

Including Layout Information in SBML Files

Ralph Gauges, Ursula Rost, Sven Sahle and Katja Wegner
European Media Laboratory
Schloss-Wolfsbrunnen Weg 33
69118 Heidelberg
Germany

July 5, 2003

Introduction

With SBML there now is a common standard for the exchange of dynamical systems data which has already been adopted by many applications in this field [1, 2, 3]. Since SBML had no means of storing layout information for reaction networks, we developed an extension to SBML that would allow us to store this layout information in SBML files. Fortunately, SBML already provides a very flexible way of extending the specification through `<annotations>` tags which is a container for arbitrary XML data.

Programs that use this means of extending the SBML specification should place those extensions in a separate unique namespace to avoid interference with extensions of other programs. There already exists an extension to SBML by Herbert M. Sauro that deals with layout information in SBML files [5] well tailored to his program JDesigner [4]. However, in order to provide generality, our specification tries to limit the extensions to just specifying information that concerns the placement of the objects and leaves the rendering to the application.

Since we only add layout information the SBML specification, our extensions are few and simple.

Namespace

For the extensions we use a separate namespace of the following form `xmlns:sl="http://projects.eml.org/bcb/sbml/level2"`. A SBML file that would utilize the extension could have the following form:

```
<?xml version="1.0" encoding="UTF-8"?>
  <sbml xmlns:sbml="http://www.sbml.org/sbml/level2" level="2" version="1"
        xmlns:sl2="http://projects.eml.org/bcb/sbml/level2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://projects.eml.org/bcb/sbml/level2
        http://projects.eml.org/bcb/sbml/level2/layout2.xsd">
```

Metainformation

All the layout classes below are now derived from a class called SBase which should be identical to the one defined in SBML2. This enables programs to store metainformation with the layout objects. Since there is no Schema for SBML Level 2 yet, we just took the SBase definition from SBML Level 1

and added an metaid attribute which is of type NMTOKEN as defined in the RDF specification.

```
<xsd:complexType name="SBase" abstract="true">
  <xsd:annotation>
    <xsd:documentation>
      Designed to allow a modeler or a package to attach
      information to each component.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="notes" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any namespace="http://www.w3.org/1999/xhtml"
            processContents="skip" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="annotations" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any processContents="skip" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="metaid" type="xsd:NMTOKEN" use="optional"/>
</xsd:complexType>
```

<listOfLayouts > and <layout >

Due to the discussion on the sbml mailing list, we took the layout information out of the actual model into a separate tag called listOfLayouts which is placed within the sbml tag after the model definition. This list can hold one or more layout objects which in turn hold layout information for some or all elements of the sbml model plus additional objects that need not be connected to the model. The only attribute for the <layout > tag is an id which uniquely identifies the layout object. Ids are defined to be the same as SId in SBML Level 2.

```

<xsd:simpleType name="SId">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="(_|[a-z]|[A-Z])(_|[a-z]|[A-Z]|[0-9])*"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ListOfCompartmentGRs">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:sequence>
        <xsd:element name="compartmentGR" type="s12:CompartmentGR" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ListOfSpeciesGRs">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:sequence>
        <xsd:element name="speciesGR" type="s12:SpeciesGR" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ListOfReactionGRs">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:sequence>
        <xsd:element name="reactionGR" type="s12:ReactionGR" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ListOfAdditionalGraphicalObjects">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:sequence>
        <xsd:any namespace="##other" processContents="skip"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Layout">
  <xsd:sequence>

```

```

    <xsd:element name="listOfCompartmentGRs" type="s12:ListOfCompartmentGRs"
        minOccurs="0"/>
    <xsd:element name="listOfSpeciesGRs" type="s12:ListOfSpeciesGRs" minOccurs="0"/>
    <xsd:element name="listOfReactionGRs" type="s12:ListOfReactionGRs" minOccurs="0"/>
    <xsd:element name="listOfAdditionalGraphicalObjects"
        type="s12:ListOfAdditionalGraphicalObjects" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="s12:SIId"/>
</xsd:complexType>

<xsd:complexType name="ListOfLayouts">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:sequence>
        <xsd:element name="layout" type="s12:Layout" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="listOfLayouts" type="s12:ListOfLayouts"/>

```

<Compartment> Layout Information

For compartments we specify a layout tag that hold the location of the compartment as x, y and z coordinates and the size as width, height and depth. All values are of type double and the origin is in the upper left to facilitate the implementation. The z and depth value are optional and their value defaults to 0.0 if not specified otherwise. In addition we specify an attribute called **id** which uniquely identifies the compartmentGR element as well as a **ref** attribute which is the id of the corresponding compartment in the model. The **ref** attribute is optional to allow the program to specify compartment representations that do not have a direct correspondence in the model.

XML Schema representation:

```

<xsd:complexType name="CompartmentGR">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:attribute name="ref" type="s12:SIId" use="optional"/>
      <xsd:attribute name="id" type="s12:SIId"/>
      <xsd:attribute name="x" type="xsd:double"/>
      <xsd:attribute name="y" type="xsd:double"/>
      <xsd:attribute name="z" type="xsd:double" use="optional" default="0.0"/>
      <xsd:attribute name="w" type="xsd:double"/>
      <xsd:attribute name="h" type="xsd:double"/>
      <xsd:attribute name="d" type="xsd:double" use="optional" default="0.0"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

example:

```
.  
.   
.   
<compartment id="compartment" volume="1"/>  
.   
.   
.   
<sl2:listOfCompartmentGRs>  
  <sl2:compartmentGR id="cGR" ref="compartment"  
    x="10.0" y="10.0" w="60" h="50"/>  
</sl2:listOfCompartmentGRs>  
.   
.   
.
```

<Specie> Layout Information

Since an sbml document can contain species that don't appear in any reaction a species can have zero or more representations on screen which represented by <specieGR> are grouped in a <listOfSpeciesGRs> tag. Each <specieGR> tag has a unique **id** which is referenced in the layout information of the corresponding SpecieReference object (see below). The actual layout information is given as x, y and z coordinates as well as width, height and depth. As in the compartment layout, the z coordinate and the depth value are optional. They default to 0.0 if not specified. In addition the specieGR object has a **ref** attribute which is the id of the corresponding species object in the model. The **ref** attribute is optional to allow the program to specify species representations that do not have a direct correspondence in the model. This might be useful if some pathway has been collapsed, but is still treated by layout programs.

XML Schema representation:

```
<xsd:complexType name="SpeciesGR">  
  <xsd:complexContent>  
    <xsd:extension base="sl2:SBase">  
      <xsd:attribute name="ref" type="sl2:SId" use="optional"/>  
      <xsd:attribute name="id" type="sl2:SId"/>  
      <xsd:attribute name="x" type="xsd:double"/>  
      <xsd:attribute name="y" type="xsd:double"/>  
      <xsd:attribute name="z" type="xsd:double" use="optional" default="0.0"/>  
      <xsd:attribute name="w" type="xsd:double"/>  
      <xsd:attribute name="h" type="xsd:double"/>  
      <xsd:attribute name="d" type="xsd:double" use="optional" default="0.0"/>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```

example:

```

.
.
.
<specie id="ATP" compartment="compartment" initialAmount="0">
.
.
.
<sl2:listOfSpeciesGRs>
  <sl2:speciesGR id="ATP_GR" ref="ATP" x="295.0" y="123.0" w="16.0"/>
  .
  .
  .
</sl2:listOfSpeciesGRs>
.
.
.

```

<Reaction> Layout Information

The reaction layout consists of two pseudo nodes. One that connects to the substrates and one that connects to the products of the reaction. Each pseudo node has a x, y and z coordinates which are named x1,y1,z1 and x2,y2,z2 respectively. The z1 and z2 coordinates are optional and they default to 0.0. We suggest connecting the two pseudo nodes by a straight line, but this is up to the application programmer. Additionally we have the **id** which identifies this graphical representation object and the **ref** attribute which is the id of the corresponding reaction in the model. Again, this reference is optional.

XML Schema representation:

```

<xsd:complexType name="ListOfSpeciesReferenceGRs">
  <xsd:complexContent>
    <xsd:extension base="sl2:SBase">
      <xsd:sequence>
        <xsd:element name="speciesReferenceGR" type="sl2:SpeciesReferenceGR"
          minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ReactionGR">
  <xsd:complexContent>
    <xsd:extension base="sl2:SBase">
      <xsd:sequence>
        <xsd:element name="listOfSpeciesReferenceGRs" type="sl2:ListOfSpeciesReferenceGRs"
          minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="ref" type="sl2:SId" use="optional"/>
      <xsd:attribute name="id" type="sl2:SId"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:attribute name="x1" type="xsd:double"/>
<xsd:attribute name="y1" type="xsd:double"/>
<xsd:attribute name="z1" type="xsd:double" use="optional" default="0.0"/>
<xsd:attribute name="x2" type="xsd:double"/>
<xsd:attribute name="y2" type="xsd:double"/>
<xsd:attribute name="z2" type="xsd:double" use="optional" default="0.0"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

example:

```

.
.
.
<reaction id="reaction_0" reversible="false">
  <listOfReactants>
    <specieReference specie="P_i" stoichiometry="1"/>
    .
    .
  </listOfReactants>
</reaction>
.
.
.
<sl2:listOfReactionGRs>
  <sl2:reactionGR id="reaction_0_GR" ref="reaction_0" x1="119.0" y1="166.0"
    x2="120.0" y2="183.0"/>
  .
  .
</sl2:listOfReactionGRs>
.
.
.

```

<SpecieReference> Layout Information

The layout information for `SpecieReference` objects is embedded in the reaction layout and consists of a reference (`refSpeciesGR`) to the corresponding species representation which has been defined in the `<listOfSpeciesGRs>` for the `Specie` object (see above). The reference is again optional. Since species references in sbml level 1 as well as level 2 do not have ids, we reference them by context. Since the layout information for a species reference is embedded in the layout information of the reaction to which the species reference belongs, we only need one additional piece of information to reference it and that is the role this species reference has in the reaction (i.e. if it is a product, modifier or substrate), we specify this through an attribute called `refRole` which can be one of substrate, product or modifier. Alternatively, if everybody could agree on this, we might

introduce an id tag into the annotations of species reference object in order to facilitate referencing them. This said, it should be clear that without unique ids for species references we can not display a reaction that references the same species twice as a product, substrate or modifier (and there should not be any need for it anyway). The speciesReferenceGR also has an **id** which is used to uniquely identify it. The **role** attribute is used to specify how the specie reference should be displayed. Allowed values are substrate, product, sidesubstrate, sideproduct, modifier, activator and inhibitor. substrate and product are used if the species reference is a main product or substrate in the reaction. sidesubstrate and sideproduct are used for stuff like ATP, NAD+ etc that is normally displayed as a side reaction. activator and inhibitor are modifiers where their influence on the reaction is known and modifier is a more general term if the influence is unknown or changes during the course of the simulation. This list is probably not exhaustive and will be updated as needed.

XML Schema representation:

```

<xsd:simpleType name="RoleString">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="substrate"/>
    <xsd:enumeration value="product"/>
    <xsd:enumeration value="sidesubstrate"/>
    <xsd:enumeration value="sideproduct"/>
    <xsd:enumeration value="modifier"/>
    <xsd:enumeration value="activator"/>
    <xsd:enumeration value="inhibitor"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="RefRoleString">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="substrate"/>
    <xsd:enumeration value="product"/>
    <xsd:enumeration value="modifier"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="SpeciesReferenceGR">
  <xsd:complexContent>
    <xsd:extension base="s12:SBase">
      <xsd:attribute name="id" type="s12:SId"/>
      <xsd:attribute name="refSpeciesGR" type="s12:SId" use="optional"/>
      <xsd:attribute name="refRole" type="s12:RefRoleString"/>
      <xsd:attribute name="role" type="s12:RoleString"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

example:

.

```

.
.
<sl2:speciesGR id="P_iGR" ref="P_i" x="63.0" y="124.0" w="16.0" h="16.0"/>
.
.
<sl2:speciesReferenceGR id="SP1_GR" refSpeciesGR="P_iGR"
                        refRole="reactant" role="reactant" />
.
.
.

```

Additional Graphical Representations

A lot of people require graphical representation for objects that have no correspondence in the model. For this purpose we added a `<listOfAdditionalGraphicalObjects>` which is exactly what the name suggests. The type and syntax of the objects that are allowed are yet to be defined. So far this list can hold objects of any type just like the annotations in sbml.

Further Plans

The extensions so far leave the rendering of the objects to the application. So each application has to come up with an extension of its own to store rendering information. In the future we hope to come up with a general way for specifying rendering information as well. This document is a work in progress and can be subject to changes any time. We are glad for any suggestions, corrections or hints to further improve these extensions. Hopefully with some help we could come up with a set of extensions that would suite the needs of many applications developed in this area. This specification was intended for the use with sbml level 2. With some slight changes it can also be used with sbml level 1 documents. Actually the only major changes that would have to be made is to change all references to SId to be references to SNames and limit roleRef in speciesReferenceGR to substrate and product.

Example File

Last but not least, we include a small sample file to illustrate and complement the paragraphs above. The model consists of two reactions. Which are the first reaction of glycolysis where glucose is converted to glucose-6-phosphate (G6P) and the reverse reaction of gluconeogenesis where glucose-6-phosphate is hydrolyzed to glucose. We did not include any coordinates in the third dimension, since we are only working in 2D space so far. As can be seen in the screenshot, the glucose SpecieReference has two representational objects on screen whereas glucose-6-phosphate only has one. This difference is reflected in the file where the glucose specie has two nodes in the listOfNodes whereas G6P only has one.

```
<?xml version="1.0" encoding="UTF-8"?>
```

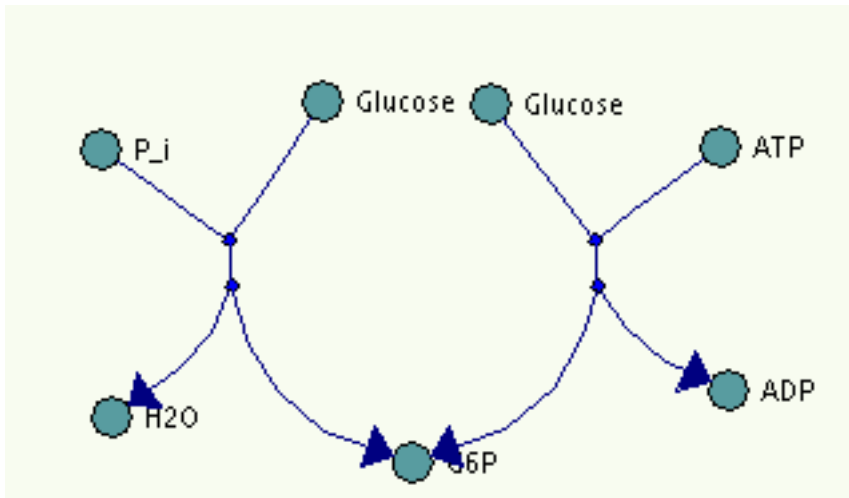


Figure 1: Partial screenshot from SimViz ReactionLayouter showing sample file.

```

<sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sl2="http://projects.embl.org/bcb/sbml/level2"
  xsi:schemaLocation="http://projects.embl.org/bcb/sbml/level2
    http://projects.embl.org/bcb/sbml/level2/layout2.xsd">
  <model name="Untitled">
    <listOfCompartments>
      <compartment id="compartment" volume="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <specie id="ATP" compartment="compartment" initialAmount="0"/>
      <specie id="P_i" compartment="compartment" initialAmount="0"/>
      <specie id="Glucose" compartment="compartment" initialAmount="0"/>
      <specie id="ADP" compartment="compartment" initialAmount="0"/>
      <specie id="H2O" compartment="compartment" initialAmount="0"/>
      <specie id="G6P" compartment="compartment" initialAmount="0"/>
    </listOfSpecies>
    <listOfReactions>
      <reaction id="reaction_0" reversible="false">
        <listOfReactants>
          <specieReference specie="P_i" stoichiometry="1"/>
          <specieReference specie="Glucose" stoichiometry="1"/>
        </listOfReactants>
        <listOfProducts>
          <specieReference specie="H2O" stoichiometry="1"/>
          <specieReference specie="G6P" stoichiometry="1"/>
        </listOfProducts>
      </reaction>
    </listOfReactions>
  </model>

```

```

<reaction id="reaction_1" reversible="false">
  <listOfReactants>
    <specieReference specie="ATP" stoichiometry="1"/>
    <specieReference specie="Glucose" stoichiometry="1"/>
  </listOfReactants>
  <listOfProducts>
    <specieReference specie="ADP" stoichiometry="1"/>
    <specieReference specie="G6P" stoichiometry="1"/>
  </listOfProducts>
</reaction>
</listOfReactions>
</model>
<sl2:listOfLayouts xmlns:sl2="http://projects.embl.org/bcb/sbml/level2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://projects.embl.org/bcb/sbml/level2\
    http://projects.embl.org/bcb/sbml/level2/layout2.xsd">
<sl2:layout id="layout1" >
  <sl2:listOfCompartmentGRs>
<sl2:compartmentGR id="compGR" ref="compartment" x="10.0" y="10.0" w="60" h="50"/>
</sl2:listOfCompartmentGRs>
<sl2:listOfSpeciesGRs>
<sl2:speciesGR id="ATP_GR" ref="ATP" x="295.0" y="123.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="P_iGR" ref="P_i" x="63.0" y="124.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="Glucose_GR1" ref="Glucose" x="146.0" y="106.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="Glucose_GR2" ref="Glucose" x="209.0" y="107.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="ADP_GR" ref="ADP" x="298.0" y="214.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="H2O_GR" ref="H2O" x="67.0" y="224.0"\
  w="16.0" h="16.0"/>
<sl2:speciesGR id="G6P_GR" ref="G6P" x="180.0" y="241.0"\
  w="15.0" h="16.0"/>
</sl2:listOfSpeciesGRs>
<sl2:listOfReactionGRs>
<sl2:reactionGR id="reaction_0_GR" ref="reaction_0" x1="119.0" y1="166.0"\
  x2="120.0" y2="183.0">
  <sl2:listOfSpeciesReferenceGRs>
<sl2:speciesReferenceGR id="SP1_GR" refSpeciesGR="P_iGR"\
  refRole="substrate" role="substrate" />
<sl2:speciesReferenceGR id="SP2_GR" refSpeciesGR="Glucose_GR1"\
  refRole="substrate" role="substrate" />
<sl2:speciesReferenceGR id="SP3_GR" refSpeciesGR="H2O_GR"\
  refRole="product" role="product" />
<sl2:speciesReferenceGR id="SP4_GR" refSpeciesGR="G6P_GR"\
  refRole="product" role="product" />
</sl2:listOfSpeciesReferenceGRs>

```

```

</sl2:reactionGR>
<sl2:reactionGR id="reaction_1_GR" ref="reaction_1" x1="256.0" y1="166.0"\
                x2="257.0" y2="183.0">
  <sl2:listOfSpeciesReferenceGRs>
    <sl2:speciesReferenceGR id="SP5_GR" refSpeciesGR="ATP_GR"\
                          refRole="substrate" role="substrate" />
    <sl2:speciesReferenceGR id="SP6_GR" refSpeciesGR="Glucose_GR2"\
                          refRole="substrate" role="substrate" />
    <sl2:speciesReferenceGR id="SP7_GR" refSpeciesGR="ADP_GR"\
                          refRole="product" role="product" />
    <sl2:speciesReferenceGR id="SP8_GR" refSpeciesGR="G6P_GR"\
                          refRole="product" role="product" />
  </sl2:listOfSpeciesReferenceGRs>
</sl2:reactionGR>
</sl2:listOfReactionGRs>
<sl2:listOfAdditionalGraphicalObjects>
<textLabel id="label1" x="100.0" y="85.0" w="180.0" h="25.0">\
  Reaction Layout\
</textLabel>
</sl2:listOfAdditionalGraphicalObjects>
</sl2:layout>
</sl2:listOfLayouts>
</sbml>

```

Contact Information

To contact any of the authors, send an email to:
FIRSTNAME.LASTNAME@eml.villa-bosch.de
 e.g.
 ralph.gauges@eml.villa-bosch.de

References

- [1] System Biology Markup Language Level 1 Website
 (<http://www.sbml.org/sbml/docs/index.html>)
- [2] Michael Hucka, Andrew Finney, Herbert Sauro, Hamid Bolouri: Systems Biology Markup Language (SBML) Level 1: Structures and Facilities for Basic Model Definitions (<http://www.sbml.org/sbml/docs/papers/sbml-level-1/html/sbml-level-1.html>)
- [3] Hucka M., Finney A., Sauro H.M., Bolouri H., Doyle J.C., Kitano H., Arkin A.P., Bornstein B.J., Bray D., Cornish-Bowden A., Cuellar A.A., Dronov S., Gilles E.D., Ginkel M., Gor V., Goryanin I.I., Hedley W.J., Hodgman T.C., Hofmeyr J.H., Hunter P.J., Juty N.S., Kasberger J.L., Kremling A., Kummer U., Le Novere N., Loew L.M., Lucio D., Mendes P., Minch E., Mjolsness E.D., Nakayama Y., Nelson M.R., Nielsen P.F., Sakurada T., Schaff J.C., Shapiro B.E., Shimizu T.S., Spence H.D., Stelling J., Takahashi K., Tomita M., Wagner J., Wang J. (2003) The systems

biology markup language (SBML): a medium for representation and exchange of biochemical network models, *Bioinformatics*, **19**, 524-31.

[4] JDesigner Website (<http://www.cds.caltech.edu/hsauro/JDesigner.htm>)

[5] Herbert Sauro: JDesigner SBML Annotation (<http://www.cds.caltech.edu/hsauro/JDSBMLEx.pdf>)