Proceedings
of the 3rd International Modelica Conference,
Linköping, November 3-4, 2003,
Peter Fritzson (editor)

Emma Larsdotter Nilsson and Peter Fritzson
PELAB, Linköping University : BioChem - A Biological and Chemical Library for Modelica
pp. 215-220

Paper presented at the 3rd International Modelica Conference, November 3-4, 2003, Linköpings Universitet, Linköping, Sweden, organized by The Modelica Association and Institutionen för datavetenskap, Linköpings universitet

All papers of this conference can be downloaded from
http://www.Modelica.org/Conference2003/papers.shtml

BioChem – A Biological and Chemical Library for Modelica

Emma Larsdotter Nilsson and Peter Fritzson
Linköpings universitet
PELAB – Programming Environment Laboratory
Department of Computer and Information Science
SE-581 83 Linköping, Sweden
{emmni, petfr}@ida.liu.se

Abstract

Many biological, biochemical and chemical systems have been mathematically defined for decades. As laboratory techniques are becoming increasingly sophisticated, more systems can be mathematically defined. But sophisticated techniques usually means more expensive and time-consuming. Simulation and modeling tools have today therefore become a very important part of biological and chemical research. In this paper the advancement on developing a library for simulation of cellular pathways in the Modelica language will be presented.

Keywords: Modelica, biological systems, chemical reactions, cellular pathways, SBML.

1 Biological Systems

All living things can be seen as systems. The prey-predator relation between foxes and rabbits, the cycle of energy-forest, the life of a bacterium or the human body are all more or less complex examples of biological systems surrounding us. Many of these systems are easy to model or simulate, their mathematical definitions have been known for years, if not decades. Other systems such as the human body or bacteria's intercellular processes, we don't know so much about, yet.

2 Cells and Cellular Pathways

One type of systems that we are just about to learn more about is the cell. Cells are the basic building blocks of all living organisms. No matter if the cells are part of a multicellular organism or constitute unicellular organisms the processes inside them do not differ that much. A cell's metabolism involves the uptake, decomposition, and rebuilding of different compounds and can be seen as a complex web or graph. The nodes are the different substances and the edges are the reactions that transform one substance to another. These complex webs, consisting of up to a couple of hundred substances and more than twice as many reactions are referred to as cellular pathways. Some of the reactions in these pathways are already well defined, while some are not even discovered yet.

3 Why Modelica

Many biological, biochemical, and chemical systems have been mathematically defined for decades. As laboratory techniques are becoming more sophisticated, even more systems are defined and sometimes redefined. Better laboratory techniques also make it possible to analyze larger and more complex systems than before. However better techniques can also mean more expensive and sometimes more time-consuming analyses. A good modeling/simulation tool can in many cases extensively cut the cost and time.

Due to being equation-based Modelica is very suitable for modeling of biological, biochemical, and chemical systems. One major benefit is that the classes are acausal and thereby adapt to more than one data flow context [1]. The complexity of these types of systems will not be a problem, Modelica's strength as a modeling language for complex technical systems is well proven [2]. Another benefit of Modelica is that it is possible to model both discrete and continuous systems as well as hybrids thereof. Especially hybrid systems are quite common in the biological/chemical area. Modelica's strong software component model makes the language ideally suited as an architectural description language for complex systems, such as complex pathway models. Finally, the use of Mathematica [6] notebooks and commands for the simulations makes Modelica easy, even for non-computer science user. This is very important since most biologists and chemists have none or very little experience in computer science.

4 BioChem Package

The work of building a Modelica library for cellular systems has only just started. So far the main effort has been to develop classes for nodes and chemical reactions frequently occurring within cellular pathways.

4.1 Package structure

The structure of the package is shown in Figure 1.

```
package BioChem
  package BioChemUnits "Types and their units"
  end BioChemUnits;
  package Icons "Definitions of icons"
  end Icons;
  package Interfaces
    "Definitions of interfaces"
  end Interfaces;
  package Compartments
    "Definitions of compartments"
  end Compartments;
  package NodeElements
    package Nodes "Substance nodes"
    end Nodes;
    package SpecialNodes "Sources and Sinks"
    end SpecialNodes;
  end NodeElements;
  package Reactions
    package BasicReactions
      "Stochiometric Reactions"
    end BasicReactions;
    package SBMLReactions "SBML Reactions"
      package GenericSBML "Generic reactions"
      end GenericSBML;
      package IrreversibleSBML
        "Irreversible reactions"
      end IrreversibleSBML;
      package ReversibleSBML
        "Reversible reactions"
      end ReversibleSBML;
      package MultiSBML
        "Multi reactant SBML reactions"
      end MultiSBML;
    end SBMLReactions;
  end Reactions;
end BioChem;
```

Figure 1. Structure of the `BioChem` package.

4.2 Package `Icons`

The package `BioChem.Icons` contains icons used in the drag-and-drop interface in MathModelica. Icons for substance nodes are represented by circles, reactions are represented by uni and bidirectional arrows, and all other chemical signs and operators are represented by their standard symbols.

4.3 Package `Interfaces`

The package `BioChem.Interfaces` contains basic objects such as connectors and partial models used for most components in the `BioChem` package. The `ReactionConnection` (Figure 2) is the connector used for connecting the different components in the model.

```
connector ReactionConnection
  extends Icons.BlueX;
  BioChemUnits.Concentration c;
  flow BioChemUnits.MolarFlowRate r;
end ReactionConnection;
```

Figure 2. The connector `ReactionConnection`.

The connector is used on all connectable ends of reaction arrows, connectable parts of special nodes and signs, and for all normal nodes. All normal nodes are represented by the partial model `NodeConnections` (Figure 3), which contains eight connectors in order to make it easier to connect more than one reaction to a node.

```
partial model NodeConnections
  ReactionConnection rc_1;
  ReactionConnection rc_2;
  ReactionConnection rc_3;
  ReactionConnection rc_4;
  ReactionConnection rc_5;
  ReactionConnection rc_6;
  ReactionConnection rc_7;
  ReactionConnection rc_8;
end NodeConnections;
```

Figure 3. The partial model `NodeConnections` used for all normal nodes in the pathway web.

4.4 Package `Compartments`

In order to be able to control the environment of the reaction during a simulation a chemical reaction must take place in a restricted screened-off volume. The `Compartments` package contains models for all the different types of compartments in a cell.

4.5 Package `NodeElements`

The package `BioChem.NodeElements.Nodes` contains the different types of nodes that can appear in a metabolic pathway. The nodes must have some attributes corresponding to the properties studied during simulation of a metabolic pathway. The name of the substance and the surrounding compartment, the electrical charge (in case of the substance being an ion) and the amount of the substance and the flow through the node are such basic attributes.

The partial model `BasicNode` (Figure 4) contains all these basic attributes along with some equations needed for calculating the flow and the concentration of the substance.

```
partial model BasicNode
  extends Interfaces.NodeConnections;
  parameter String substanceName("");
  outer parameter String compartmentName;
  outer parameter BioChemUnits.Volume V_0;
  parameter BioChemUnits.Charge charge = 0;
  parameter BioChemUnits.Concentration
    tolerance = -1e-6;
  outer BioChemUnits.Volume V;
  BioChemUnits.Concentration c;
  BioChemUnits.MolarFlowRate r_net;
equation
  r_net = rc_1.r + rc_2.r + rc_3.r + rc_4.r +
          rc_5.r + rc_6.r + rc_7.r + rc_8.r;
  c = rc_1.c; rc_1.c = rc_2.c;
  rc_2.c = rc_3.c; rc_3.c = rc_4.c;
  rc_4.c = rc_5.c; rc_5.c = rc_6.c;
  rc_6.c = rc_7.c; rc_7.c = rc_8.c;
end BasicNode;
```

Figure 4. The partial model for the properties of a node.

In most cases the model `Node` (Figure 5) is used to represent a substance. In this type of node the concentration of the substance is allowed to change throughout the simulation without any restrictions. The total amount of substance in the node is though conserved at any time.

```
model Node
  extends BasicNode;
  extends Icons.WhiteNode;
  parameter BioChemUnits.Concentration
    c_0 = 1.0;
  BioChemUnits.AmountOfSubstance
    n(start = c_0 * V_0;
equation
  der(n) = r_net;
  c = n/V;
  assert(c > tolerance,
         "Node: c is  negative!");
end Node;
```

Figure 5. The most commonly used node model.

All nodes except the node used for static reactions have an assert statement that checks that the concentration never falls lower than the tolerance below zero. If the concentration goes more than the tolerance below zero during simulation an error will be generated.

```
model FixedConcentrationNode
  extends NonStaticSubstanceNode;
  extends Icons.YellowNode;
  parameter BioChemUnits.Concentration
    c_fixed = 1;
  BioChemUnits.AmountOfSubstance n;
equation
  c = c_fixed;
  c = n/V;
  assert(c > tolerance,
    "FixedConcentrationNode: c is negative!");
end FixedConcentrationNode;
```

Figure 6. Model for nodes with fixed concentration.

Under some circumstances it is desirable to keep the concentration of a substance at a fixed value during the whole simulation. For these cases the model `FixedConcentrationNode` (Figure 6) is used to represent the substance node. The total amount of substance in the node is still conserved at any time.

Under some circumstances it is desirable to statically pump a substance into a node from a sink or from a node into a source (Figure 7). The pump (flow) rate to or from the node is in most simulations kept at a constant level, but it is also possible to change the flow during a simulation.

```
model FixedSink
  extends Icons.YellowNode;
  extends Icons.Sink;
  parameter BioChemUnits.MolarFlowRate
    sinkFlow = 1;
  Interfaces.ReactionConnection
    sinkConnection;
equation
  sinkConnection.r = -sinkFlow;
end FixedSink;

model FixedSource
  extends Icons.YellowNode;
  extends Icons.Source;
  parameter BioChemUnits.MolarFlowRate
    sourceFlow = 1;
  Interfaces.ReactionConnection
    sourceConnection;
equation
  sourceConnection.r = sourceFlow;
end FixedSource;
```

Figure 7. Models for source and sink nodes.

4.6 Package `Reactions`

The package `BioChem.Reactions.BasicReactions` contains different types of elementary reactions needed in a metabolic pathway. An elementary reaction is a reaction that cannot be broken down into a simpler reaction.

For a reaction to take place there need to be at least one starting substance, the substrate, and one ending substance, the product. The substrates appear on the left side, and the products on the right side of the reaction arrow in a reaction equation. A reaction can be either irreversible, going in one direction, or reversible, going in both directions. A reaction coefficient determines the speed with which the substrate is turned into the product. The reversible reaction can be seen as two irreversible reactions and have therefore got one forward and one backward reaction coefficients.

All reactions inherit some basic attributes, such as concentration of one substrate and one product, forward reaction coefficient, and the maximum speed of the reaction (maximum volumetric reaction rate) along with some basic equations from the partial model `BasicReaction` (Figure 8).

```
partial model BasicReaction
  parameter String reactionName("");
  parameter BioChemUnits.ReactionCoef k1=1;
  parameter BioChemUnits.StoichiometricCoef
    n_S1=1;
  parameter BioChemUnits.StoichiometricCoef
    n_P1=1;
  BioChemUnits.Concentration c_S1;
  BioChemUnits.Concentration c_P1;
  BioChemUnits.VolumetricReactionRate v;
  outer BioChemUnits.Volume V;
  Interfaces.ReactionConnection rc_S1;
  Interfaces.ReactionConnection rc_P1;
equation
  c_S1 = rc_S1.c;
  c_P1 = rc_P1.c;
end BasicReaction;
```

Figure 8. The partial model for elementary reactions.

The partial models for irreversible (`OneWayReaction`) and reversible (`TwoWayReaction`) reactions are shown in Figure 9.

```
partial model OneWayReaction
  extends BasicReaction;
equation
  rc_S1.r = n_S1*v*V;
  rc_P1.r = -n_P1*v*V;
end OneWayReaction;

partial model TwoWayReaction
  extends BasicReaction;
  parameter BioChemUnits.ReactionCoef k2=1;
equation
  rc_S1.r = n_S1*v*V;
  rc_P1.r = -n_P1*v*V;
end OneWayReaction;
```

Figure 9. The partial models for irreversible and reversible reactions.

The foundation of chemical kinetics is the so called law of mass action, which states that the rate of an elementary reaction is proportional to the amount of substance present. For the simplest elementary reaction:

$$n_A A \xrightarrow{k} n_B B$$

k is the reaction coefficient, and n_A and n_B are the stoichiometric coefficients for the substances A and B, respectively. The reaction rate (v) for the reaction is expressed as:

$$v = k[A]^{nA}$$

where $[A]$ is the concentration of substrate A. The Modelica code for the uni-uni irreversible reaction is shown in Figure 10.

```
model OneWayReactionUniUni
  extends OneWayReaction;
  extends Icons.Irreversible1to1Arrow;
equation
  v = k1*c_S1^(n_S1);
end OneWayReactionUniUni;
```

Figure 10. The model for uni-uni irreversible reactions.

The uni-uni irreversible reaction is quite simple. A more thorny elementary reaction is the bi-bi reversible reaction:

$$n_A A + n_B B \underset{k_2}{\overset{k_1}{\rightleftharpoons}} n_C C + n_D D$$

This reaction has two substrates, A and B, which are turned into two products, C and D, under the influence of the forward reaction coefficient k_1. The products are also reversibly turned into the substrates under the influence of the backward reaction coefficient k_2. The reaction rate (v) for the reaction is expressed as:

$$v = k_1[A]^{nA}[B]^{nB} - k_2[C]^{nC}[D]^{nD}$$

where $[A]$, $[B]$, $[C]$, and $[D]$ are the concentrations of the substances A, B, C, and D, respectively. The Modelica code for the bi-bi reversible reaction is shown in Figure 11.

```
model TwoWayReactionBiBi
  extends TwoWayReaction;
  extends Icons.Reversible2To2Arrow;
  parameter BioChemUnits.StoichiometricCoef
    n_S2=1;
  parameter BioChemUnits.StoichiometricCoef
    n_P2=1;
  BioChemUnits.Concentration c_S2;
  BioChemUnits.Concentration c_P2;
  Interfaces.ReactionConnection rc_S2;
  Interfaces.ReactionConnection rc_P2;
equation
  c_S2 = rc_S2.c;
  c_P2 = rc_P2.c;
  rc_S2.r = n_S2*v*V;
  rc_P2.r = -n_P2*v*V;
  v = (k1*c_S1^(n_S1))*c_S2^(n_S2) -
      (k2*c_P1^(n_P1))*c_P2^(n_P2);
end TwoWayReactionBiBi;
```

Figure 11. The model for bi-bi reversible reactions.

Reactions can also be under the influence of an inhibitor or an activator. An inhibitor is a substance that through its presence slows the reaction down, but is neither consumed nor changed during the process. The basic partial model for an inhibited reaction inherits properties from the basic irreversible reaction and is thereby an irreversible reaction. The difference is the addition of the inhibitor, and some equations making sure that the inhibitor is not consumed during simulation (Figure 12).

```
partial model InhibitedReaction
  extends OneWayReaction;
  extends Icons.InhibitorSign;
  extends Icons.SingleArrowModulation;
  BioChemUnits.Concentration c_I;
  Interfaces.ReactionConnection rc_I;
equation
  c_I = n_I.c;
  rc_I.r = 0;
end InhibitedReaction;
```

Figure 12. The partial model for inhibition reactions.

For the elementary inhibited reaction:

$$n_A A \xrightarrow{k(I)} n_B B + n_C C$$

k(I) is the reaction coefficient, and n_A, n_B and n_C are the stoichiometric coefficients for substance *A*, *B*, and *C*, respectively. The reaction rate (*v*) for the reaction is expressed as:

$$v = k(I)[A]^{nA}[I]^{-1}$$

where [*A*] and [*I*] are the concentrations of substrate *A* and the inhibitor *I*. The Modelica code for the uni-bi irreversible inhibition reaction is shown in Figure 13.

```
model InhibitedReactionUniBi
  extends InhibitedReaction;
  extends Icons.Irreversible1To2Arrow;
  parameter StoichiometricCoef n_P2=1;
  BioChemUnits.Concentration c_P2;
  Interfaces.ReactionConnection rc_P2;
equation
  c_P2 = rc_P2.c;
  rc_P2.r = -n_P2*v*V;
  v = k1/c_I*c_S1^(n_S1);
end InhibitedReactionUniBi;
```

Figure 13. The model for uni-bi inhibition reactions.

An activator is a substance that through its presence speeds up the reaction, but is neither consumed nor changed during the process. The basic partial model for an activated reaction looks just like the partial model for the inhibited reaction (Figure 14).

```
partial model ActivatedReaction
  extends OneWayReaction;
  extends Icons.ActivatorSign;
  extends Icons.SingleArrowModulation;
  BioChemUnits.Concentration c_Ac;
  Interfaces.ReactionConnection rc_Ac;
equation
  c_Ac = n_Ac.c;
  rc_Ac.r = 0;
end ActivatedReaction;
```

Figure 14. The partial model for activation reactions.

For the elementary activated reaction:

$$n_A A + n_B B \xrightarrow{k(A_C).} n_C C$$

k(Ac) is the reaction coefficient, and n_A, n_B and n_C are the stoichiometric coefficients for substance *A*, *B*, and *C*, respectively. The reaction rate (*v*) for the reaction is expressed as:

$$v = k(Ac)[Ac][A]^{nA}[B]^{nB}$$

where [*A*], [*B*] and [*Ac*] are the concentrations of substrate *A* and *B*, and the activator *Ac*. The Modelica code for the bi-uni irreversible activation reaction is shown in Figure 15.

```
model ActivatedReactionBiUni
  extends ActivatedReaction;
  extends Icons.Irreversible2To1Arrow;
  parameter BioChemUnits.StoichiometricCoef
    n_S2=1;
  BioChemUnits.Concentration c_S2;
  Interfaces.ReactionConnection rc_S1;
equation
  c_S2 = rc_S2.c;
  rc_S2.r = n_S2*v*V;
  v = k1*c_Ac*c_S1^(n_S1)*c_S2^(n_S2);
end ActivatedReactionBiUni;
```

Figure 15. The model for bi-uni activation reactions.

4.7 SBML

The Systems Biology Markup Language (SBML) is a computer-readable format for representing models of biochemical reaction networks. SBML is applicable to metabolic networks, cell-signaling pathways, genomic regulatory networks, and many other areas in systems biology [8].

In order to make models created in Modelica interchangeable with other biological or chemical simulation and modeling tools a two-way translator between Modelica and SBML is under development. The package `BioChem.Reactions.SBMLReactions` contains reactions specified in the second release (Level 2) of SBML [3, 5].

5 Current work

To make it even easier for biologist/chemist to use Modelica for modeling and simulation a drag-and-drop graphical interface is currently being developed for MathModelica [4, 7]. Current work also focuses on building pathway models using the classes in the `BioChem` package.

Acknowledgments

The authors whish to thank Eva-Lena Lengquist Sandelin, Linköpings universitet for valuable feedback, and MathCore Engineering AB for supplying the MathModelica tool. Emma Larsdotter Nilsson was in part funded by the Swedish National Graduate School in Computer Science (CUGS).

References

[1] Fritzson, P. Principles of Object-Oriented Modeling and Simulation with Modelica. IEEE Press and Wiley. 2003.

[2] Fritzson P and P. Bunus. Modelica – A General Object-Oriented Language for Continuous and Discrete-Event System Modeling and Simulation. (In Proceedings of the 35th Annual Simulation Symposium. San Diego, California, April 14-18, 2002. IEEE Press. 2002.

[3] Finney, A. and M. Hucka. Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. http://www.cds.caltech.edu/erato/sbml/docs/. 2003.

[4] Fritzson P, J. Gunnarsson, and M. Jirstrand. MathModelica – An Extensible Modeling and Simulation Environment with Integrated Graphics and Literate Programming, In Proceedings of 2nd International Modelica Conference Munich, March 2002 (www.modelica.org). 2002.

[5] Hucka, M. et al. Systems Biology Markup Language (SBML) Level 1: Structures and Facilities for Model Definitions. http://www.cds.caltech.edu/erato/sbml/docs/. 2003.

[6] Wolfram, S. The Mathematica Book. Wolfram Media. 1997.

[7] MathModelica web site. www.mathcore.com.

[8] SBML web site. www.sbml.org.