Cell System Markup Language – CSML

Language Specification v. 1.9

Version 1.0
August 31, 2005

1 About This Document

1.1 Scope

This specification defines the Cell System Markup Language (CSML), the language and data format used for description of biological systems with Hybrid Functional Petri Nets with extensions.

1.2 Status

This specification defines the 1.9 version of CSML.

1.3 How This Document Is Organized

- **Section 2** gives an introductory description of the CSML format, showing how it relates to biological system models.
- **Section 3** gives detailed specification of CSML tags.
- **Section 4** lists publication references relevant to this document.
- **Section 5** presents example models and their respective CSML representation.

1.4 Document Conventions

In this document, the following conventions are used:

Convention	Meaning
`code sample`	Courier New typeface indicates CSML code, CSML tag name, attribute name, or attribute value.
`code sample`	Courier New typeface in a frame indicates an example of CSML code.
. . .	In code examples, an ellipsis means that not all of the code is shown – either tags or tag attributes were omitted for readability.

In section 3, "CSML Reference", description for each of the CSML tags is broken into following parts:

- general description;
- table of attributes with respective descriptions;
- table of inner tags with respective descriptions;
- code example.

2 Key CSML Concepts

2.1 What Is CSML?

The Cell System Markup Language (CSML) is the language and data format used for description of biological systems with Hybrid Functional Petri Nets with extensions. CSML provide constructing nets modeling metabolic pathways, signal transduction cascades, gene regulatory pathways as well as dynamic interactions of various biological entities such as genomic DNA, mRNA and proteins.

CSML is used to represent biological system models using hybrid functional Petri nets. CSML provides a way to describe both the Petri net structure of the model and its graphical representation. CSML uses Extensible Markup Language (XML) format.

The website of CSML Project is: http://www.csml.org/.

2.2 Modeling Biological Systems with Hybrid Functional Petri Nets

In the post-genome era, biopathways information processing is one of the most important research topics in Bioinformatics. From a biomedical viewpoint, research progress will be very beneficial to human beings, e.g. developments of medicine manufacture and effective medical treatment for each patient.

In the HFPN hybrid systems, continuous and discrete events can be modeled with functional enhancement for processes, which enables to model various interactions and reactions. In addiction, complex objects can be modeled with the enhancement of hierarchization, which enables easy modeling for large-scale pathways. The HFPNe is an extension of the HFPN architecture for easy modeling/simulation of more complex biological information such as localization or multicellular processes.

2.3 What's new in CSML version 1.9?

- new kinetic styles for easier simulation of biological processes

  The new kinetic styles of a process: *mass, stochastic mass* and *connector rate* CSML provides building models more easily and define the kinetics of chemical reactions and biological processes.

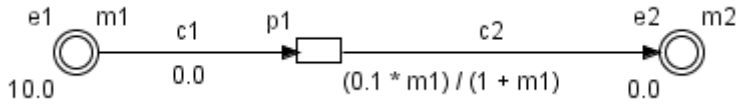
- support for SVG files

  Scalable Vector Graphics (SVG) files can be added to the models as any other image files (PNG, JPEG). All image files, in the PNG, JPEG and now also SVG format can be saved internally or externally in the CSML files.

2.4 Model Representation

For the purpose of description of how a model is represented in CSML, let us consider an example of a simple irreversible Michaelis-Menten type reaction.

The model shown below consists of two continuous entities e1 and e2, and the continuous process between them. The entity e1 is connected to p1 with the input connector c1. The process p1 is connected to e2 with the output connector c2. The entities e1 and e2 have initial values of 10 and 0, respectively. The input connector c1 has the threshold parameter equal zero.



In this model the consumption ratio of e1, or the speed of the process p1, is described by the following formula:

$$-\frac{dm_1}{dt} = \frac{dm_2}{dt} = \frac{0.1m_1}{1+m_1}$$

2.4.1 General Structure

At the top level, the CSML code for this example model looks like the excerpt shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
      xmlns:csml="http://www.csml.org/csml/version1"
      majorVersion="1" minorVersion="9">
 <csml:unitdefs>
   ...
 </csml:unitdefs>
 <csml:net>
  <csml:entity label="e1" name="e1" type="continuous">
   ...
  </csml:entity>
  <csml:entity label="e2" name="e2" type="continuous">
   ...
  </csml:entity>
  <csml:process label="p1" name="p1" type="continuous">
   ...
  </csml:process>
 </csml:net>
 <csml:simulation ... />
 <csml:charts>
  ...
 </csml:charts>
</csml:model>
```

The top-level element of a CSML document is model. The model description is contained in the net element, which contains the descriptions of the entities and of the process. The connectors are described inside the process element, as will be shown further below.
The details of the inner tags and attributes were omitted, and will be presented next.

2.4.2 Entities

Going deeper into the CSML code, the representation for the entities e1 and e2 is shown below:

```
<csml:entity label="e1" name="e1" type="continuous">
 <csml:parameter label="m1" type="Double" initialValue="10"
   minimumValue="0" maximumValue="infinite"/>
 <csml:graphics>
  <csml:figure>
   <csml:continuousEntity location="154.0 154.25"/>
  </csml:figure>
 </csml:graphics>
</csml:entity>
<csml:entity label="e2" name="e2" type="continuous">
 <csml:parameter label="m2" type="Double" initialValue="0"
   minimumValue="0" maximumValue="infinite"/>
 <csml:graphics>
  <csml:figure>
   <csml:continuousEntity location="465.0 154.25"/>
  </csml:figure>
 </csml:graphics>
</csml:entity>
```

The variables associated with the entities are described by the parameter tags. These tags define variable names, their initial values, and possible value ranges.

In addition to the tags representing the logical structure, tags for graphical representation of the entities are included. All graphical element definitions are contained in graphics tags. A set of standard graphical elements for entities is available (continuousEntity, discreteEntity, and genericEntity), but any other of the elements listed in 3.74.3 can be used, with the exception of text and textArea.

2.4.3 Processes and Connectors

The CSML representation of the process p1 and connectors c1 and c2 is shown below:

```
<csml:process label="p1" name="p1" type="continuous">
 <csml:simulationCondition>
  <csml:priority value="0"/>
  <csml:firing type="Boolean" value="true" firingStyle="and"/>
  <csml:delay type="Long" value="0.0" delayStyle="nodelay"/>
  <csml:calc calcStyle="speed"/>
  <csml:kinetic type="Double" kineticStyle="custom">
   <csml:parameter name="custom" value="(0.1*m1)/(1+m1)"/>
  </csml:kinetic>
 </csml:simulationCondition>
 <csml:function>
  <csml:connector label="c1" name="c1" type="process" from="e1"
   to="p1" linestyle="straight" supportpath="true">
   <csml:firing type="Double" value="0"
   connectorFiringStyle="threshold"/>
   <csml:kinetic/>
   <csml:graphics>
    ...
   </csml:graphics>
  </csml:connector>
  <csml:connector label="c2" name="c2" type="process" from="p1"
   to="e2" linestyle="straight" supportpath="true">
   <csml:firing type="Double" value="0"
   connectorFiringStyle="threshold"/>
   <csml:kinetic/>
   <csml:graphics>
```

```
      ...
     </csml:graphics>
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    ...
   </csml:graphics>
  </csml:process>
```

The kinetic of the process is defined by the kinetic element, using a script.
The definition of the connectors between this process and the entities in the model are in the function element. Each connector, specified with a connector element, has several parameters, such as firing, kinetic, etc.
Just as with the entities previously, there are elements for graphical representation (only the tags for the process image are shown in their full extent). Below is an excerpt showing how a connector's graphic representation is notated:

```
    <csml:connector label="c1" name="c1" type="process" from="e1"
     to="p1" linestyle="straight" supportpath="true">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="176.0 165.0 303.0 165.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="303.0 162.0 303.0 168.0
           311.0 165.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
    <csml:connector label="c2" name="c2" type="process" from="p1"
     to="e2" linestyle="straight" supportpath="true">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="333.0 165.0 457.0 165.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="457.0 162.0 457.0 168.0
           465.0 165.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
```

In this example, a connector line and an end arrow are defined. Connector arrows are an exception – they are contained in sourceArrow and targetArrow elements instead of figure.

2.5 Simulation Settings

All simulation settings are specified by the attributes of the simulation tag:

```
<csml:simulation enhancedFiring="true" samplingInterval="0.1"
 simulationTime="200.0" logUpdateInterval="1.0"
 plotUpdateInterval="1.0"/>
```

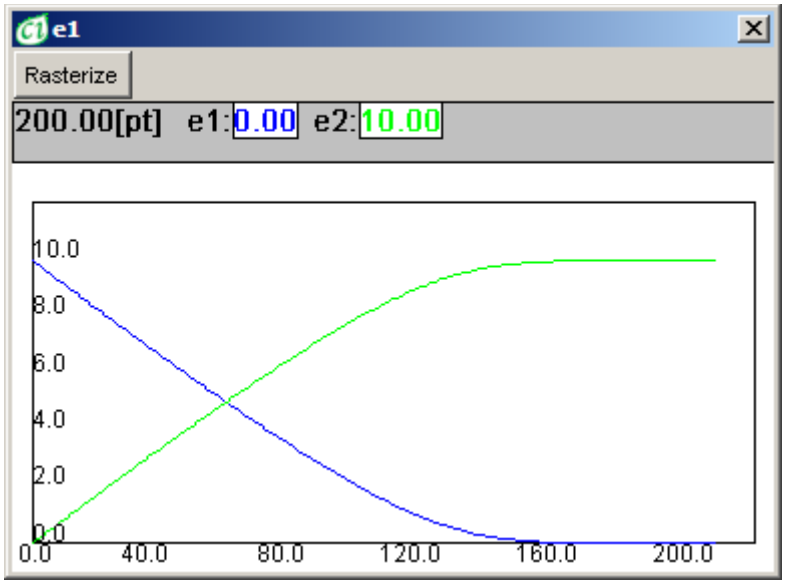
All time and interval parameters defined by the attributes are in Petri time units.
The sampling interval controls the accuracy of the simulation, or how fine-grained are the variable value updates during the simulation.
The simulation time controls how long the simulation should run. The `simulationTime`/`samplingInterval` ratio gives the total number of steps in the simulation.
The plot update interval controls how often the charts (see 2.6) should be updated if a given simulation application provides chart visualization.
The log update interval controls how often the values in the model should be logged (see 2.10).
The `enhancedFiring` attribute is a legacy feature and its value is always "true".

2.6 Charts

CSML allows for defining simulation charts for a model. Using the above example model, let us consider a chart for the values of entities e1 and e2, defined by the following code:

```
<csml:charts>
 <csml:chart name="e1">
  <csml:property key="width" value="388"/>
  <csml:property key="height" value="288"/>
  <csml:property key="y" value="126"/>
  <csml:property key="x" value="567"/>
  <csml:entity labelRef="e1"/>
  <csml:entity labelRef="e2"/>
 </csml:chart>
</csml:charts>
```

The tags used here to reference the entities whose variable values are plotted on the chart are of a different type (chart/entity) than that used to define the entities in the model (entity). Any number of entity variables can be plotted on a single chart.
The property tags define the absolute location on screen and the size of the chart window. There is one tag for each of the four values.
The picture below shows what a possible rendering of this chart for the example model could look like:



2.7 Mathematical Expressions and Scripting

CSML provides the ability of scripting various elements of the model. The meaning of "scripting" in CSML is two-fold:

- A numeric value in a model may be specified using an expression instead of a fixed value;
- Interaction of a model element with other elements may be specified using a set of expressions.

The standard language used for scripting in CSML is Pnuts. Other languages could be allowed in future versions of CSML.

The following numeric values can be specified with expressions:

- Entity initial value (see parameter),
- Process activity (see firing),
- Process kinetic (see kinetic),
- Process delay (see delay),
- Connector kinetic (see kinetic),
- Connector threshold (see firing).

In the example model from 2.4, the process kinetic is defined as an expression. The respective fragment of CSML code is shown below.

```
   <csml:process label="p1" name="p1" type="continuous">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" value="true" firingStyle="and"/>
    <csml:delay type="Long" value="0.0" delayStyle="nodelay"/>
    <csml:calc calcStyle="speed"/>
    <csml:kinetic type="Double" kineticStyle="custom">
     <csml:parameter name="custom" value="(0.1*m1)/(1+m1)"/>
    </csml:kinetic>
   </csml:simulationCondition>
   ...
  </csml:process>
```

The kinetic tag contains an inner */parameter tag having the `value` attribute containing the expression.

2.8 Model Hierarchy

CSML allows for models to be imported into other models and used as integral elements. To be usable from other models, a given model must have at least one public entity, i.e. one whose associated variable value is visible from an importing model.
In the example below, the model contains an entity e1, a process p1, and an imported model o1. The process is connected from entity e2 in imported model o1 and to e1. The entity e2 in o1 is made public by the use of a public tag. Note how the reference to the public entity is noted in the connector as "`o1 e2`".

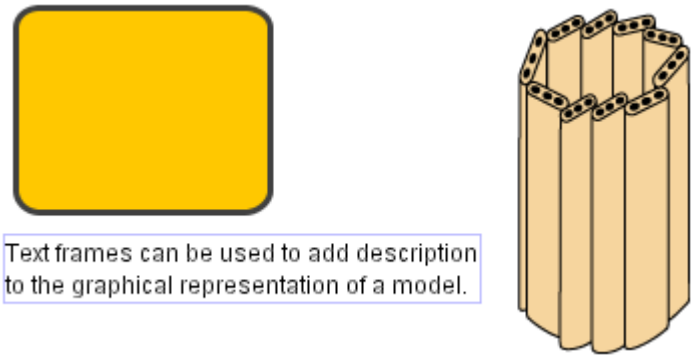
```
<csml:net>
  <csml:entity label="e1" name="e1" type="continuous">
   ...
  </csml:entity>
  <csml:process label="p1" name="p1" type="continuous">
   ...
   <csml:function>
    <csml:connector label="c1" name="c1" type="normal" from="o1 e2"
     to="p1">
     ...
    </csml:connector>
    <csml:connector label="c2" name="c2" type="normal" from="p1"
     to="e1">
     ...
    </csml:connector>
    ...
   </csml:function>
   ...
  </csml:process>
  <csml:object label="o1" name="h1" referenceType="fileType"
     reference="simple_export.gon">
   <csml:entityProxy type="continuous" reference="e2">
    <csml:parameter label="m2" initialValue="0.0"
     minimumValue="0.0" maximumValue="infinite"/>
    <csml:public value="true"/>
    <csml:graphics>
     ...
    </csml:graphics>
   </csml:entityProxy>
   <csml:graphics>
    <csml:figure>
     <csml:objectFigure location="100.0 60.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:object>
 </csml:net>
```

The imported model is contained in the object element. The entities are defined using entityProxy elements instead of entity. In this example, the model o1 is imported by file reference; only the entity e2 which is public is defined inside the object element. The entire imported model has a graphical representation as a single figure.

2.9 Visualization Enhancements

In addition to the model structure, graphical elements can be used to enhance the visual representation of a given model. The most typical use of graphical elements is to specify how different elements of a given model should be displayed in a visualization

application – the graphical symbols and their location. Purely illustrational elements can also be added to a CSML model.
Consider a CSML file with only graphical elements – with a rectangle, a text frame, and an external image (graphic file) like on the picture below:



It is represented with the following CSML code:

```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
 xmlns:csml="http://www.csml.org/csml/version1" majorVersion="1"
     minorVersion="9">
 <csml:unitdefs/>
 <csml:net>
  <csml:figure>
   <csml:roundBounds location="13.0 10.0" size="127.0 102.0"
      fillColor="255 200 0 255" outlineColor="64 64 64 255"
      outlineWidth="3.0" arcRatio="20"/>
  </csml:figure>
  <csml:figure>
   <csml:textArea location="7.0 123.0" size="238.0
      34.0"><![CDATA[Text frames can be used to add description
      to the graphical representation of a model.]]>
   </csml:textArea>
  </csml:figure>
  <csml:figure>
   <csml:image location="264.5 11.5" size="86.0 172.0"
      filePath="centrosome-1.png" fileType="localFile"
      fileFormat="img"/>
  </csml:figure>
 </csml:net>
 <csml:simulation enhancedFiring="true" samplingInterval="0.1"
     simulationTime="2000.0" logUpdateInterval="1.0"
     plotUpdateInterval="1.0"/>
</csml:model>
```

Every graphical element is represented with a specific tag inside a figure tag. The figure tags are contained directly inside the net tag.
Graphical element tags are listed in 3.75.

2.10 Simulation Log

CSML allows for the results of simulation to be logged into a given CSML document. Values of entity variables and of process speeds can be logged. Those values should be logged at specific points in time, specified by the log update interval (see 2.5). For brevity, repeating values can be omitted in the log.
The example below shows an excerpt of a simulation log.

```
<csml:log>
 <csml:logItem filename="sim12.cil" status="finished"
    timeStamp="Dec    2, 2004 2:06:39 PM" server="localhost">
  <csml:comment><![CDATA[test]]>
  </csml:comment>
 </csml:logItem>
 <csml:logVariableList>
  <csml:variable labelRef="e2" logIndex="0" elementType="entity"
    propertyType="value"/>
  <csml:variable labelRef="e1" logIndex="1" elementType="entity"
    propertyType="value"/>
 </csml:logVariableList>
 <csml:logDataList>
  <csml:logData timePoint="0">
   <csml:logValue id="0" value="0.0"/>
   <csml:logValue id="1" value="10.0"/>
  </csml:logData>
  ...
  <csml:logData timePoint="1999000000"/>
  <csml:logData timePoint="2000000000"/>
 </csml:logDataList>
</csml:log>
```

The logItem element contains basic information on the simulation: its status (whether it has finished, was interrupted, etc.), its end date and time, etc. A user comment on a particular simulation run can also be included.
The logVariableList element contains a list of values in the model that were logged. “Column index” for each of the values is also specified.
The logDataList element contains a list of simulation time points, each of which contains the logged values. As said before, when a given value doesn’t change from one time point to the next one, it can be omitted in the log, as is the case with the last two time points in the example.

3 CSML Reference

This chapter describes all tags used in the CSML format. For each of the tags the following information is given:

- Parent tag or tags;
- The purpose of a given tag;
- The attributes of a given tag and their respective meaning;
- The structure of a given tag – a table listing of the tags that appear inside the one being described (only the first level);
- An example of use – a code sample snippet.

At the end of the chapter, in sections 3.74–3.78, common features of various groups of tags are specified in detail.

3.1 The CSML Namespace URI

The namespace URI for CSML elements is "`http://www.csml.org/csml/version1`".

3.2 `associationConnector`

Parent tag: figure

The `associationConnector` tag defines the graphical representation of an association connector.

3.2.1 Attributes

See 3.75.1; the `points` attribute contains pairs of *x*-*y* coordinates of points that define the connector's poly-line.

3.2.2 Structure

See 3.75.2

3.2.3 Example

```
<csml:figure>
 <csml:associationConnector points="235.0 218.0 207.0 245.5
     236.44385982982155 304.8338387479737"/>
</csml:figure>
```

3.3 `associationConnectorArrow`

Parent tag: sourceArrow, targetArrow

The `associationConnectorArrow` tag defines the graphical representation of a target arrow of an association connector.

3.3.1 Attributes

See 3.75.1

3.3.2 Structure

See 3.75.2

3.3.3 Example

```
      <csml:targetArrow>
       <csml:associationConnectorArrow points="0.0 0.0 0.0 6.0 8.0
           3.0" pointsInAbsoluteCoordinate="223.13117029933142
           807.5002861841567 217.75654936031168 810.1673913117905
           224.00000000000003 815.9999999999999"/>
      </csml:targetArrow>
```

3.4 bioInhibitoryConnectorArrow

Parent tag: sourceArrow, targetArrow

The `bioInhibitoryConnectorArrow` tag defines the end arrow of an inhibitory connector's graphical representation. This tag corresponds to the traditional "biological community" style of the arrow; see also 3.35.

3.4.1 Attributes

See 3.75.1

3.4.2 Structure

See 3.75.2

3.4.3 Example

```
      <csml:targetArrow>
       <csml:bioInhibitoryConnectorArrow location="0.0 0.0"
           pointsInAbsoluteCoordinate="269.0 245.0 269.0 259.0
277.0            259.0 277.0 245.0"/>
      </csml:targetArrow>
```

3.5 biological

Parent tag: process

The `biological` tag is a reserved tag for future CSML format. Sub elements in the tag will be used for describing detailed biological roles e.g. ligand, receptor and enzyme, and storing other biological database access ID, e.g. unigeneID.

3.5.1 Attributes

The `biological` tag has no attributes.

3.5.2 Structure

The `biological` tag contains the following inner tag:

Name	Details
`effectList`	Biological role in the pathway, e.g. ligand, receptor.

3.5.3 Example

```
<csml:biological>
 <csml:effectList/>
</csml:biological>
```

3.6 calc

Parent tag: process

The `calc` tag defines a style of calculations in process or connector / production or consumption of the entities connected to process.

3.6.1 Attributes

The `calc` tag has the following attributes:

Name	Details
`calcStyle`	Style of calculations in process, possible values: `speed`, `add`, `update`.

3.6.2 Structure

The tag has no inner tags.

3.6.3 Example

```
<csml:calc calcStyle=”speed"/>
```

3.7 chart

Parent tag: charts

The `chart` tag specifies a simulation chart for a model. The chart specification includes geometrical location of the chart on the screen and a list of entities whose variable values are to be drawn on the chart.

3.7.1 Attributes

The `chart` tag has the following attributes:

Name	Details
`name`	Text that can be used for labeling the chart in simulation visualization.

3.7.2 Structure

The `chart` tag contains the following inner tags:

Name	Occurrence	Details
`chart/entity`	any number	References a model entity whose value will be presented on the chart.

property	any number	Specifies a parameter of geometrical location of the chart on the screen. There must be 4 property tags for a chart, one for each of the geometrical location parameters (x, y, width, or height).

3.7.3 Example

```
<csml:chart name="e1">
 <csml:property key="width" value="388"/>
 <csml:property key="height" value="288"/>
 <csml:property key="y" value="126"/>
 <csml:property key="x" value="567"/>
 <csml:entity labelRef="e1"/>
 <csml:entity labelRef="e2"/>
</csml:chart>
```

3.8 charts

Parent tag: model

The charts tag contains a list of all simulation charts for a model.

3.8.1 Attributes

The charts tag has no attributes.

3.8.2 Structure

The charts tag contains the following inner tags:

Name	Occurrence	Details
chart	any number	Specifies the details of a single simulation chart.

3.8.3 Example

```
<csml:charts>
 <csml:chart name="e1">
  ...
 </csml:chart>
 <csml:chart name="e4">
  ...
 </csml:chart>
</csml:charts>
```

3.9 class

Parent tag: model

The class tag contains an imported, internalized model. To be used in a model, an internalized model must be referenced in an object tag by its label attribute.

3.9.1 Attributes

The class tag has the following attributes:

Name	Details
label	Unique identifier of the internalized model, by which it is referenced in an object tag.

3.9.2 Structure

The `class` tag contains the same inner tags as a net tag.

3.9.3 Example

```
<csml:class label="internalized1">
 <csml:entity label="e1" name="e1" type="continuous">
  ...
 </csml:entity>
 <csml:entity label="e2" name="e2" type="continuous">
  ...
 </csml:entity>
 <csml:process label="p1" name="p1" type="continuous">
  <csml:simulationCondition>
   ...
  </csml:simulationCondition>
  <csml:function>
   ...
  </csml:function>
  <csml:graphics>
   ...
  </csml:graphics>
 </csml:process>
</csml:class>
```

3.10 comment

Parent tag: connector, entity, entityProxy, logData, object, process

The `comment` tag contains a comment for the item associated with its parent tag.

3.10.1Attributes

The `comment` tag has no attributes.

3.10.2Structure

The `comment` tag has no inner tags. The comment text is contained as text or in a CDATA section.

3.10.3Example

```
   <csml:comment><![CDATA[test]]>
   </csml:comment>
```

3.11 connector

Parent tag: function

The `connector` tag describes a connector between an entity and a process. The direction of a connector can be from an entity to a process or from a process to an entity.

3.11.1Attributes

The `connector` tag has the following attributes:

Name	Details
`label`	Unique identifier. Not reference elsewhere in the model.
`name`	Text that can be used for labeling the connector in model visualiza-

	tion.
type	Possible values: normal, test, inhibitor.
from	The label attribute of the entity or process from which the connector goes out.
to	The label attribute of the entity or process to which the connector goes in.
linestyle	If "straight", the connector should be represented by a straight or angled line. If "curve", a Bézier curve should be displayed.
supportpath	If true and the linestyle attribute is "curve", the "curve hints" for the connector Bézier curve should be displayed.

3.11.2 Structure

The connector tag contains the following inner tags:

Name	Occurrence	Details
firing	optional	The definition of firing style.
kinetic	optional	The definition of kinetic style.
graphics	optional	Graphical representation of the connector.
comment	optional	Comment for the connector.
references	optional	List of URL sites related to the connector.

3.11.3 Example

```
<csml:connector label="c1" name="c1" type="inhibitor" from="e1"
            to="p3" linestyle="curve" supportpath="true">
 <csml:firing type="Boolean" connectorFiringStyle="rule">
  <csml:script><![CDATA[return true;]]>
  </csml:script>
 </csml:firing>
 <csml:kinetic>
  <csml:parameter name="stoichiometry" value="1"/>
 </csml:kinetic>
 <csml:graphics>
  <csml:figure>
   <csml:inhibitoryConnector points="133.0 89.0
       251.6657835397849 55.52812288473106 307.6935659005844
       265.958724897353 169.96436409782143
       284.4672549391083"/>
  </csml:figure>
  <csml:text name="firingStyle" location="327.60767382790107
            180.8625660807629" fillColor="255 255 0 255"
                  arrangement="None" offsets="57.0 20.0"/>
  <csml:text name="firingOperation"
       location="216.60767382790107
       166.8625660807629" fillColor="255 255 0 255"
       arrangement="None" offsets="-35.0 6.0"/>
  <csml:text name="name" location="303.60767382790107
       124.86256608076289" fillColor="255 255 0 255"
       arrangement="None" offsets="29.0 -36.0"/>
  <csml:targetArrow>
   <csml:bioInhibitoryConnectorArrow location="0.0 0.0"
            pointsInAbsoluteCoordinate="170.89666795438197
                  291.40489211029575 169.03206024126087
                  277.5296177679208 165.06769614343946
```

```
                    278.0623628288125 166.93230385656054
          291.9376371711875"/>
     </csml:targetArrow>
    </csml:graphics>
   </csml:connector>
```

3.12 continuousEntity

Parent tag: figure

The `continuousEntity` tag defines the graphical representation of a continuous entity.

3.12.1Attributes

See 3.75.1

3.12.2Structure

See 3.75.2

3.12.3Example

```
   <csml:figure>
    <csml:continuousEntity location="103.0 117.0"/>
   </csml:figure>
```

3.13 continuousProcess

Parent tag: figure

The `continuousProcess` tag defines the graphical representation of a continuous process.

3.13.1Attributes

See 3.75.1

3.13.2Structure

See 3.75.2

3.13.3Example

```
   <csml:figure>
    <csml:continuousProcess location="224.5 121.5"/>
   </csml:figure>
```

3.14 cross

Parent tag: figure

The `cross` tag defines an X-cross figure in the graphical representation of the model.

3.14.1Attributes

See 3.75.1

3.14.2Structure

See 3.75.2

3.14.3Example

```
<csml:figure>
 <csml:cross location="590.0 158.0"/>
</csml:figure>
```

3.15 delay

Parent tag: process

The `delay` tag defines the firing delay for discrete and generic processes (for these processes the delay style value is always "`delay`"). For continuous processes the delay style has always "`nodelay`" value. A discrete process will fire just after its delay time if the process is still active. Otherwise, it can lose its chance to fire.

3.15.1Attributes

The `delay` tag has the following attributes:

Name	Details
`value`	the delay value - non-negative Double value
`delayStyle`	the delay style, possible values: "`delay`" (for generic and discrete processes) and "`nodelay`" (for continuous processes)

3.15.2Structure

The `delay` tag is scriptable: See 3.74.2; the inner `script` tag contains a function returning non-negative double values.

3.15.3Example

```
<csml:delay value="0.0" delayStyle="nodelay"/>
```

3.16 discreteEntity

Parent tag: figure

The `discreteEntity` tag defines the graphical representation of a discrete entity.

3.16.1Attributes

See 3.75.1

3.16.2Structure

See 3.75.2

3.16.3Example

```
<csml:figure>
 <csml:discreteEntity location="232.0 199.0"/>
</csml:figure>
```

3.17 discreteProcess

Parent tag: figure

The `discreteProcess` tag defines the graphical representation of a discrete process.

3.17.1 Attributes

See 3.75.1

3.17.2 Structure

See 3.75.2

3.17.3 Example

```
<csml:figure>
 <csml:discreteProcess location="231.5 312.0"/>
</csml:figure>
```

3.18 effectList

Parent tag: biological

The `effectList` is reserved for future CSML format.

3.18.1 Attributes

The `effectList` tag has no attributes.

3.18.2 Structure

The `effectList` tag has no inner tags.

3.18.3 Example

```
<csml:effectList/>
```

3.19 ellipse

Parent tag: figure

The `ellipse` tag defines an ellipse in the graphical representation of the model.

3.19.1 Attributes

See 3.75.1

3.19.2 Structure

See 3.75.2

3.19.3 Example

```
<csml:figure>
 <csml:ellipse location="590.0 158.0" size="24.0 48.0"/>
</csml:figure>
```

3.20 ellipseArrow

Parent tag: sourceArrow, targetArrow

The `ellipseArrow` tag represents an ellipse used for the start or end arrow in the graphical representation of a process connector.

3.20.1 Attributes

See 3.75.1

3.20.2 Structure

See 3.75.2

3.20.3 Example

```
        <csml:ellipseArrow location="0.0 0.0"
pointsInAbsoluteCoordinate="333.0 278.0 333.0 310.0 365.0 310.0
365.0 278.0"/>
```

3.21 entity

Parent tag: net

> **Note:** There is also an `entity` tag with a different meaning and structure (see 3.21) that can appear inside a `chart` tag.

The `entity` tag describes a Petri net entity in a model. The properties defined by this tag's attributes and inner tags are the type (continuous, discrete, or generic), the associated variable, initialization and updating of the associated variable, graphical representation of the entity, etc.

3.21.1 Attributes

The `entity` tag has the following attributes:

Name	Details
`label`	Unique identifier by which a given `entity` tag is referenced in other tags, such as `connector`.
`name`	Text that can be used for labeling the entity in model visualization.
`type`	Type of entity – continuous, discrete, or generic. Possible values: `continuous`, `discrete`, `generic`.

3.21.2 Structure

The `entity` tag contains the following inner tags:

Name	Occurrence	Details
`comment`	optional	Comment for the entity.
`references`	optional	List of URL sites related to the entity.
`parameter`	optional	Details of the variable associated with this entity.
`public`	optional	If true, this entity is visible for importing model. Assumed false if not present.
`logging`	optional	If true, the value of the associated variable will be logged during simulation.

evaluateScriptOnlyForInitialization	optional	The value of this tag is ignored if the initialValue attribute of the parameter tag is a fixed value. Otherwise, if the initialValue is a script, there are 2 possibilities: 1) If it is true, the value of the associated variable is initialized with value set to the script at the beginning of simulation. 2) If it is false, the value of the associated variable is set by evaluating the script at each simulation step.
graphics	optional	Graphical representation of the entity.

3.21.3Example

```
<csml:entity label="e2" name="e2" type="continuous">
 <csml:parameter label="m2" type="Double" initialValue="0"
   minimumValue="0" maximumValue="infinite"/>
 <csml:graphics>
  <csml:figure>
   <csml:continuousEntity location="465.0 154.25"/>
  </csml:figure>
 </csml:graphics>
</csml:entity>
```

3.22 chart/entity

Parent tag: chart

> **Note:** There is also an entity tag with a different meaning and structure (see 3.21) that can appear inside a net tag.

The entity tag contained in a chart tag defines that a Petri net entity (see 3.21) specified by the labelRef attribute should appear on the chart defined by the parent chart tag.

3.22.1Attributes

The entity tag has the following attributes:

Name	Details
labelRef	The value of the label of the Petri net entity (see 3.21) referenced by this tag.

3.22.2Structure

The entity tag has no inner tags.

3.22.3Example

```
<csml:entity labelRef="e1"/>
<csml:entity labelRef="e2"/>
```

3.23 entityProxy

Parent tag: object

The `entityProxy` tag describes a reference to an entity that is defined in an external CSML document.

3.23.1 Attributes

The `entityProxy` tag has the following attributes:

Name	Details
`type`	Type of entity – continuous, discrete, or generic. Possible values: `continuous`, `discrete`, `generic`.
`reference`	The value of the `label` attribute of the referenced entity.

3.23.2 Structure

The `entityProxy` tag contains the following inner tags:

Name	Occurrence	Details
`comment`	optional	Comment.
`references`	optional	List of URL sites related to the entity.
`parameter`	optional	Details of the variable associated with this entity.
`public`	optional	Always present and always true, because an entity must be public to be imported into another model.
`logging`	optional	If true, the value of the associated variable will be logged during simulation.
`evaluateScriptOnlyForInitialization`	optional	The value of this tag is ignored if the `initialValue` attribute of the `parameter` tag is a fixed value. Otherwise, if the `initialValue` is a script, there are 2 possibilities: 1) If it is true, the value of the associated variable is initialized with value set to the script at the beginning of simulation. 2) If it is false, the value of the associated variable is set by evaluating the script at each simulation step.
`graphics`	optional	Graphical representation of the entity.

3.23.3 Example

```
<csml:entityProxy type="continuous" reference="e2">
 <csml:parameter label="m2" initialValue="0.0"
  minimumValue="0.0" maximumValue="infinite"/>
 <csml:public value="true"/>
 <csml:graphics>
  <csml:figure>
   <csml:continuousEntity location="78.0 73.0"/>
  </csml:figure>
 </csml:graphics>
</csml:entityProxy>
```

3.24 evaluateScriptOnlyForInitialization

Parent tag: entity, entityProxy

The `evaluateScriptOnlyForInitialization` tag specifies whether the value associated with an entity should be evaluated at the beginning of simulation or at each simulation step. The value of this tag is ignored if the `initialValue` attribute of the `parameter` tag associated with the entity is a fixed value. Otherwise, if the `initialValue` is a script, there are 2 possibilities:

1. If the `evaluateScriptOnlyForInitialization` tag is true, the value of the associated variable is initialized with value set to the script at the beginning of simulation.
2. If the `evaluateScriptOnlyForInitialization` tag is false, the value of the associated variable is set by evaluating the script at each simulation step.

If this tag is not present in a given context, its value is assumed to be true.

3.24.1 Attributes

The `evaluateScriptOnlyForInitialization` tag has the following attributes:

Name	Details
`value`	A Boolean value.

3.24.2 Structure

The `evaluateScriptOnlyForInitialization` tag has no inner tags.

3.24.3 Example

```
<csml:entity label="e2" name="e2" type="continuous">
   <csml:parameter label="m2" type="Double" minimumValue="0"
     maximumValue="infinite">
    <csml:script><![CDATA[m1+m2]]>
    </csml:script>
   </csml:parameter>
   <csml:evaluteScriptOnlyForInitialization value="false"/>
   ...
  </csml:entity>
```

3.25 figure

Parent tag: graphics, net

The `figure` tag specifies the properties of a graphical element in the model. That graphical element may be a visual representation for a model object, such as an entity, a process, or a connector; or it can be a standalone, illustrative element, such as a text frame with caption for some part of the model.
The exact type of graphical element represented by a given `figure` tag is specified by the contained inner tag.

3.25.1 Attributes

The `figure` tag has no attributes.

3.25.2 Structure

The `figure` tag contains the following inner tags:

Name	Occurrence	Details

associationConnector	once exclusively	See 3.76
continuousEntity	once exclusively	See 3.76
continuousProcess	once exclusively	See 3.76
cross	once exclusively	See 3.76
discreteEntity	once exclusively	See 3.76
discreteProcess	once exclusively	See 3.76
ellipse	once exclusively	See 3.76
genericEntity	once exclusively	See 3.76
genericProcess	once exclusively	See 3.76
image	once exclusively	See 3.76
inhibitoryConnector	once exclusively	See 3.76
objectFigure	once exclusively	See 3.76
processConnector	once exclusively	See 3.76
rectangle	once exclusively	See 3.76
roundBounds	once exclusively	See 3.76
roundRectangle	once exclusively	See 3.76
textArea	once exclusively	See 3.76

3.25.3Example

```
<csml:figure>
 <csml:continuousEntity location="392.0 78.0"/>
</csml:figure>
```

3.26 firing

Parent tag: process, connector

The `firing` specifies a Boolean value that defines process or connector activity, that is, whether the process or connector can fire or not.

3.26.1Attributes

The `firing` tag has the following attributes:

Name	Details
type	the type of firing value, always is "Boolean"
value	the value of firing, possible values: "true" or "false", if the tag has inner script there is no value attribute
firingStyle	the style of firing for process, possible values: "and", "or"
connectorFiringStyle	the style of firing for connector, possible values: "thresh-old", "rule", "nocheck"

3.26.2Structure

The tag has no inner tags.

3.26.3Example

```
<csml:simulationCondition>
 ...
 <csml:firing type="Boolean" value="true" firingStyle="and"/>
 ...
</csml:simulationCondition>
```

3.27 function

Parent tag: process

The `function` tag specifies the connections between the process defined by its parent tag and entities in the model.

3.27.1Attributes

The `function` tag has no attributes.

3.27.2Structure

The `function` tag contains the following inner tags:

Name	Occurrence	Details
`connector`	any number	Connectors between a process and entities.

3.27.3Example

```
<csml:function>
 <csml:connector label="c1" name="c1" type="process" from="e1"
  to="p1">
  ...
 </csml:connector>
 <csml:connector label="c2" name="c2" type="process" from="p1"
  to="e2">
  ...
 </csml:connector>
</csml:function>
```

3.28 genericEntity

Parent tag: figure

The `genericEntity` tag defines the graphical representation of a generic entity.

3.28.1Attributes

See 3.75.1

3.28.2Structure

See 3.75.2

3.28.3Example

```
<csml:figure>
 <csml:genericEntity location="312.0 73.0"/>
</csml:figure>
```

3.29 genericProcess

Parent tag: figure

The `genericProcess` tag defines the graphical representation of a generic process.

3.29.1Attributes

See 3.75.1

3.29.2Structure

See 3.75.2

3.29.3Example

```
    <csml:figure>
     <csml:genericProcess location="82.0 78.0"/>
    </csml:figure>
```

3.30 graphics

Parent tag: connector, entity, entityProxy, object, process

The `graphics` tag, along with the set of its inner tags, describes the graphical representation of an entity, process, connector, or imported model described by its parent tag.

3.30.1Attributes

The `graphics` tag has no attributes.

3.30.2Structure

The `graphics` tag contains the following inner tags:

Name	Occurrence	Details
`figure`	optional	Contains the tag that defines the graphical representation of the entity, process, or connector.
`sourceArrow`	optional	For a connector, contains the tag that defines the graphical representation of the start arrow of the connector.
`targetArrow`	optional	For a connector, contains the tag that defines the graphical representation of the end arrow of the connector.
`text`	any number	Defines how a given parameter (speed, variable name, variable value, etc.) of an entity, process, or connector should be displayed.

3.30.3Example

```
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="125.0 128.0 217.0 128.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
pointsInAbsoluteCoordinate="217.0 125.0 217.0 131.0 225.0 128.0"/>
      </csml:targetArrow>
     </csml:graphics>
```

3.31 group

Parent tag: net

The `group` tag is used for grouping model and graphical elements. Grouped elements are contained *inside* the respective group tag. This tag only provides the support for visual editing of the model, such as binding together of several elements whose prop-

erties can be modified at once with a visual CSML editor. It doesn't influence the model from the functional point of view.

3.31.1Attributes

The `group` tag has the following attributes:

Name	Details
`label`	Unique identifier by which a given group tag is referenced in other tags.
`name`	Text that may be used for labeling the group in model visualization.

3.31.2Structure

The `group` tag contains the same inner tags as a net tag. Other `group` tags are allowed inside a `group` tag.

3.31.3Example

```
<csml:group label="g1" name="g1">
 <csml:process label="p1" name="p1" type="continuous">
  <csml:simulationCondition>
   <csml:priority value="0"/>
   <csml:firing type="Boolean" value="true" firingStyle="and"/>
   <csml:delay type="Long" value="0.0" delayStyle="nodelay"/>
   <csml:calc calcStyle="speed"/>
   <csml:kinetic type="Double" value="1" kineticStyle="custom">
    <csml:parameter name="custom" value="1"/>
   </csml:kinetic>
  </csml:simulationCondition>
  <csml:function>
   <csml:connector label="c1" name="c1" type="process" from="e1"
         to="p1" linestyle="straight" supportpath="true">
    <csml:firing type="Double" value="0"
         connectorFiringStyle="threshold"/>
    <csml:kinetic/>
    <csml:graphics>
     <csml:figure>
      <csml:processConnector points="183.0 70.0 271.0 70.0"/>
     </csml:figure>
     <csml:targetArrow>
      <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0
         3.0" pointsInAbsoluteCoordinate="271.0 67.0 271.0 73.0
         279.0 70.0"/>
     </csml:targetArrow>
    </csml:graphics>
   </csml:connector>
   <csml:connector label="c4" name="c2" type="process" from="e1"
         to="p1" linestyle="straight" supportpath="false">
    <csml:firing type="Double" value="0"
         connectorFiringStyle="threshold"/>
    <csml:kinetic/>
    <csml:graphics>
     <csml:figure>
      <csml:processConnector points="183.0 70.0 271.0 70.0"/>
     </csml:figure>
     <csml:targetArrow>
      <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0
         3.0" pointsInAbsoluteCoordinate="271.0 67.0 271.0 73.0
         279.0 70.0"/>
     </csml:targetArrow>
```

```
      </csml:graphics>
     </csml:connector>
     <csml:connector label="c2" name="c2" type="process" from="p1"
           to="e2" linestyle="straight" supportpath="true">
      <csml:firing type="Double" value="0"
                 connectorFiringStyle="threshold"/>
      <csml:kinetic/>
      <csml:graphics>
       <csml:figure>
        <csml:processConnector points="301.0 70.0 397.0 70.0"/>
       </csml:figure>
       <csml:targetArrow>
        <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0
           3.0" pointsInAbsoluteCoordinate="397.0 67.0 397.0 73.0
           405.0 70.0"/>
       </csml:targetArrow>
      </csml:graphics>
     </csml:connector>
     <csml:connector label="c3" name="c1" type="process" from="p1"
           to="e2" linestyle="straight" supportpath="false">
      <csml:firing type="Double" value="0"
           connectorFiringStyle="threshold"/>
      <csml:kinetic/>
      <csml:graphics>
       <csml:figure>
        <csml:processConnector points="301.0 70.0 397.0 70.0"/>
       </csml:figure>
       <csml:targetArrow>
        <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0
           3.0" pointsInAbsoluteCoordinate="397.0 67.0 397.0 73.0
           405.0 70.0"/>
       </csml:targetArrow>
      </csml:graphics>
     </csml:connector>
    </csml:function>
    <csml:biological>
     <csml:effectList/>
    </csml:biological>
    <csml:graphics>
     <csml:figure>
      <csml:continuousProcess location="279.0 63.5"/>
     </csml:figure>
    </csml:graphics>
   </csml:process>
   <csml:entity label="e2" name="e2" type="continuous">
    <csml:parameter label="m2" type="Double" initialValue="0"
           minimumValue="0" maximumValue="infinite"/>
    <csml:public value="true"/>
    <csml:graphics>
     <csml:figure>
      <csml:continuousEntity location="405.0 58.5"/>
     </csml:figure>
    </csml:graphics>
   </csml:entity>
   <csml:entity label="e1" name="e1" type="continuous">
    <csml:parameter label="m1" type="Double" initialValue="10"
           minimumValue="0" maximumValue="infinite"/>
    <csml:graphics>
     <csml:figure>
      <csml:continuousEntity location="161.0 58.5"/>
```

```
    </csml:figure>
   </csml:graphics>
  </csml:entity>
 </csml:group>
```

3.32 image

Parent tag: figure

The `image` tag represents a graphical image used in the graphical representation of a model. Images can be used either for purely illustrational purposes or instead of standard graphical elements representing model elements.

3.32.1 Attributes

See 3.75.1 and 3.76.1

3.32.2 Structure

See 3.75.2 and 3.76.2

3.32.3 Example

An image used for illustration:

```
<csml:image location="264.5 11.5" size="86.0 172.0"
  filePath="centrosome-1.png" fileType="localFile"/>
```

An entity with an image instead of a `continuousEntity` tag:

```
<csml:entity label="n0" name="Fas ligand trimer, Fas receptor
   complex" type="continuous">
 <csml:parameter label="m3" type="Double" initialValue="0"
   minimumValue="0" maximumValue="infinite"/>
 <csml:graphics>
  <csml:figure>
   <csml:image location="96.5 229.5" size="30.0 72.0"
   filePath="images/FasRecep.png" fileType="localFile"
   fileFormat="img"/>
  </csml:figure>
  <csml:text name="value" location="89.5 299.5"
   fontName="sansserif.plain" fontSize="12"/>
  <csml:text name="variable" location="124.5 213.5"
   fontName="sansserif.plain" fontSize="12"/>
  <csml:text name="name" location="16.0 201.0"
   fontName="sansserif.plain" fontSize="12" arrangement="None"
   offsets="14.0 -55.5"/>
 </csml:graphics>
</csml:entity>
```

3.33 imageArrow

Parent tag: sourceArrow, targetArrow

The `imageArrow` tag represents a graphical image used for the start or end arrow in the graphical representation of a process connector.

3.33.1 Attributes

See 3.75.1 and 3.76.1

3.33.2Structure

See 3.75.2 and 3.76.2

3.33.3Example

```
<csml:imageArrow location="0.0 0.0" size="32.0 32.0"
     pointsInAbsoluteCoordinate="545.0 278.0 545.0 310.0 577.0
     310.0 577.0 278.0" filePath="arrows0g.gif"
     fileType="localFile"/>
```

3.34 inhibitoryConnector

Parent tag: figure

The `inhibitoryConnector` tag defines the graphical representation of an inhibitory connector.

3.34.1Attributes

See 3.75.1

3.34.2Structure

See 3.75.2

3.34.3Example

```
      <csml:figure>
       <csml:inhibitoryConnector points="191.0 252.0 269.0 252.0"/>
      </csml:figure>
```

3.35 inhibitoryConnectorArrow

Parent tag: sourceArrow, targetArrow

The `inhibitoryConnectorArrow` tag defines the end arrow of an inhibitory connector's graphical representation. This tag corresponds to the traditional "Petri net community" style of the arrow; see also 3.4.

3.35.1Attributes

See 3.75.1

3.35.2Structure

See 3.75.2

3.35.3Example

```
      <csml:targetArrow>
       <csml:inhibitoryConnectorArrow location="0.0 0.0"
           pointsInAbsoluteCoordinate="157.0 246.0 151.0 246.0
           151.0 260.0 157.0 260.0"/>
      </csml:targetArrow>
```

3.36 item

Parent tag: references

The item tag specifies the URL of a website related to a model element.

3.36.1Attributes

The item tag has the following attributes:

Name	Details
url	URL of a site related to the model element.

3.36.2Structure

The item tag has no inner tags. The site reference comment text is contained as text or in a CDATA section.

3.36.3Example

```
<csml:references>
 <csml:item
  url="http://www.ncbi.nlm.nih.gov/PubMed/"><![CDATA[PubMed]]>
 </csml:item>
</csml:references>
```

3.37 kinetic

Parent tag: process, connector

The kinetic tag defines a style of kinetic of process or connector.

3.37.1Attributes

The kinetic tag has the following attributes:

Name	Details
kineticStyle	Style of kinetic in process, possible values: "mass", "stochastic-mass", "connectorrate", "connectorcustom" or "custom".

3.37.2Structure

The kinetic tag contains the following inner tag:

Name	Occurrence	Details
*/parameter	optional	Parameters related to the style of kinetic, there are five possibilities: 1) If kineticStyle="mass" the kinetic tag contains variable number of parameters inside (coefficient1, coefficient2 and stoichiometry for each connector connected to process). 2) If kineticStyle="stochasticmass" the kinetic tag contains three parameters inside (coefficient1, coefficient2, standard deviation and stoichiometry for each connector connected to process). 3) If kineticStyle="connectorrate" the kinetic tag contains variable number of parameters inside

		(`rate` and `stoichiometry` for each connector connected to process). 4) If `kineticStyle="connectorcustom"` the `kinetic` tag contains no parameter inside. The kinetic of the process is defined on the each connector. 5) If `kineticStyle="custom"` the `kinetic` tag contains one parameters inside (`custom`).

3.37.3Example

```
    <csml:kinetic kineticStyle="custom">
     <csml:parameter name="custom" value="1"/>
    </csml:kinetic>

    <csml:kinetic kineticStyle="custom">

    <csml:kinetic kineticStyle="stochasticmass">
     <csml:parameter name="coefficient2" value="0.1"/>
     <csml:parameter name="coefficient1" value="0.01"/>
     <csml:parameter name="variance" value="1.0"/>
    </csml:kinetic>
```

3.38 log

Parent tag: model

The `log` tag contains a simulation log.

3.38.1Attributes

The `log` tag has no attributes.

3.38.2Structure

The `log` tag contains the following inner tags:

Name	Occurrence	Details
`logItem`	once	Basic information about the simulation.
`logVariableList`	once	List of variables that are logged in the simulation.
`logDataList`	once	List of time with the logged property values.

3.38.3Example

```
 <csml:log>
  <csml:logItem filename="sim12.cil" status="finished"
     timeStamp="Dec    2, 2004 2:06:39 PM" server="localhost">
   <csml:comment><![CDATA[test]]>
   </csml:comment>
  </csml:logItem>
  <csml:logVariableList>
   <csml:variable labelRef="e2" logIndex="0" elementType="entity"
     propertyType="value"/>
   <csml:variable labelRef="e1" logIndex="1" elementType="entity"
     propertyType="value"/>
  </csml:logVariableList>
  <csml:logDataList>
   <csml:logData timePoint="0">
    <csml:logValue id="0" value="0.0"/>
```

```
    <csml:logValue id="1" value="10.0"/>
   </csml:logData>
   ...
   <csml:logData timePoint="1999000000"/>
   <csml:logData timePoint="2000000000"/>
  </csml:logDataList>
 </csml:log>
```

3.39 logData

Parent tag: logDataList

The `logData` tag contains the logged values for one given moment in time. The logged values are saved into a sequence of `logValue` tags. Logged values that didn't change from the previous logged time point may be omitted.

3.39.1Attributes

The `logData` tag has the following attributes:

Name	Details
`timePoint`	Time for which the values have been logged.

3.39.2Structure

The `logData` tag contains the following inner tags:

Name	Occurrence	Details
`logValue`	any number	A single logged value for a given time point.

3.39.3Example

```
  <csml:logDataList>
   <csml:logData timePoint="0">
    <csml:logValue id="0" value="0.0"/>
    <csml:logValue id="1" value="10.0"/>
   </csml:logData>
   ...
   <csml:logData timePoint="1999000000"/>
   <csml:logData timePoint="2000000000"/>
  </csml:logDataList>
```

3.40 logDataList

Parent tag: log

The `logDataList` tag contains a set of time points with the logged property values.

3.40.1Attributes

The `logDataList` tag has the following attributes:

Name	Details
`logSteps`	Number of time points that have been logged.

3.40.2Structure

The `logDataList` tag contains the following inner tags:

Name	Occurrence	Details
`logData`	any number	A single time point with logged values.

3.40.3Example

```
<csml:logDataList>
 <csml:logData timePoint="0">
  <csml:logValue id="0" value="0.0"/>
  <csml:logValue id="1" value="10.0"/>
 </csml:logData>
 ...
 <csml:logData timePoint="1999000000"/>
 <csml:logData timePoint="2000000000"/>
</csml:logDataList>
```

3.41 logging

Parent tag: entity, entityProxy, process

The `logging` tag defines whether an entity's associated variable values or process' state should be logged.

3.41.1Attributes

The `logging` tag has the following attributes:

Name	Details
`value`	A Boolean value.

3.41.2Structure

The `logging` tag has no inner tags.

3.41.3Example

```
<csml:logging value="true"/>
```

3.42 logItem

Parent tag: log

The `logItem` tag contains "basic report" information about a simulation that has been conducted for a given model.

3.42.1Attributes

The `logItem` tag has the following attributes:

Name	Details
`author`	Name of the user that ran the simulation.
`filename`	Name of the file to which the simulation log was written.
`server`	Reserved for future use. The only possible value is "`localhost`".
`status`	Current status of the simulation. Possible values are: "`unknown`", "`running`", "`finished`", and "`interrupted`".
`timeStamp`	Date and time of the simulation. Platform-dependent format is used.

3.42.2Structure

The `logItem` tag contains the following inner tags:

Name	Occurrence	Details
comment	optional	User comment for the simulation.

3.42.3Example

```
<csml:logItem filename="sim12.cil" status="finished"
   timeStamp="Dec    2, 2004 2:06:39 PM" server="localhost">
 <csml:comment><![CDATA[test]]>
 </csml:comment>
</csml:logItem>
```

3.43 logValue

Parent tag: logData

The `logValue` tag contains one logged value for one given element.

3.43.1Attributes

The `logValue` tag has the following attributes:

Name	Details
`id`	Logged value index, as specified by the `logIndex` attribute of the `variable` tag.
`value`	Logged value.

3.43.2Structure

The `logValue` tag has no inner tags.

3.43.3Example

```
<csml:logData timePoint="0">
 <csml:logValue id="0" value="0.0"/>
 <csml:logValue id="1" value="10.0"/>
</csml:logData>
```

3.44 logVariableList

Parent tag: log

The `logVariableList` tag specifies a list of variables that should be logged for a given simulation.

3.44.1Attributes

The `logVariableList` tag has the following attributes:

Name	Details
`entityCount`	Number of entities that should be logged. (Logged processes are not counted.

3.44.2Structure

The `logVariableList` tag contains the following inner tags:

Name	Occurrence	Details
`variable`	any number	Specifies a single variable that should be logged.

3.44.3Example

```
<csml:logVariableList>
 <csml:variable labelRef="e2" logIndex="0" elementType="entity"
   propertyType="value"/>
```

```
   <csml:variable labelRef="e1" logIndex="1" elementType="entity"
     propertyType="value"/>
  </csml:logVariableList>
```

3.45 model

Root tag

The `model` tag is the top-level tag of the CSML data structure.

3.45.1 Attributes

The `model` tag has the following attributes:

Name	Details
`version`	Major version of the CSML standard used in the document. It has the value of "1" in CSML 1.0 and in CSML 1.9.
`minorVersion`	Minor version of the CSML standard used in the document. It has the value of "0" in CSML 1.0 and "9" value in CSML 1.9.

3.45.2 Structure

The `model` tag contains the following inner tags:

Name	Occurrence	Details
`net`	optional	Description of the Petri net of the model
`class`	any number	Imported, internalized model
`unitdefs`	optional	Provide support for physical units
`simulation`	optional	Simulation settings
`charts`	optional	Simulation chart settings
`log`	optional	Simulation log

3.45.3 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
 xmlns:csml="http://www.csml.org/csml/version1">
 <csml:unitdefs>
  ...
 </csml:unitdefs>
 <csml:net>
  ...
 </csml:net>
 <csml:simulation ... />
 <csml:charts>
  ...
 </csml:charts>
</csml:model>
```

3.46 net

Parent tag: model

The `net` tag defines the Petri net of a model and optionally its graphical representation and illustration. It encloses a list of model objects: entities, processes, figures, references to imported models, and information on element grouping.

See also: 3.9

3.46.1Attributes

The `net` tag has the following attribute:

Name	Details
`label`	Name of the net.

3.46.2Structure

The `net` tag contains the following inner tags:

Name	Occurrence	Details
`entity`	any number	Entities in the model.
`process`	any number	Processes in the model.
`figure`	any number	Graphical elements used for illustration, with no relation to model elements.
`object`	any number	References to `net` tags in other CSML documents or internally defined classes.
`group`	any number	Groups of model and/or graphical illustration elements. This provides support for visual editing of the model. Elements grouped by a `group` tag are contained *inside* it and not at the top level of the `net` tag.

3.46.3Example

```
<csml:net label=”root”>
 <csml:entity label="e1" name="e1" type="continuous">
  ...
 </csml:entity>
 <csml:entity label="e2" name="e2" type="continuous">
  ...
 </csml:entity>
 <csml:process label="p1" name="p1" type="continuous">
  ...
 </csml:process>
</csml:net>
```

3.47 `object`

Parent tag: net

The `object` tag defines a model imported into another model. Elements from such an imported model are referenced in other parts of the higher-order model by “model-label element-label” identifiers (with a space between the two labels); for example, “`o1 e3`” may mean entity with label “`e3`” from model with label “`o1`”.

3.47.1Attributes

The `object` tag has the following attributes:

Name	Details
`label`	Unique identifier by which the model defined by this tag is referenced elsewhere in the model (that is, in the “`from`” and “`to`” attributes of the connectors and in scripts).

name	Text label that may be used for labeling the imported model in model visualization.
referenceType	The type of the reference; one of fileType or internalType.
reference	For file-type references, this attribute is the path to an external file with the referenced model definition. For internalized model references, this is the label attribute of the respective class tag.

3.47.2Structure

The object tag contains the following inner tags:

Name	Occurrence	Details
comment	optional	Comment for the imported model.
references	optional	List of URL sites related to the object.
entityProxy	any number	Details of a referenced entity in the imported model. The entity must be made public in the imported model.
graphics	optional	Graphical representation of the imported model.

3.47.3Example

Model from another file:

```
<csml:object label="o1" name="h1" referenceType="fileType"
   reference="simple_grouped_export.gon">
 <csml:entityProxy type="continuous" reference="e2">
  <csml:parameter label="m2" initialValue="0.0"
   minimumValue="0.0"      maximumValue="infinite"/>
  <csml:public value="true"/>
  <csml:graphics>
   <csml:figure>
    <csml:continuousEntity location="78.0 73.0"/>
   </csml:figure>
  </csml:graphics>
 </csml:entityProxy>
 <csml:graphics>
  <csml:figure>
   <csml:objectFigure location="100.0 60.0"/>
  </csml:figure>
 </csml:graphics>
</csml:object>
```

Internalized model:

```
<csml:object label="o1" name="h1" referenceType="internalType"
   reference="internalized1">
 <csml:graphics>
  <csml:figure>
   <csml:objectFigure location="474.0 307.0"/>
  </csml:figure>
 </csml:graphics>
</csml:object>
```

3.48 objectFigure

Parent tag: figure

The objectFigure tag defines the graphical representation of an imported model, seen as a single “block”.

3.48.1Attributes

See 3.75.1

3.48.2Structure

See 3.75.2

3.48.3Example

```
<csml:figure>
 <csml:objectFigure location="100.0 60.0"/>
</csml:figure>
```

3.49 parameter

Parent tag: entity, entityProxy

The `parameter` tag specifies the initial value, range, and type for the variable associated with the parent `entity`.
This tag is a scriptable tag; see 3.74

> Note: The inner `parameter` tag occurs in the kinetic tag. The meanings of both tags are different.

3.49.1Attributes

The `parameter` tag has the following attributes:

Name	Details
`initialValue`	Initial value (Double for a continuous entity, Integer or Long for a discrete entity, Boolean or String for generic entity). This attribute is illegal if there is an inner `script` tag.
`label`	Name of the variable associated with the parent `entity` tag.
`maximumValue`	The maximum value that can be assigned to the variable associated with the entity. It can be 1 or "`infinite`" (limited by the type of associated variable only).
`minimumValue`	The minimum value that can be assigned to the variable associated with the entity. It can be 0 or "`-infinite`" (limited by the type of associated variable only).
`type`	The type of the variable associated with the parent `entity`. It can be: "`Boolean`", "`Double`", "`Integer`", "`Long`" or "`String`".
`unit`	Name of the unit (possible with prefix) being referred to. It can be: "`gram`", "`kilogram`", "`lumen`", etc.

3.49.2Structure

The `parameter` tag contains the following inner tags:

Name	Occurrence	Details
`script`	optional	If the `initialValue` attribute is a script, there are 2 possibilities: 1) If the `evaluateScriptOnlyForInitialization` attribute of `parameter`'s parent `entity` tag is true, the associated variable is initialized with value set to the script at the

		beginning of simulation. 2) If the `evaluateScriptOnlyForInitialization` attribute of `parameter`'s parent `entity` tag is false, the associated variable is set by evaluating the script at each simulation step. This tag is illegal if the `initialValue` attribute is present.

3.49.3Example

With `initialValue` set directly:

```
   <csml:parameter label="m1" initialValue="10.0"
minimumValue="0.0"     maximumValue="infinite" type=”Double”
unit=”candela” />
```

With `initialValue` set as a script:

```
   <csml:parameter label="m1" minimumValue="0.0"
     maximumValue="infinite" type=”Double” unit=”hertz”>
    <csml:script><![CDATA[PI()*2.4]]>
    </csml:script>
   </csml:parameter>
```

*3.50 */parameter*

Parent tag: kinetic

The `parameter` tag specifies the name and value of parameters related to kinetic style of parent tag `kinetic`.

Note: The other `parameter` tag occurs in an `entity` and `entityProxy` tag. The meanings of both tags are different.

3.50.1Attributes

The `parameter` tag has the following attributes:

Name	Details
`name`	Name of the parameter, possible values: “`coefficient1`”, “`coefficient2`”, “`stoichiometry`”, “`standard deviation`”, “`rate`” and “`custom`”.
`value`	Value of the parameter – Double, Integer, Long value or a script.

3.50.2Structure

The `parameter` tag has no inner tags.

3.50.3Example

```
   <csml:parameter name="coefficient2" value="0.1"/>
   <csml:parameter name="coefficient1" value="0.01"/>
   <csml:parameter name="variance" value="1.0"/>
   <csml:parameter name="rate" value="m1*0.77"/>
   <csml:parameter name="standard deviation" value="1.0"/>
```

3.51 priority

Parent tag: process

The priority tag defines the priority of the process during simulation. If two processes P1 and P2 with equal priority fire at the same time, the order of firing P1 and P2 is random. But if the priority of P1 is larger than P2, then P1 always fires before P2.

3.51.1 Attributes

The priority tag has the following attributes:

Name	Details
value	Priority of a process during simulation; a positive integer from "0" to "20" value.

3.51.2 Structure

The priority tag has no inner tags.

3.51.3 Example

```
    <csml:priority value="0"/>
```

3.52 process

Parent tag: model

The process tag describes a process and connectors associated with it. The properties defined by this tag's attributes and inner tags are the process type (continuous, discrete, or generic), its activity (whether it can fire), its speed, its connections with entities, its graphical representation, etc.

3.52.1 Attributes

The process tag has the following attributes:

Name	Details
label	Unique identifier by which a given process tag is referenced in other tags, such as connector.
name	Text that may be used for labeling the process in model visualization.
type	Type of process – continuous, discrete, or generic. Possible values: continuous, discrete, generic.

3.52.2 Structure

The process tag contains the following inner tags:

Name	Occurrence	Details
comment	optional	Comment for the process.
references	optional	List of URL sites related to the process.
firing	optional	Activity and speed of the process.
function	optional	Connections between the process and the entities in

		the model.
biological	optional	Reserved for future CSML release.
logging	optional	If true, the state of the process will be logged during simulation.
graphics	optional	Graphical representation of the process.

3.52.3Example

```
<csml:process label="p1" name="p1" type="continuous">
 <csml:simulationCondition>
  ...
 </csml:simulationCondition>
 <csml:function>
  <csml:connector ...>
   ...
  </csml:connector>
  ...
 </csml:function>
 <csml:biological>
   <csml:effectList/>
 </csml:biological>
 <csml:graphics>
  ...
 </csml:graphics>
</csml:process>
```

3.53processConnector

Parent tag: figure

The processConnector tag defines the graphical representation of a (discrete or continuous) process connector.

3.53.1Attributes

See 3.75.1

3.53.2Structure

See 3.75.2

3.53.3Example

```
    <csml:figure>
     <csml:processConnector points="125.0 128.0 217.0 128.0"/>
    </csml:figure>
```

3.54processConnectorArrow

Parent tag: sourceArrow, targetArrow

The processConnectorArrow tag defines the graphical representation of an end arrow of a normal process connector.

3.54.1Attributes

See 3.75.1

3.54.2Structure

See 3.75.2

3.54.3Example

```
<csml:targetArrow>
 <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
     pointsInAbsoluteCoordinate="197.0 105.0 197.0 111.0
     205.0 108.0"/>
</csml:targetArrow>
```

3.55 property

Parent tag: chart

The `property` tag defines the geometrical location of a chart specified by its parent tag. The location is defined by left-upper corner position of the chart and its size (width and height). There must be exactly four `property` tags inside a `chart` tag, each for one location parameter.

3.55.1Attributes

The `property` tag has the following attributes:

Name	Details
`key`	One of `x`, `y`, `width`, or `height`. Defines which of the location parameters is defined by this tag.
`value`	Position or length, in pixels.

3.55.2Structure

The `property` tag has no inner tags.

3.55.3Example

```
<csml:property key="width" value="388"/>
<csml:property key="height" value="288"/>
<csml:property key="y" value="126"/>
<csml:property key="x" value="567"/>
```

3.56 public

Parent tag: entity, entityProxy

The `public` tag defines whether the entity represented by its parent tag is visible in higher-level models. The true value denotes that the associated variable can be referenced in models in which the entity is encapsulated in a hierarchical object element. If this tag is not present in a given context, its value is assumed to be false.

3.56.1Attributes

The `public` tag has the following attributes:

Name	Details
`value`	Boolean value.

3.56.2Structure

The `public` tag has no inner tags.

3.56.3Example

```
    <csml:public value="true"/>
```

3.57 rectangle

Parent tag: figure

The `rectangle` tag defines a rectangle in graphical representation of the model.

3.57.1Attributes

See 3.75.1

3.57.2Structure

See 3.75.2

3.57.3Example

```
    <csml:figure>
     <csml:rectangle location="590.0 158.0" size="24.0 48.0"/>
    </csml:figure>
```

3.58 rectangleArrow

Parent tag: sourceArrow, targetArrow

The `rectangleArrow` tag represents a rectangle used for the start or end arrow in the graphical representation of a process connector.

3.58.1Attributes

See 3.75.1

3.58.2Structure

See 3.75.2

3.58.3Example

```
<csml:rectangleArrow location="0.0 0.0" size="36.0 36.0"
     pointsInAbsoluteCoordinate="275.0 276.0 275.0 312.0 311.0
312.0      311.0 276.0"/>
```

3.59 references

Parent tag: connector, entity, entityProxy, object, process

The `references` tag contains a list of URLs of websites related to its parent tag's associated model element. The website references are used for purely informational purposes (for example, as information on where to find publications concerning a given part of the model).

3.59.1Attributes

The `references` tag has no attributes.

3.59.2Structure

The `references` tag has the following inner tags:

Name	Occurrence	Details
`item`	any number	A single website URL (with comments or site name as text or in a CDATA section).

3.59.3Example

```
<csml:references>
 <csml:item
  url="http://www.ncbi.nlm.nih.gov/PubMed/"><![CDATA[PubMed]]>
 </csml:item>
</csml:references>
```

3.60 roundBounds

Parent tag: figure

The `roundBounds` tag defines a rectangle with rounded corners in the graphical representation of the model.

3.60.1Attributes

See 3.75.1

3.60.2Structure

See 3.75.2

3.60.3Example

```
<csml:figure>
 <csml:roundBounds depth="1" location="213.0 360.0" size="215.0
   119.0" fillColor="255 255 0 255"/>
</csml:figure>
```

3.61 roundRectangle

Parent tag: figure

The `roundRectangle` tag defines a rectangle with rounded corners in the graphic representation of the model.

3.61.1Attributes

See 3.75.1

3.61.2Structure

See 3.75.2

3.61.3Example

```
<csml:figure>
 <csml:roundRectangle location="590.0 158.0" size="24.0 48.0"/>
</csml:figure>
```

3.62 script

Parent tag: delay, parameter, firing

The `script` tag contains the function definition for its parent tag. For tags that may have a `value` attribute, such as `delay`, it is used instead of that attribute. It is illegal for those tags to have both a `value` attribute and an inner `script` tag. The return type of the function should be consistent with the type of the parent tag value.

3.62.1 Attributes

The inner `script` tag has the following attributes:

Name	Details
`scriptType`	Specifies the type of the interpreter. Currently, the only supported type is "`gon`"; it is the default value if the attribute is not present. The interpreter for the "`gon`" type is Pnuts with some custom functions.
`name`	Not used – backwards compatibility only.

3.62.2 Structure

The inner `script` tag has no inner tags. The function definition is contained as text or in a CDATA section.

3.62.3 Example

```
   <csml:parameter label="m1" type="Double" minimumValue="0"
maximumValue="infinite">
    <csml:script><![CDATA[return 77;]]>
    </csml:script>
   </csml:parameter>
```

3.63 simulation

Parent tag: model

The `simulation` tag describes the simulation time and update settings.

3.63.1 Attributes

The `simulation` tag has the following attributes:

Name	Details
`enhancedFiring`	Backwards compatibility; always true.
`samplingInterval`	The interval time of the simulation.
`simulationTime`	This attribute tells how long the simulation should last. The ratio `simulationTime`/`samplingInterval` defines the number of steps to be executed in the simulation.
`plotUpdateInterval`	This attribute defines how often the chart windows should be updated. It has no impact on simulation results.
`logUpdateInterval`	This attribute defines how often the internal simulation log should be updated. It has no impact on simulation results. It is ignored if the simulation is played in step mode.

3.63.2Structure

The `simulation` tag has no inner tags.

3.63.3Example

```
<csml:simulation enhancedFiring="true" samplingInterval="0.1"
  simulationTime="200.0" logUpdateInterval="1.0"
  plotUpdateInterval="1.0"/>
```

3.64 sourceArrow

Parent tag: graphics

The `sourceArrow` tag defines a start arrow of the graphical representation of a connector.
See also: 3.78

3.64.1Attributes

See 3.78.1

3.64.2Structure

See 3.78.2

3.64.3Example

```
      <csml:sourceArrow>
       <csml:ellipseArrow location="0.0 0.0"
           pointsInAbsoluteCoordinate="333.0 278.0 333.0 310.0
365.0            310.0 365.0 278.0"/>
      </csml:sourceArrow>
```

3.65 simulationCondition

Parent tag: process

The `simulationCondition` tag describes the activity and calculation style of its parent `process` tag.

3.65.1Attributes

The `simulationCondition` tag has no attributes.

3.65.2Structure

The `simulationCondition` tag contains the following inner tags:

Name	Occurrence	Details
`priority`	optional	A positive integer value. Sets the priority of the process during simulation. If two processes P1 and P2 with equal priority fire at the same time, the order of firing P1 and P2 is random. But if the priority of P1 is larger than P2, then P1 always fires before P2.
`firing`	optional	Boolean function that turns the process on and off, depending on its result and defined firing style. If not present, it is assumed to be always true.
`delay`	optional	A positive double value. Sets the firing delay style for

		process.
calc	optional	Sets the calculation style for process.
kinetic	optional	Sets the kinetic style for process.

3.65.3Example

```
<csml:simulationCondition>
 <csml:priority value="0"/>
 <csml:firing type="Boolean" value="true" firingStyle="or"/>
 <csml:delay type="Long" value="0.0" delayStyle="nodelay"/>
 <csml:calc calcStyle="speed"/>
 <csml:kinetic type="Double" kineticStyle="stochasticmass">
  <csml:parameter name="standard deviation" value="1.0"/>
  <csml:parameter name="coefficient2" value="0.1"/>
  <csml:parameter name="coefficient1" value="0.01"/>
 </csml:kinetic>
</csml:simulationCondition>
```

3.66 targetArrow

Parent tag: graphics

The targetArrow tag defines an end arrow of the graphical representation of a connector.
See also: 3.64, 3.78

3.66.1Attributes

See 3.78.1

3.66.2Structure

See 3.78.2

3.66.3Example

```
<csml:targetArrow>
 <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
     pointsInAbsoluteCoordinate="197.0 105.0 197.0 111.0
     205.0 108.0"/>
</csml:targetArrow>
```

3.67 text

Parent tag: graphics

The text tag defines the graphical representation (as a text label) of a parameter of a model element. The parameter, depending on the element, can be one of: model element name, process or connector speed, connector threshold, variable value, or variable name. This tag may be omitted if default values are used.
See also 3.76

3.67.1Attributes

See 3.75.1 and 3.77.1; additionally, the `text` tag has the following attributes:

Name	Details
`arrangement`	Possible values: "`None`", "`Center`", "`Top`", "`Bottom`", "`Left`", "`Right`", "`Top Left`", "`Top Right`", "`Bottom Left`", "`Bottom Right`"
`name`	The name of the model element's parameter, whose graphical representation is described by this tag. One of `variable`, `value`, `threshold`, `speed`, `name`; see below.
`offsets`	The x- and y-shift relative to the graphical representation of a given element.

The table below shows the possible `name` attribute values and model concepts they refer to:

Value	Refers to
`name`	Entity, process, or connector name.
`speed`	Process or connector speed;
`threshold`	Connector threshold;
`value`	Value of the variable associated to an entity;
`variable`	Name of the variable associated to an entity;

3.67.2Structure

See 3.77.2

3.67.3Example

```
    <csml:text name="speed" location="83.0 258.0"
arrangement="None"     offsets="20.5 15.0"/>
```

3.68 textArea

Parent tag: figure

The `textArea` tag defines a rectangle with text. This graphical element is used for model illustration purposes only.
See also 3.76

3.68.1Attributes

See 3.75.1 and 3.77.1

3.68.2Structure

See 3.75.2; the text is contained as text or in a CDATA section.

3.68.3Example

```
   <csml:textArea location="16.0 200.0" size="388.0 48.0"
fillColor="0 0 0 255" outlineWidth="1.0" textColor="255 255 255
255" fontName="Arial Black" fontStyle=" italic"
fontSize="16"><![CDATA[Text frames can be used to add description
to the graphical representation of a model.]]>
   </csml:textArea>
```

3.69 triangleArrow

Parent tag: sourceArrow, targetArrow

The `triangleArrow` tag represents a triangle used for the start or end arrow in the graphical representation of a process connector.

3.69.1Attributes

See 3.75.1

3.69.2Structure

See 3.75.2

3.69.3Example

```
<csml:triangleArrow points="0.0 0.0 0.0 32.0 32.0 16.0"
    pointsInAbsoluteCoordinate="103.0 278.0 103.0 310.0
    135.0 294.0" outlineWidth="0.0"/>
```

3.70 unit

Parent tag: units

The `unit` tag defines each base element of new physical unit defined in `units` tag.

3.70.1Attributes

The `unit` tag can have the following attributes:

Name	Details
units	Name of the unit being referred to.
prefix	SI decimal scale prefix. If not present, unity scale (1.0) is assumed.
exponent	Unit dimension exponent. If not present, 1.0 is assumed.
multiplier	A multiplier attribute can be used to pre-multiply the quantity to be converted by any real scale factor. For instance, a multiplier of 1.8 is used to define Fahrenheit in terms of Celsius. If not present, 1.0 is assumed.
offset	The offset attribute is used to represent the addition of a constant in the transformation between the current units and the base units. This should only be necessary for the definition of temperature scales. For instance, an offset attribute value of 32.0 is needed to define Fahrenheit in terms of Celsius. If not present, 0.0 is assumed. An offset attribute cannot appear in complex unit definitions (that is, having more than one `unit` tag).

3.70.2Structure

The `unit` tag has no inner tags.

3.70.3Example

```
<csml:units name="inch" symbol="in">
 <csml:unit units="metre" multiplier="0.0254"/>
</csml:units>
```

3.71 unitdefs

Parent tag: model

The `unitdefs` tag describes user-defined physical units.

3.71.1 Attributes

The `unitdefs` tag has no attributes.

3.71.2 Structure

The `unitdefs` tag can contain the following inner tags:

Name	Occurrence	Details
units	any number	Definitions of individual units.

3.71.3 Example

```
<csml:unitdefs>
 <csml:units name="pascal" symbol="Pa">
  <csml:unit units="newton"/>
  <csml:unit units="metre" exponent="-2"/>
 </csml:units>
 <csml:units name="inch" symbol="in">
  <csml:unit units="metre" multiplier="0.0254"/>
 </csml:units>
</csml:unitdefs>
```

3.72 units

Parent tag: unitdefs

3.72.1.1 Attributes

The `units` tag may have the following attributes:

Name	Details
name	The name of the defined unit.
symbol	The symbol of the defined unit.
base_units	If present and "true", this unit is a new base unit, not derived from any other units. If a unit is defined as a new base unit, inner unit tags are disallowed. Assumed false if not present.

3.72.1.2 Structure

The `units` tag can contain the following inner tags:

Name	Occurrence	Details
unit	any number	"Component" units of the defined unit. If a unit is defined as a new base unit, inner unit tags are disallowed.

3.72.2 Example

```
  <csml:units name="pascal" symbol="Pa">
```

```
    <csml:unit units="newton"/>
    <csml:unit units="metre" exponent="-2"/>
  </csml:units>
```

3.73 variable

Parent tag: logVariableList

The `variable` tag specifies a logged variable in the log variable list.

3.73.1 Attributes

The `variable` tag has the following attributes:

Name	Details
`elementType`	Type of model element that should be logged. Currently it can only be "`entity`" or "`process`". Other types, like "`connector`", etc., could be added in the future.
`labelRef`	The value of the `label` attribute of the model object that should be logged.
`logIndex`	The order in which `logValue` tags for a given variable appear inside `logData` tags (zero-based).
`propertyType`	Type of property that should be logged. Currently it can only be "`value`". Other types, like "`speed`", "`delay`", etc., could be added in the future.

3.73.2 Structure

The `variable` tag has no inner tags.

3.73.3 Example

```
    <csml:variable labelRef="e2" logIndex="0" elementType="entity"
      propertyType="value"/>
    <csml:variable labelRef="e1" logIndex="1" elementType="entity"
      propertyType="value"/>
```

3.74 Common properties of scriptable tags

The following tags are scriptable: parameter, delay and firing.

3.74.1 Attributes

Scriptable tags have the following attributes:

Name	Details
`type`	The type of the value returned by this tag. This may be one of: "`Boolean`", "`Double`", "`Integer`", "`Long`", "`String`". Not present in contexts where the value type can be implied, for example in a `parameter` tag for a continuous entity (which always returns a double value).
`value`	Possible values depend on the expected function type; they can be Boolean, integer, or double. This attribute is illegal if there is an inner `script` tag.

3.74.2Structure

Scriptable tags have no inner tags.

3.74.3Example

```
<csml:parameter label="m1" minimumValue="0.0"
  maximumValue="infinite" type=”Double” unit=”hertz”>
 <csml:script><![CDATA[PI()*2.4]]>
 </csml:script>
</csml:parameter>
```

3.75 Common properties of graphical element tags

All tags that define graphical elements have common attributes, described below.
The following tags define graphical elements:

Name	Details
associationConnector	Graphical representation of an association connector.
associationConnectorArrow	End arrow of an association connector.
bioInhibitoryConnectorArrow	End arrow of an inhibitory connector – “biological community” style.
continuousEntity	Graphical representation of a continuous entity.
continuousProcess	Graphical representation of a continuous process.
cross	X-cross. Used for imported model graphical representation.
discreteEntity	Graphical representation of a discrete entity.
discreteProcess	Graphical representation of a discrete process.
ellipse	Ellipse. Used for imported model graphical representation.
ellipseArrow	Start or end arrow of a connector with elliptical shape.
genericEntity	Graphical representation of a generic entity.
genericProcess	Graphical representation of a generic process.
image	External or library image.
imageArrow	Start or end arrow of a connector that uses an image file.
inhibitoryConnector	Graphical representation of an inhibitory connector.
inhibitoryConnectorArrow	End arrow of an inhibitory connector – “Petri net community” style.
objectFigure	Graphical representation of an imported model.
processConnector	Graphical representation of a process connector.
processConnectorArrow	Start or end arrow of a connector – standard arrow.
rectangle	Rectangle. Used for imported model graphical

	representation.
`rectangleArrow`	Start or end arrow of a connector with rectangular shape.
`roundBounds`	Rectangle with rounded corners. Used for illustration only.
`roundRectangle`	Rectangle with rounded corners. Used for imported model graphical representation.
`text`	Text label for a graphical representation of a model element. If not present, default settings should be used.
`textArea`	Text frame. Used for illustration only.
`triangleArrow`	Start or end arrow of a connector with triangular shape.

3.75.1Attributes

Tags representing graphical elements have the following attributes:

Name	Details
`depth`	The Z-order (overlap depth) of the graphical element. It defines the way in which overlapping figures are displayed. A figure with a lower depth value is displayed over a figure with a higher depth value.
`fillColor`	Element surface color given in 8-bit RGBA values; for example "`255 200 0 255`".
`location`	The *x* and *y* coordinates of the element; for example "`405.0 117.0`".
`outlineColor`	Graphical element border line color given in 8-bit RGBA values; for example "`64 64 64 255`".
`outlineDash`	Graphical element stroke (line pattern – dash length and spacing in pixels); for example "`8.0 8.0`".
`outlineWidth`	Graphical element border line width in pixels; for example "`3.0`".
`points`	The *x* and *y* coordinates of the element points in pixels; for example "`125.0 128.0 217.0 128.0`". For tags that have the `pointsInAbsoluteCoordinate` attribute, these are coordinates relative to the first element point (which is then "`0.0 0.0`").
`pointsInAbsoluteCoordinate`	The absolute *x* and *y* coordinates of the element points in pixels; for example "`197.0 105.0 197.0 111.0 205.0 108.0`".
`size`	Width and height of the element in pixels; for example "`164.0 94.0`".
`visible`	If false, a given graphical element should not be displayed. Assumed true if not present.

3.75.2Structure

Graphical element tags have no inner tags.

3.75.3Example

```
<csml:image location="264.5 11.5" size="86.0 172.0"
  filePath="centrosome-1.png" fileType="localFile"/>
```

3.76 Common properties of image tags

The following tags are image tags: image, imageArrow.
Image tags define graphical elements that use image files.

3.76.1Attributes

Image tags have the following attributes:

Name	Details
`filePath`	Relative or absolute path to the image file. The path may refer to a library image, for example in Java class path.
`fileType`	Type of image file. Can be one of "`localFile`", "`resourceFile`" or "`internal`"
`fileFormat`	Format of image file. Can be one of "`img`" (for jpeg, gif, png etc. formats) or "`svg`" (for SVG image file format)

3.76.2Structure

Image tags have no inner tags.

3.76.3Example

```
<csml:image location="264.5 11.5" size="86.0 172.0"
            filePath="centrosome-1.png"
            fileType="localFile"/>

<csml:image location="344.5 233.0" size="428.0 557.0"
            fileType="internal" fileFormat="img">
            <![CDATA[/9j/…xxxx…]]>
</csml:image>
```

3.77 Common properties of text tags

The following tags describe text elements: text, textArea.
Text tags define graphical elements that contain text.

3.77.1Attributes

Text tags have the following attributes:

Name	Details
`fontName`	Font name. Default font face is assumed if this attribute is absent.
`fontSize`	Font size in points. Default font size is assumed if this attribute is absent.
`fontStyle`	Font style – that is, bold and/or italic – as a whitespace-separated list; for example "`bold italic`". Plain font (equivalent to "`plain`" value of the attribute) is assumed if this attribute is absent.
`textColor`	Text color given in 8-bit RGBA values; for example "`255 200 0 255`". Default color is assumed if this attribute is absent.

3.77.2Structure

Text tags have no inner tags.

3.77.3Example

```
<csml:text name="value" location="69.0 208.5" textColor="255 0
       0 255" fontName="Arial Narrow" fontStyle=" italic bold"
       fontSize="24" arrangement="None" offsets="-8.5 45.0"/>
<csml:text name="variable" location="101.0 139.0"
       textColor="255    0 0 255" fontName="Arial Narrow"
       fontStyle=" italic bold"      fontSize="24"/>
<csml:text name="name" location="44.0 138.5" textColor="255 0 0
 255" fontName="Arial Narrow" fontStyle=" italic bold"
 fontSize="24" arrangement="None" offsets="-36.0 -25.0"/>
```

3.78 Common properties of connector arrow tags

The following tags specify connector arrows: sourceArrow, targetArrow.
Connector arrow tags specify start or end arrows of process connector graphical representation.

3.78.1Attributes

Connector arrow tags have no attributes.

3.78.2Structure

Connector arrow tags contain the following inner tags:

Name	Occurrence	Details
associationConnectorArrow	once exclusively	
bioInhibitoryConnectorArrow	once exclusively	
ellipseArrow	once exclusively	
imageArrow	once exclusively	
inhibitoryConnectorArrow	once exclusively	
processConnectorArrow	once exclusively	
rectangleArrow	once exclusively	
triangleArrow	once exclusively	

3.78.3Example

```
<csml:targetArrow>
 <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
     pointsInAbsoluteCoordinate="197.0 105.0 197.0 111.0
     205.0 108.0"/>
</csml:targetArrow>
```

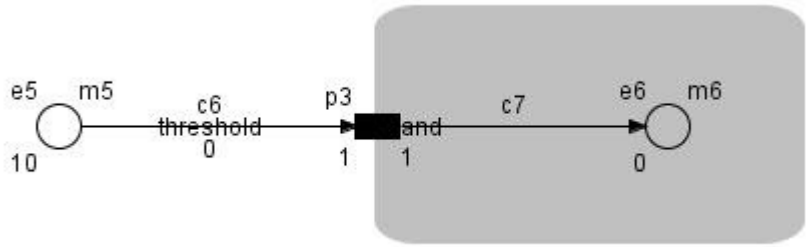
4 Appendix A – References

1. Doi A, Fujita S, Matsuno H, Nagasaki M, Miyano S (2004) Constructing biological pathway models with hybrid functional Petri nets. In Silico Biology, 4(0013), in press.
2. Doi A, Nagasaki M, Fujita S, Matsuno H, Miyano S (2004) Genomic Object Net: II. Modeling biopathways by hybrid functional Petri net with extension. Applied Bioinformatics 2:185–188
3. Matsuno H, Doi A, Hirata Y, Miyano S (2001) XML documentation of biopathways and their simulations in Genomic Object Net. Genome Informatics 12:54–62
4. Matsuno H, Doi A, Nagasaki M, Miyano S (2000) Hybrid Petri net representation of gene regulatory network. Pacific Symposium on Biocomputing 5: 341–352
5. Matsuno H, Fujita S, Doi A, Nagasaki M, Miyano S (2003) Towards biopathway modeling and simulation. In: 24th International Conference on Applications and Theory of Petri Nets (ICATPN 2003). Volume 2679., Lecture Notes in Computer Science, pp 3–22
6. Matsuno H, Murakami R, Yamane R, Yamasaki N, Fujita S, Yoshimori H, Miyano S (2003) Boundary formation by notch signaling in Drosophila multicellular systems: experimental observations and gene network modeling by Genomic Object Net. Pacific Symposium on Biocomputing 8:152–163
7. Matsuno H, Tanaka Y, Aoshima H, Doi A, Matsui M, Miyano S (2003) Biopathways representation and simulation on hybrid functional Petri net. In Silico Biol. 3:389–404, http://www.bioinfo.de/isb/toc_vol_03.html/.
8. Nagasaki M, (2004) A Platform for Biopathway Modeling/Simulation and Recreating Biopathway Databases Towards Simulation, Ph.D Thesis, The University of Tokyo, http://genomicobject.net/pub/nagasaki_phd.pdf.
9. Nagasaki M, Doi A, Matsuno H, Miyano S, (2003) Recreating biopathway databases towards simulation. In: Computational Methods in Systems Biology. Volume 2602 of Lecture Notes in Computer Science., Springer-Verlag, pp 191–192
10. Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) A versatile Petri net based architecture for modeling and simulation of complex biological processes. Genome Informatics 15(1): 180-197.
11. Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) Genomic Object Net:I. A platform for modeling and simulating biopathways. Applied Bioinformatics 2:181–184
12. Nagasaki M, Doi A, Matsuno H, Miyano S, (2004) Integrating biopathway databases for large-scale modeling and simulation. In: The Second Asia-Pacific Bioinformatics Conference. Volume 29 of Conferences in Research and Practice in Information Technology., Australian Computer Society, pp 43–52

5 Appendix B – CSML Examples

5.1 Discrete Model Example

This example shows a simple discrete process, e.g. carrier-mediated transport into a compartment (marked by the yellow box on the drawing) independent on substance gradient.



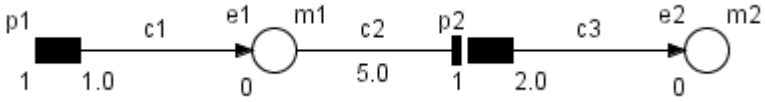
Tokens are transported from place e5 to place e6. The speed of process p3 is governed by its activity parameter and by values of connectors c6 and c7.
The CSML code for this model is presented below:

```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
     xmlns:csml="http://www.csml.org/csml/version1"
     majorVersion="1"  minorVersion="9">
 <csml:unitdefs/>
 <csml:net>
  <csml:entity label="e6" name="e6" type="discrete">
   <csml:parameter label="m6" type="Integer" initialValue="0"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="406.0 157.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:entity label="e5" name="e5" type="discrete">
   <csml:parameter label="m5" type="Integer" initialValue="10"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="102.0 157.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:process label="p3" name="p3" type="discrete">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" value="true" firingStyle="and"/>
    <csml:delay type="Long" value="1.0" delayStyle="delay"/>
    <csml:calc calcStyle="add"/>
    <csml:kinetic type="Double" value="1" kineticStyle="custom">
     <csml:parameter name="custom" value="1"/>
    </csml:kinetic>
   </csml:simulationCondition>
   <csml:function>
```

```
    <csml:connector label="c6" name="c6" type="process" from="e5"
     to="p3" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="124.0 168.0 254.0 168.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="254.0 165.0 254.0 171.0
           262.0 168.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
    <csml:connector label="c7" name="c7" type="process" from="p3"
     to="e6" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="284.0 168.0 398.0 168.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="398.0 165.0 398.0 171.0
           406.0 168.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    <csml:figure>
     <csml:discreteProcess location="261.5 162.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:process>
  <csml:figure>
   <csml:roundBounds depth="1" location="271.0 108.0" size="215.0
     119.0" fillColor="192 192 192 255" arcRatio="20"/>
  </csml:figure>
 </csml:net>
 <csml:simulation enhancedFiring="true" samplingInterval="0.1"
     simulationTime="2000.0" logUpdateInterval="1.0"
     plotUpdateInterval="1.0"/>
</csml:model>
```

5.2 Inhibition Example

This example shows an inhibition process.



Tokens are created in entity e2 by process p2 with speed of 2. At the same time tokens are created in entity e1 by process p1 with speed of 1. The process p2 is disabled when value m1 of entity e1 reaches 5.0.

The inhibition connector is an equivalent of the process p2 activity condition:

```
if (m1 > 5.0) {
    return false;
} else {
    return true;
}
```

The CSML code for this model is presented below:

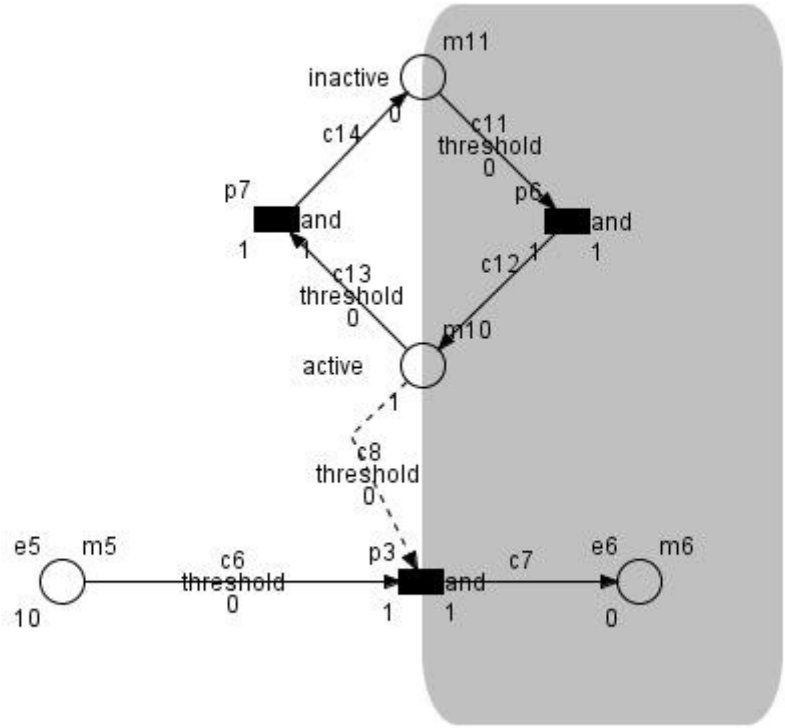
```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
     xmlns:csml="http://www.csml.org/csml/version1"
     majorVersion="1"  minorVersion="9">
 <csml:unitdefs/>
 <csml:net>
  <csml:entity label="e1" name="e1" type="discrete">
   <csml:parameter label="m1" type="Integer" initialValue="0"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="145.0 45.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:entity label="e2" name="e2" type="discrete">
   <csml:parameter label="m2" type="Integer" initialValue="0"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="367.0 59.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:process label="p1" name="p1" type="discrete">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" value="true" firingStyle="and"/>
    <csml:delay type="Long" value="1.0" delayStyle="delay"/>
    <csml:calc calcStyle="add"/>
    <csml:kinetic type="Double" value="1" kineticStyle="custom">
     <csml:parameter name="custom" value="1"/>
    </csml:kinetic>
   </csml:simulationCondition>
   <csml:function>
    <csml:connector label="c1" name="c1" type="process" from="p1"
     to="e1" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
```

```
      <csml:figure>
       <csml:processConnector points="59.0 56.0 137.0 56.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="137.0 53.0 137.0 59.0 145.0
           56.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    <csml:figure>
     <csml:discreteProcess location="37.0 50.0"/>
    </csml:figure>
    <csml:text name="speed" location="64.0 62.0" arrangement="None"
     offsets="20.5 15.0"/>
   </csml:graphics>
  </csml:process>
  <csml:process label="p2" name="p2" type="discrete">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" value="true" firingStyle="and"/>
    <csml:delay type="Long" value="1.0" delayStyle="delay"/>
    <csml:calc calcStyle="add"/>
    <csml:kinetic type="Double" value="2" kineticStyle="custom">
     <csml:parameter name="custom" value="2"/>
    </csml:kinetic>
   </csml:simulationCondition>
   <csml:function>
    <csml:connector label="c2" name="c2" type="inhibitory"
     from="e1"   to="p2" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="5"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:inhibitoryConnector points="167.0 56.0 245.0 56.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:bioInhibitoryConnectorArrow location="0.0 0.0"
           pointsInAbsoluteCoordinate="245.0 49.0 245.0 63.0 253.0
           63.0 253.0 49.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
    <csml:connector label="c3" name="c3" type="process" from="p2"
     to="e2" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="275.0 57.0 359.06719671051025
           67.96528652745785"/>
      </csml:figure>
      <csml:targetArrow>
```

```
        <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
            pointsInAbsoluteCoordinate="359.4552142627135
            64.99048529389918 358.6791791583069 70.94008776101651
            367.0 69.0"/>
       </csml:targetArrow>
      </csml:graphics>
     </csml:connector>
    </csml:function>
    <csml:biological>
     <csml:effectList/>
    </csml:biological>
    <csml:graphics>
     <csml:figure>
      <csml:discreteProcess location="253.0 50.0"/>
     </csml:figure>
     <csml:text name="speed" location="280.0 62.0"
      arrangement="None"      offsets="20.5 15.0"/>
    </csml:graphics>
   </csml:process>
  </csml:net>
  <csml:simulation enhancedFiring="true" samplingInterval="0.1"
      simulationTime="10.0" logUpdateInterval="1.0"
      plotUpdateInterval="1.0"/>
</csml:model>
```

5.3 Catalysis Example

This section demonstrates an example of use of an association connector. The model shown may be an example of carrier-mediated transport where carrier may have active or inactive conformation.



Tokens are transported from place e5 to e6. Process p3 is enabled when entity “active” value m10 is non-zero. A single token is transported cyclically between the “active” and “inactive” entities. The activity of processes p6 and p7 is specified as:

```
if (rand() < 0.5) {
    return true;
} else {
    return false;
}
```

As a result, process p3 is active randomly and not more often than every second step of the simulation.

The CSML code for this model is presented below:

```
  <?xml version="1.0" encoding="UTF-8"?>
<csml:model
     xmlns:csml="http://www.csml.org/csml/version1"
     majorVersion="1"  minorVersion="9">
 <csml:unitdefs/>
 <csml:net>
  <csml:entity label="e6" name="e6" type="discrete">
   <csml:parameter label="m6" type="Integer" initialValue="0"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="494.0 363.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:entity label="e5" name="e5" type="discrete">
   <csml:parameter label="m5" type="Integer" initialValue="10"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="206.0 363.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:entity label="e11" name="inactive" type="discrete">
   <csml:parameter label="m11" type="Integer" initialValue="0"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="386.0 111.0"/>
    </csml:figure>
    <csml:text name="name" location="339.0 113.0"
     arrangement="None"      offsets="-36.5 0.0"/>
   </csml:graphics>
  </csml:entity>
  <csml:entity label="e10" name="active" type="discrete">
   <csml:parameter label="m10" type="Integer" initialValue="1"
     minimumValue="0" maximumValue="infinite"/>
   <csml:graphics>
    <csml:figure>
     <csml:discreteEntity location="386.0 255.0"/>
    </csml:figure>
    <csml:text name="name" location="336.0 257.0"
     arrangement="None"      offsets="-44.5 0.0"/>
   </csml:graphics>
  </csml:entity>
```

```
<csml:process label="p5" name="p3" type="discrete">
 <csml:simulationCondition>
  <csml:priority value="0"/>
  <csml:firing type="Boolean" value="true" firingStyle="and"/>
  <csml:delay type="Long" value="1.0" delayStyle="delay"/>
  <csml:calc calcStyle="add"/>
  <csml:kinetic type="Double" value="1" kineticStyle="custom">
   <csml:parameter name="custom" value="1"/>
  </csml:kinetic>
 </csml:simulationCondition>
 <csml:function>
  <csml:connector label="c10" name="c6" type="process" from="e5"
   to="p5" linestyle="straight" supportpath="false">
   <csml:firing type="Double" value="0"
   connectorFiringStyle="threshold"/>
   <csml:kinetic/>
   <csml:graphics>
    <csml:figure>
     <csml:processConnector points="228.0 374.0 378.0 374.0"/>
    </csml:figure>
    <csml:targetArrow>
     <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
         pointsInAbsoluteCoordinate="378.0 371.0 378.0 377.0
         386.0 374.0"/>
    </csml:targetArrow>
   </csml:graphics>
  </csml:connector>
  <csml:connector label="c8" name="c8" type="association"
   from="e10" to="p5" linestyle="straight" supportpath="false">
   <csml:firing type="Double" value="0"
   connectorFiringStyle="threshold"/>
   <csml:kinetic/>
   <csml:graphics>
    <csml:figure>
     <csml:associationConnector points="389.0 274.0 361.0 301.5
         390.4438598298215 360.8338387479737"/>
    </csml:figure>
    <csml:targetArrow>
     <csml:associationConnectorArrow points="0.0 0.0 0.0 6.0 8.0
         3.0" pointsInAbsoluteCoordinate="393.13117029933136
         359.5002861841568 387.7565493603116 362.16739131179065
         393.99999999999994 368.0"/>
    </csml:targetArrow>
   </csml:graphics>
  </csml:connector>
  <csml:connector label="c9" name="c7" type="process" from="p5"
   to="e6" linestyle="straight" supportpath="false">
   <csml:firing type="Double" value="0"
   connectorFiringStyle="threshold"/>
   <csml:kinetic/>
   <csml:graphics>
    <csml:figure>
     <csml:processConnector points="408.0 374.0 486.0 374.0"/>
    </csml:figure>
    <csml:targetArrow>
     <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
         pointsInAbsoluteCoordinate="486.0 371.0 486.0 377.0
         494.0 374.0"/>
    </csml:targetArrow>
   </csml:graphics>
```

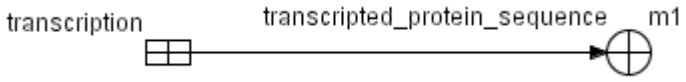
```
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    <csml:figure>
     <csml:discreteProcess location="385.5 368.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:process>
  <csml:process label="p7" name="p7" type="discrete">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" firingStyle="and">
     <csml:script><![CDATA[if (rand() < 0.5) {  return true; } else
           {     return false; }]]>
     </csml:script>
    </csml:firing>
    <csml:delay type="Long" value="1.0" delayStyle="delay"/>
    <csml:calc calcStyle="add"/>
    <csml:kinetic type="Double" value="1" kineticStyle="custom">
     <csml:parameter name="custom" value="1"/>
    </csml:kinetic>
   </csml:simulationCondition>
   <csml:function>
    <csml:connector label="c13" name="c13" type="process"
     from="e10"  to="p7" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="389.0 258.0
           336.65685424949237 205.65685424949237"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="334.5355339059327
           207.778174593052 338.778174593052 203.5355339059327
           331.0 199.99999999999997"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
    <csml:connector label="c14" name="c14" type="process" from="p7"
     to="e11" linestyle="straight" supportpath="false">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic/>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="332.0 188.0
           383.39254778542636 135.70582856921527"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="381.25286207197064
           133.60303398875016 385.5322334988821 137.8086231496804
           389.0 130.0"/>
      </csml:targetArrow>
```

```
    </csml:graphics>
   </csml:connector>
  </csml:function>
  <csml:biological>
   <csml:effectList/>
  </csml:biological>
  <csml:graphics>
   <csml:figure>
    <csml:discreteProcess location="313.5 187.5"/>
   </csml:figure>
  </csml:graphics>
 </csml:process>
 <csml:process label="p6" name="p6" type="discrete">
  <csml:simulationCondition>
   <csml:priority value="0"/>
   <csml:firing type="Boolean" firingStyle="and">
    <csml:script><![CDATA[if (rand() < 0.5) {
    return true;
    } else {
    return false;
    }]]>
    </csml:script>
   </csml:firing>
   <csml:delay type="Long" value="1.0" delayStyle="delay"/>
   <csml:calc calcStyle="add"/>
   <csml:kinetic type="Double" value="1" kineticStyle="custom">
    <csml:parameter name="custom" value="1"/>
   </csml:kinetic>
  </csml:simulationCondition>
  <csml:function>
   <csml:connector label="c11" name="c11" type="process"
    from="e11"  to="p6" linestyle="straight" supportpath="false">
    <csml:firing type="Double" value="0"
    connectorFiringStyle="threshold"/>
    <csml:kinetic/>
    <csml:graphics>
     <csml:figure>
      <csml:processConnector points="405.0 130.0
           457.34314575050763 182.34314575050763"/>
     </csml:figure>
     <csml:targetArrow>
      <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="459.4644660940673
           180.221825406948 455.221825406948 184.4644660940673
           463.00000000000006 188.00000000000003"/>
     </csml:targetArrow>
    </csml:graphics>
   </csml:connector>
   <csml:connector label="c12" name="c12" type="process" from="p6"
    to="e10" linestyle="straight" supportpath="false">
    <csml:firing type="Double" value="0"
    connectorFiringStyle="threshold"/>
    <csml:kinetic/>
    <csml:graphics>
     <csml:figure>
      <csml:processConnector points="463.0 200.0
           410.65685424949237 252.34314575050763"/>
     </csml:figure>
     <csml:targetArrow>
```

```
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="412.77817459305203
           254.46446609406726 408.53553390593277
           250.22182540694797 405.0 258.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    <csml:figure>
     <csml:discreteProcess location="458.0 188.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:process>
  <csml:figure>
   <csml:roundBounds depth="1" location="397.0 86.0" size="180.0
     360.0" fillColor="192 192 192 255" outlineColor="0 0 0 0"
     arcRatio="20"/>
  </csml:figure>
 </csml:net>
 <csml:simulation enhancedFiring="true" samplingInterval="1.0"
     simulationTime="2000.0" logUpdateInterval="1.0"
     plotUpdateInterval="1.0"/>
</csml:model>
```

5.4 Generic Example

Generic entities are of many value types. For example, you can use an entity that holds a String variable to describe a translation/transcription process with a generic process.



In this example, a script shown below is assigned to the Updater property of the connector:

```
total_sequence =
"mapqfeyhqsvqqkveydwerpilqgenyramlaamfavtelafrpacyalirlt";

current_position = m1.length() + 1;
m1 = total_sequence.substring(0, current_position);

return m1;
```

```
<?xml version="1.0" encoding="UTF-8"?>
<csml:model
     xmlns:csml="http://www.csml.org/csml/version1"
majorVersion="1" minorVersion="9">
 <csml:unitdefs/>
 <csml:net>
```

```
  <csml:entity label="e1" name="transcripted_protein_sequence"
     type="generic">
   <csml:parameter label="m1" type="String" initialValue=""/>
   <csml:graphics>
    <csml:figure>
     <csml:genericEntity location="381.0 171.0"/>
    </csml:figure>
   </csml:graphics>
  </csml:entity>
  <csml:process label="p1" name="transcription" type="generic">
   <csml:simulationCondition>
    <csml:priority value="0"/>
    <csml:firing type="Boolean" value="true" firingStyle="and"/>
    <csml:delay type="Long" value="1.0" delayStyle="delay"/>
    <csml:calc calcStyle="update"/>
    <csml:kinetic type="Double" value="1"
     kineticStyle="connectorcustom"/>
   </csml:simulationCondition>
   <csml:function>
    <csml:connector label="c1" name="c1" type="process" from="p1"
     to="e1" linestyle="straight" supportpath="true">
     <csml:firing type="Double" value="0"
     connectorFiringStyle="threshold"/>
     <csml:kinetic>
      <csml:parameter name="custom" value="total_sequence =
&amp;quot;mapqfeyhqsvqqkveydwerpilqgenyramlaamfavtelafrpacyalirlt&amp;q
     uot;;current_position = m1.length() + 1;m1 =
     total_sequence.substring(0, current_position);return m1;"/>
     </csml:kinetic>
     <csml:graphics>
      <csml:figure>
       <csml:processConnector points="151.0 182.0 373.0 182.0"/>
      </csml:figure>
      <csml:targetArrow>
       <csml:processConnectorArrow points="0.0 0.0 0.0 6.0 8.0 3.0"
           pointsInAbsoluteCoordinate="373.0 179.0 373.0 185.0
           381.0 182.0"/>
      </csml:targetArrow>
     </csml:graphics>
    </csml:connector>
   </csml:function>
   <csml:biological>
    <csml:effectList/>
   </csml:biological>
   <csml:graphics>
    <csml:figure>
     <csml:genericProcess location="129.0 176.0" fillColor="0 0 0
     0" outlineColor="0 0 0 255"/>
    </csml:figure>
   </csml:graphics>
  </csml:process>
 </csml:net>
 <csml:simulation enhancedFiring="true" samplingInterval="0.1"
     simulationTime="2000.0" logUpdateInterval="1.0"
     plotUpdateInterval="1.0"/>
</csml:model>
```