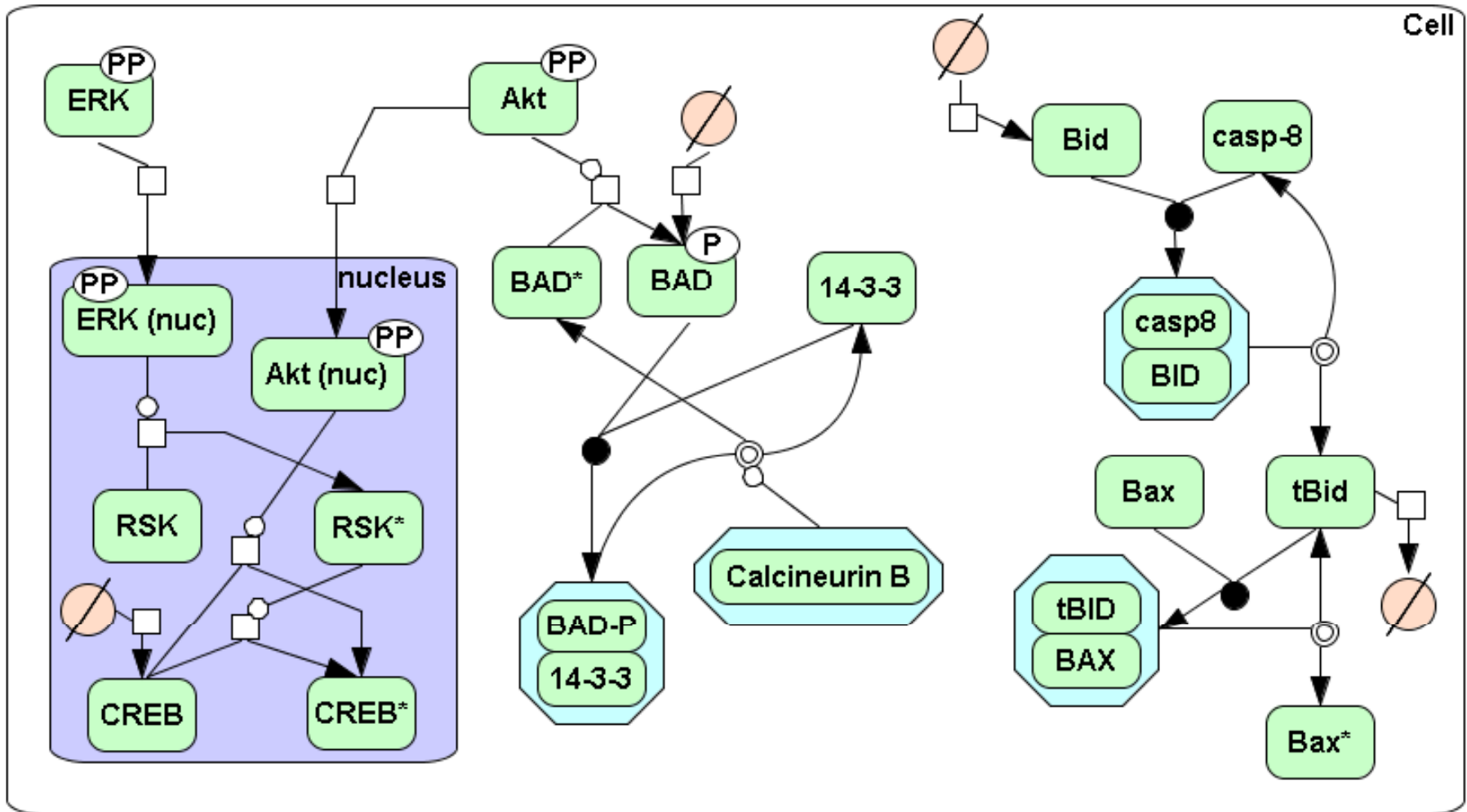
```
  color            CDATA                #REQUIRED
```

```
  . . .
```

```
>
```

```
<!ELEMENT font>
<!ATTLIST font
  id          CDATA          #REQUIRED
  style       CDATA          #REQUIRED
  size        CDATA          #REQUIRED
  bold        CDATA
  italic      CDATA
  underline   CDATA
  color       CDATA
>
```

Example: SBML diagram in SBGN notation



Example: XML fragment for SBML diagram in SBGN notation

```
<annotation>
  <sbgn xmlns:biouml="http://www.biouml.org/ns"
        notation="./graphic notations/sbgn_simulation.xml">

    <nodes>

      <node id="CMP0219" parent="" ref="CMP0219"
            type="compartment">
        <nodeLayout height="450" width="810" x="65" y="40"/>
      </node>

      <node id="PRT003759" parent="CMP0219.CMP0003"
            ref="CMP0219_CMP0003_PRT003759" type="macromolecule">
        <nodeLayout height="40" width="94" x="96" y="195"/>
      </node>

      <node id="PP" parent="CMP0219.CMP0003.PRT003759"
            ref="" type="variable">
        <nodeLayout x="102" y="185"/>
      </node>

    </nodes>

  </sbgn>
</annotation>
```

```
<edges>
```

```
  <edge id="RCT005640: PRT003081 as product"  
    ref="RCT005640: PRT003081 as product"  
    from="CMP0219.CMP0003.RCT005640"  
    to="CMP0219.CMP0003.PRT003081"  
    type="production">
```

```
    <path>
```

```
      <segment segmentType="moveTo" x0="798" y0="224" />
```

```
      <segment segmentType="quadric" x0="822" y0="163" />
```

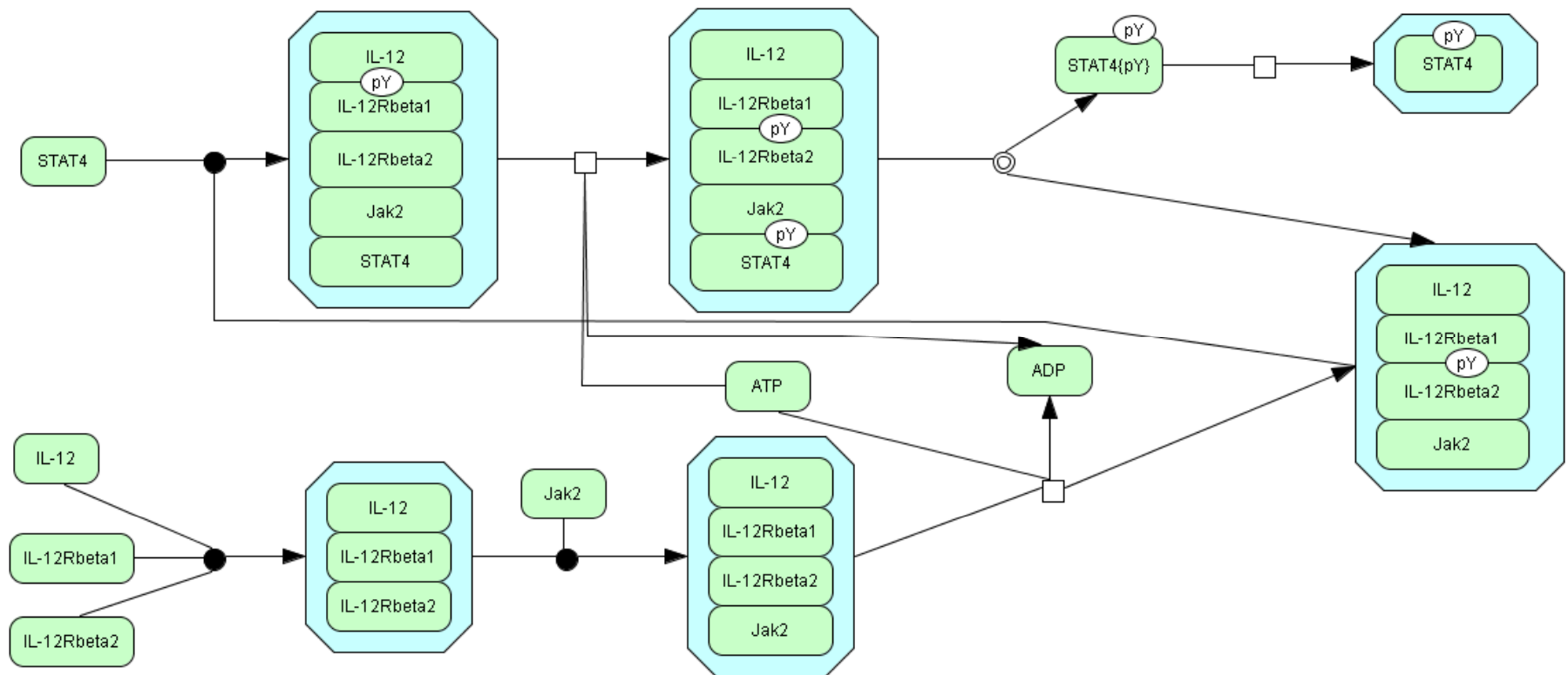
```
      <segment segmentType="lineTo" x0="763" y0="134" />
```

```
    </path>
```

```
  </edge>
```

```
...
```

Example: TRANSPATH pathway in SBGN notation (automatically generated diagram)

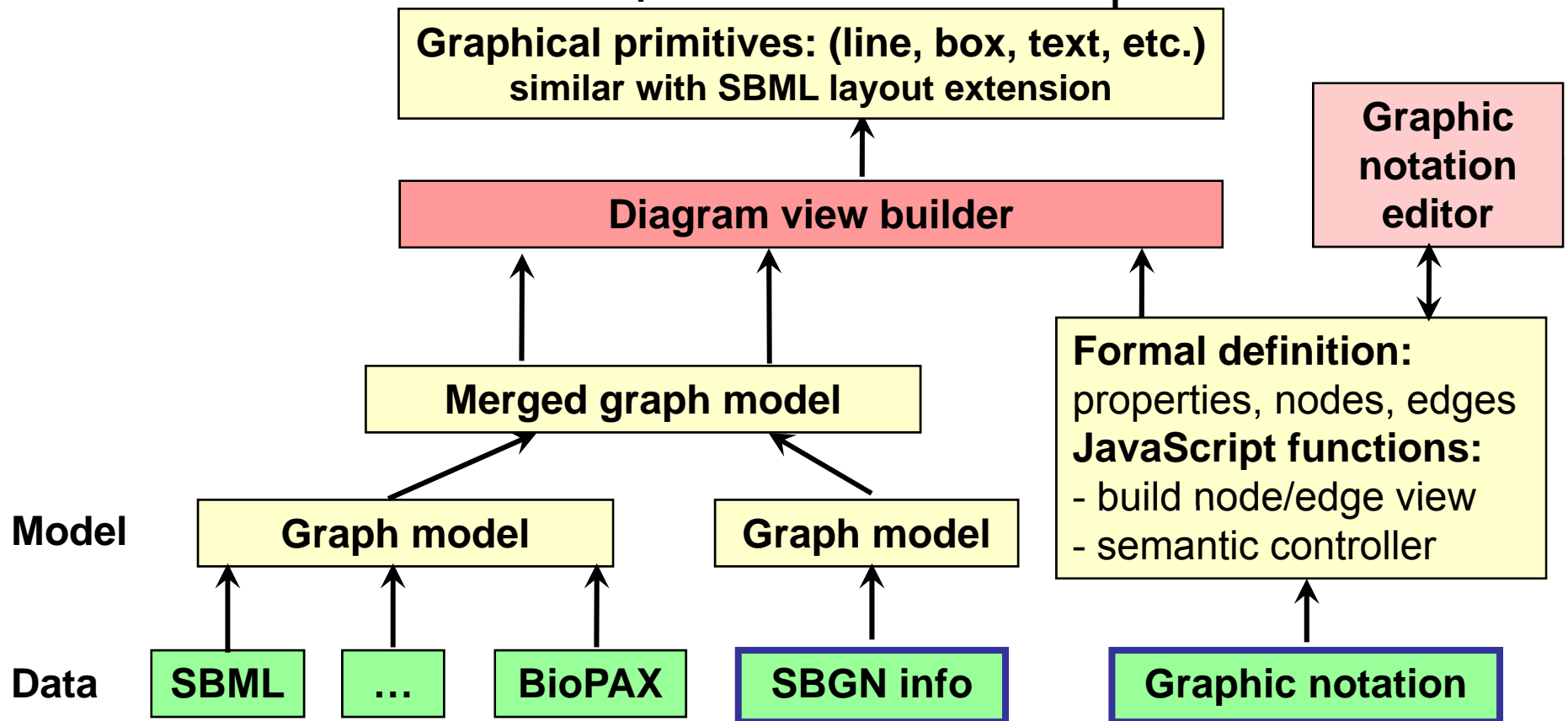


Rendering,
editing

Java client
(BioUML workbench)

Web browser using
HTML5 <canvas> element
(BioUML web edition)

Architecture



XML formats

Acknowledgements

Part of this work was partially supported by following grants:

- European Committee grant “Net2Drug” and “LipidomicNet”
- Siberian Branch of Russian Academy of Sciences (interdisciplinary projects № 46, 17, 91)

BioUML team (2010):

- Nikita Tolstyh
- Tagir Valeev
- Alexey Shadrin
- Ilya Kiselev
- Elena Kutumova
- Anna Ryabova
- Boris Semisalov