

Biochemical and Metabolic Modeling and Simulation with Modelica

Emma Larsdotter Nilsson and Peter Fritzson

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

Abstract

In the drug industry, the later a substance is discharged from the drug development pipeline, the higher the financial cost. In order to reduce the number of lead compounds a number of computerized systems have been suggested, and in most of these systems modeling and simulation of the lead compound's effects on different metabolic pathways are essential. In these systems, substances that are expected to be harmful or lethal can be removed at an early stage and a reduced number of lead compounds can be chosen for the concluding tests.

*Given Modelica's previous success with modeling and simulation of huge and complex systems it is likely that it will also be suitable for modeling, simulation, and visualization of metabolic pathway systems, e.g., those systems used in the drug industry. A Modelica library designed to be used for modeling, simulation, and visualization of metabolic pathways is the special-purpose library *Metabolic*, an extension of the abstract Modelica library *BioChem*.*

Keywords: Pathway modeling, pathway libraries, template models, *BioChem*, *Metabolic*.

1. Introduction

There is currently a great interest in the development of novel analytical technologies for rapid screening of biological dysfunctions in pharmaceutical and clinical applications. In the drug industry the later a substance is discharged from the drug development pipeline, the higher the financial cost. Not only is it costly to test many substances, the price of the tests increase along the development pipeline. Minimizing the number of substances that are fully tested, i.e., becoming lead compounds, is therefore one of the most important aims of all pharmaceutical discovery programs [1].

In order to reduce the number of lead compounds a number of systems have been suggested, out of which some have been realized [2-5]. In most of these systems modeling and simulation of the lead compound's effects on different metabolic pathways are included. A metabolic pathway can be seen as a complex web made up of several hundred substances and more than twice as many reactions. Substances that are expected to in-

teract in a harmful or lethal way with essential metabolic pathways can be removed at an early stage and a reduced number of promising lead compounds can be chosen for the concluding tests.

In theory, simulations of a single or a few interconnected pathways can be useful when the metabolic pathways under study are relatively isolated from each other. In practice, even the simplest and most well-studied metabolic pathways can exhibit complex behavior due to connections in-between different levels of the whole-cell or whole-organism system.

Given previous success of the equation-based object-oriented language Modelica [6-9] with modeling and simulation of huge and complex technical, physical, electrical, and thermodynamic systems it is likely that it will also be a suitable language for modeling, simulation, and visualization of metabolic pathway systems. A prototype system for modeling, simulation, and visualization of metabolic pathways implemented in Modelica, i.e., *BioChem*, proved Modelica to be suitable for such applications [10, 11]. The prototype has since been further developed and the original *BioChem* library has been split into two. What remains of *BioChem* is now an abstract general-purpose library for biological and biochemical systems. This library is not intended, nor designed, to be used directly for creating models and running simulations. The intention with the library is to provide some common basic behaviors, attributes, and environmental properties to be used in special-purpose libraries.

The metabolic part of the old *BioChem* library has been moved to a new library, called *Metabolic*. *Metabolic* which is a special-purpose library that extends, i.e., inherits, the basic behaviors, attributes, and environmental properties provided in *BioChem*. The *Metabolic* library is designed to be used for modeling, simulation, and visualization of metabolic pathways. The components specified in the library describe basic substances and general reactions that are common in metabolic pathways.

Provided with the substances and reactions in *Metabolic*, a library of metabolic pathway templates can be built. The idea is that these general model-templates can easily be extended and adapted to concrete species-specific models which can then be used in simulations of metabolic pathways.

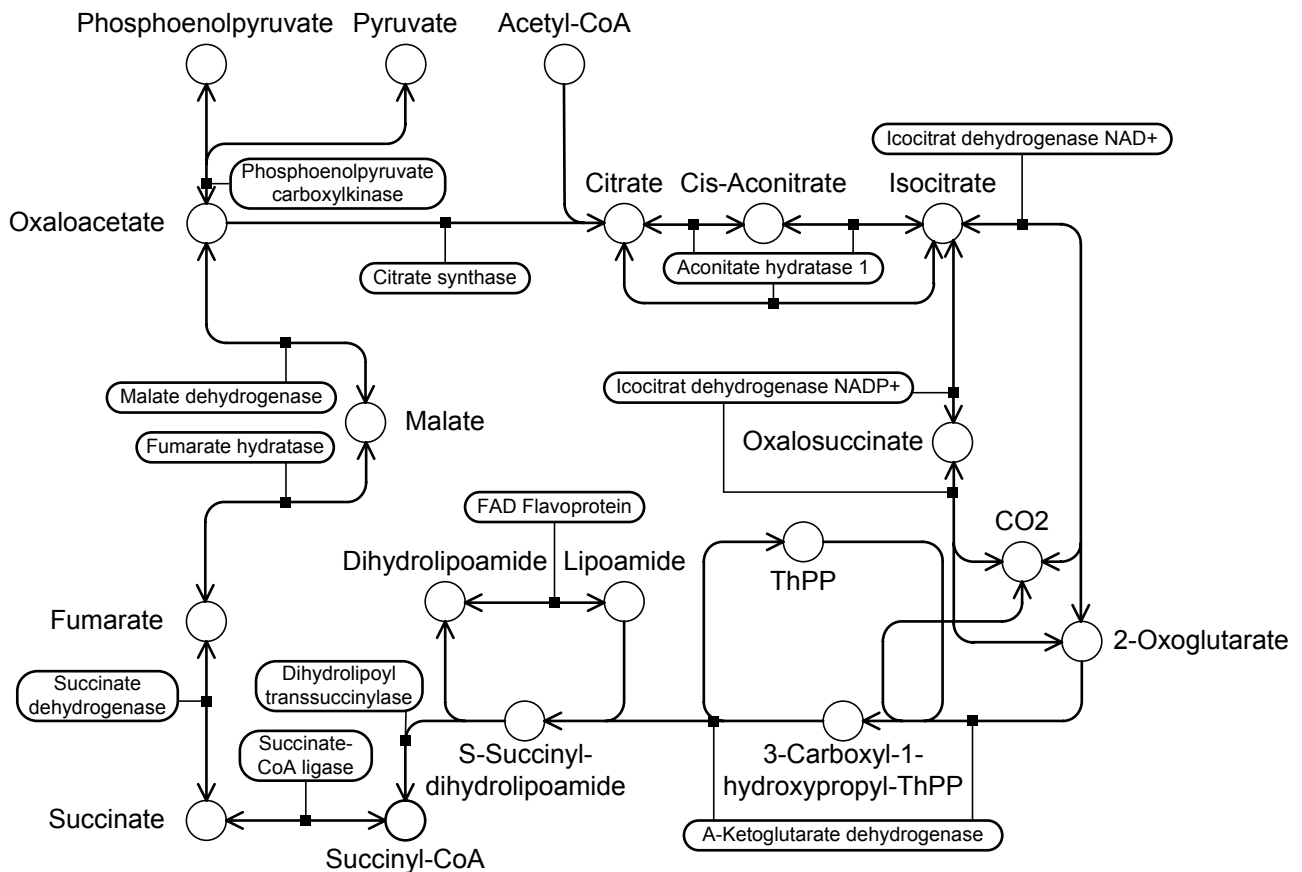


Figure 1. The metabolic pathway *Citrate cycle* for Bakers yeast (*Saccharomyces cerevisiae*). The substances that participate in the pathway are represented by sheres, while the enzymes that control the metabolic reactions are connected to the reaction arrows.

2. The Cell as a System

During the past ten to fifteen years the development and introduction of new analytical techniques in the area of biology and biochemistry have greatly increased the amount of experimental data obtained from experiments performed in the area. Automated DNA sequencing [12-14], microarray-analysis of gene expressions [15, 16], and protein profiling [17, 18] are just a few of the methods that have made a significant contribution to the extensive amount of data available.

The obtained data can be useful in modeling, simulation, and visualization of cellular processes starting with *genomics* [19] where the total genome is investigated in order to understand the significance of an individual gene, through *transcriptomics* [20] where change of individual transcript expressions are related to the total expression of RNA transcripts, through *proteomics* [21] where changes of individual protein concentrations are related to the total expression of proteins, and ending with *metabonomics* [1] where changes in metabolic profiles, i.e., metabolite concentrations, at an organism level, are of concern.

2.1 Whole-Cell System Levels

The cell itself is a huge system that easily can be divided into several subsystems. Looking only upon the processes that take place inside the cell the whole-cell system can be divided into three levels (Figure 2), i.e., cell division, gene expression, and metabolism. Each level might then, depending on the context, be divided further into sub-levels.

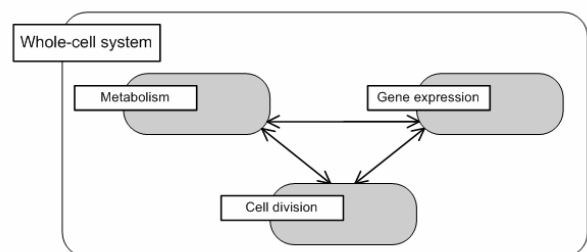


Figure 2. Levels in a whole-cell system.

2.2 The Metabolic Level

Metabolic processes usually consist of sequences of enzymatic steps, also called metabolic pathways [22-24]. The connection of all possible metabolic pathways

for a cell will result in a fully functional system level in the whole-cell system, i.e., the metabolic level.

A cell's metabolism involves the uptake, decomposition, and rebuilding of different compounds and can be seen as several complex webs transporting matter and energy, e.g. the Citrate cycle (Figure 1), the Starch and sucrose metabolism, the Glycolysis, and the Gluconeogenesis [24]. Most of the reactions in these pathways are, in one way or another, controlled by enzymes [25, 26], mostly proteins. Enzymes can either activate or inhibit the reaction in question and the amount of a protein in the cell is controlled by the expression of the gene that codes for that specific protein.

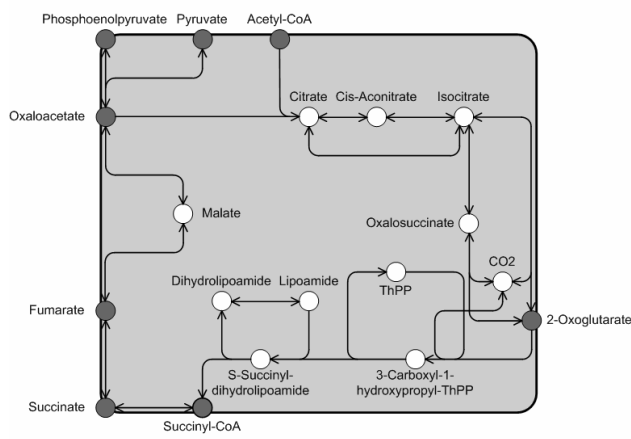


Figure 3. The metabolic pathway Citrate cycle for Baker's yeast (*Saccharomyces cerevisiae*) seen as a sub-system. Substances that are connection points to other metabolic pathways are represented by filled spheres, while substances that are internal with respect to the metabolic pathway are represented by unfilled spheres. (Compare to Figure 1.)

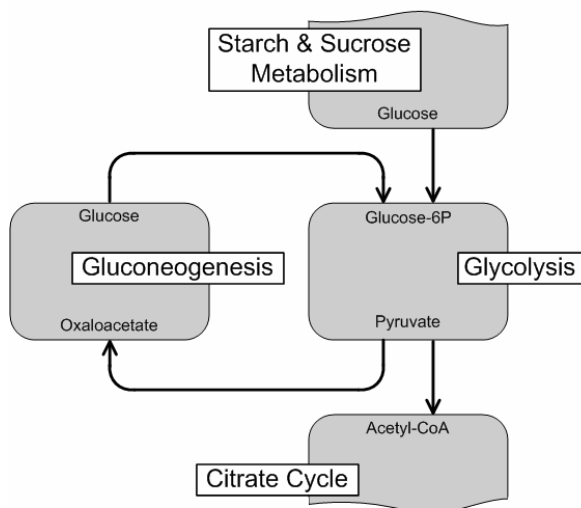


Figure 4. Interconnection of the four metabolic pathways, the Starch and sucrose metabolism, the Glycolysis, the Gluconeogenesis and the Citrate cycle. More pathways do connect to each one of the four pathways, but for simplicity these have been edited out.

Each metabolic pathway is highly compartmentalized (Figure 3) with a few in-flows and/or out-flows that can be connected to preceding and following metabolic pathways (Figure 4). For example, the Starch and Sucrose metabolism is a preceding pathway and the Citrate Cycle is a following pathway of the Glycolysis while the Gluconeogenesis is both a preceding and following pathway of the Glycolysis.

Many of the reactions that participate in these pathways are more or less the same in all cells, while others are highly dependent on the species, the type of cell, or even on the individual that the cell belongs to.

2.3 Databases Containing Metabolic Data

Much of the data regarding metabolic pathways obtained through experiments and analysis is accessible in different public and commercial reference databases. In order to be able to model metabolic pathways one needs to know the participating substances and the reactions in-between them. The organization of entire blocks of metabolic pathways can be found in human-curated maps in public databases, i.e., KEGG [27] and BioCarta's "Proteomic Pathway Project" [28]. The equations specifying the reactions can, however not be found in those maps. This information can instead be retrieved from databases that provide data on individual enzymatic reactions, i.e., BRENDA [29] and EMP [30], and in databases that provide data on multi-step metabolic pathways, i.e., MPW [31] and EcoCyc/HumanCyc [32].

When the modeled pathway is not fully mathematically defined it can be useful to employ enzyme databases that contain data that can not be displayed on maps or through equations. Unique identifiers, synonyms, enzyme classification, encoding genes, protein sequences, protein structure, motifs, other reactions that the enzyme is involved in, cross-references, references, and comments are just some of the information that can be found in these types of databases. The fastest way to retrieve data from all these databases is by using a database retrieval tool like DBGET [33]. One problem with these databases is that the data contained in different databases might be inconsistent.

Although all the above resources together represent a good general reference in the work of modeling and simulation of metabolic pathways, they also have significant limitations. The usually non species-specific information causes many errors and inconsistencies, and in many cases the amount of data that can be found for a pathway is not enough for building accurate pathway models [34]. Yet another problem with these databases is that the data contained in different databases might be inconsistent. But even with the mentioned limitation it is still possible to perform modeling and simulation of metabolic pathways with the information provided by the above resources.

3. Problem Description

3.1 Problem Context and Relevance

There is currently a great interest in the development of novel analytical technologies for rapid screening of biological dysfunctions in pharmaceutical and clinical applications. In the pharmaceutical industry the later a substance is discharged from the drug development pipeline, the higher the development cost. Not only is it costly to test many substances, the price of the tests increase along the development pipeline. Minimizing the number of substances that are fully tested, i.e., becoming lead compounds, is therefore one of the most important aims of all pharmaceutical discovery programs.

Until only a few years back most of the approaches to study biology had a reductionistic approach. Typically, researchers focused on one isolated component of a biological system at a time, e.g. a single gene, a set of genes, or a protein. The researchers then took all the component-data and tried to work up to the system level, independent of the system. The problem with this bottom-up approach was that the data from an individual component at a system level often was unconnected from the context in which it was originally gathered.

A new approach to study biological systems came with Systems Biology [35, 36] where researchers seek to integrate biological data that the new laboratory techniques provide in order to attempt to understand how large and complex biological systems function. The goal is to create a comprehensive and understandable model of the targeted system as a whole.

Up until Systems Biology was introduced, modeling and simulation of metabolic pathways had a single-level approach. As new research areas were introduced in system biology, additional system levels were revealed and could be incorporated in the model. One multi-level system approach to model metabolic pathways is metabolic reconstruction [37, 38]. The term refers to the process of deducing the metabolism and functional organization of an organism from its genetic sequence data supplemented by known biochemical and phenotypic data, or in practice to visually link metabolic blocks corresponding to the genetic component of the organism in wire diagrams. Several investigations have confirmed the power of these metabolic reconstruction models in predicting gene functions and finding novel pathways [39, 40].

There are also some severe limitations in the metabolic reconstruction methodology. The next step would be to create algorithms that are capable of inferring condition-specific functional blocks by analyzing the whole space of molecular interaction networks alongside high-throughput data. Several such algorithms have been recently described [41-43]. However, their applicability to the reconstruction of human and mammalian pathways has yet to be proven.

3.2 Open Research Problems

There are many open research questions in the area of systems biology. One of the greatest challenges in the area right now is to figure out which proteins interact with which reactions and then try to find the corresponding coding gene in the DNA for these proteins. The knowledge of which proteins control which metabolic processes is of great importance when modeling metabolic pathways. Some of the reactions in the metabolic pathways are already well-known as well as mathematically defined. Other parts of these pathways are more or less undetermined, ranging from not being fully mathematically defined to not being fully discovered yet.

Finding the mathematics behind the metabolic pathways, especially for those pathways involved in the xenobiotic metabolism, is of great importance for the pharmaceutical industry. In light of the rapid development of the new biology and the high cost of the development of a new drug, there is a need for a consistent framework for modeling, simulation, and visualization of metabolic pathways at all system levels, including the impact of xenobiotica.

4. Modelica

4.1 The language

Modelica [6-9] is one of the newer object-oriented equation-based languages, and was originally developed for hierarchical physical and technical modeling. Primarily, Modelica is a modeling language that allows the user to specify mathematical models of complex systems, but it is also an object-oriented equation-based programming language, oriented towards computational applications with high complexity requiring high performance. Modelica unifies and generalizes previous object-oriented modeling languages in the physical and technical area, and compared to other object-oriented languages for modeling and simulation the most important advantages are [44]:

- *Acausal modeling* is permitted since Modelica is based on ordinary differential equations and differential algebraic equations. This offers opportunities to reuse classes since a class can adapt to more than one data flow context.
- *The general type system* of Modelica unifies object-orientation, multiple inheritance, and generics templates within a single class construct. This facilitates the reuse of components and the evolution of models.
- *The multi-domain modeling capability* of Modelica, i.e., the possibility to describe and connect model components from several different domains within the same application model.

- *The strong software component model of Modelica has constructs for creating and connecting components. This makes the language ideally suited as an architectural description language for complex systems.*

4.2 Benefits of Using Modelica for Biochemical and Biological Systems

Biological and biochemical systems can often easily be described using mathematical relations and expressions. This makes the equation-based Modelica a suitable programming and modeling language for mathematical modeling of such systems. First of all, Modelica classes are acausal, i.e., can adapt to more than one data flow context [8], which is a great benefit when dealing with chemical reactions where the flow of matter can move in two directions.

The complexity of biological and biochemical models can be rather high, containing several hundreds of items. However, this will not be a problem since Modelica's strength as a modeling language for complex technical systems is well proven [7].

Moreover, Modelica's strong software component model also makes it ideal as an architectural description language for complex systems [7], e.g. metabolic pathway webs. It is also possible to model both discrete and continuous systems, as well as hybrids thereof [8]. Especially hybrid systems are quite common in the subject area of biology and biochemistry.

Finally, Modelica is an object-oriented language which makes it possible to reuse of models at both component and template levels and to add new system functions in modules. This will result in fewer, but more generic models, a higher-level of structure, and the possibility of running the modeling and simulation system with an optional number of the modules present.

5. Development of the Libraries

5.1 Development Environment

The `BioChem` and `Metabolic` libraries have been developed using the MathModelica [45, 46] environment that consist of the Dymola kernel [47], the Mathematica notebook environment [48], and the graphical Model Editor.

In the MathModelica environment the Modelica code along with the documentation for each library is integrated in Mathematica notebooks [48]. This does not only make it easier for non-computer science users to navigate in the code, it also facilitates for these users to write their own Modelica classes. The Model Editor is a graphical drag-and-drop interface currently based on Microsoft Visio [49]. The user creates models in the graphical environment by dragging and dropping com-

ponents from existing model libraries onto the diagram area and then connecting them in a suitable manner. Models can also be created in the Mathematica notebook textual environment, but the models must then first be transferred to the Model Editor in order to get a graphical view of the model.

Once a model has been created it can either be transferred to a notebook for further processing and documentation or simulated in the simulation environment provided by MathModelica. The Dymola kernel handles the simulations by receiving, compiling, and executing the model. The result from the simulation can then be presented with different types of diagram. The parameters and the initial values of the model can also be altered in-between simulations.

5.2 Basic Idea of Library Design

The design idea behind the `BioChem` library (Figure 5) is to create a general purpose Modelica library for modeling and simulation of biological and biochemical systems. The `BioChem` library is not intended, nor designed to be used directly for creating models and running simulations, but rather to provide some common basic behaviors, attributes, and environmental properties to be used in special-purpose biological and biochemical libraries. With the basic features provided in `BioChem` it is easy to create new special-purpose libraries without extensive addition of new code.

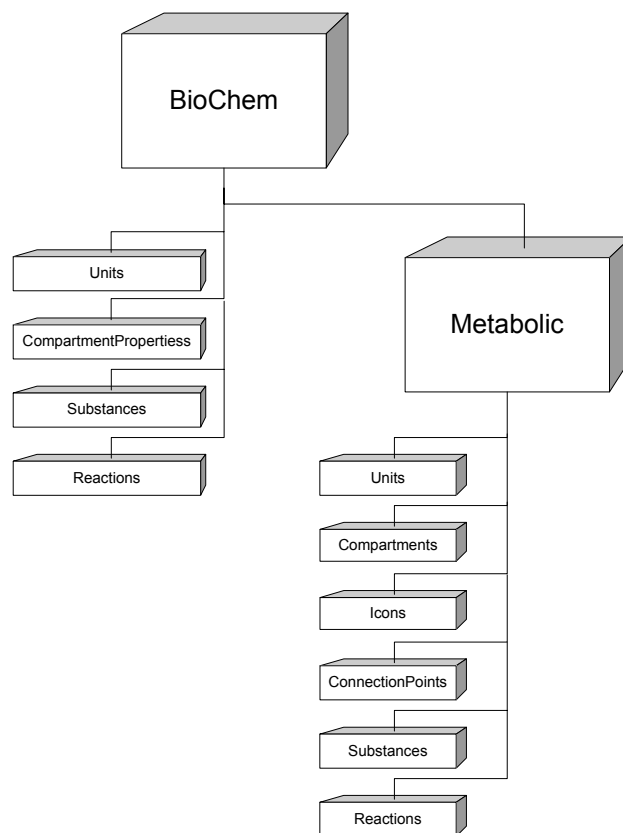


Figure 5. Simplified view of the structure of the `BioChem` library with the special-purpose library `Metabolic`.

So far the Modelica library `Metabolic` (Figure 5) is the only library to use the features provided by `BioChem`. The design idea behind `Metabolic` is to create a special-purpose Modelica library for modeling, simulation, and visualization of metabolic pathways, i.e., modeling, simulation, and visualization of the metabolic level in cells. The classes implemented in `Metabolic` describe substances and reactions that can take place in-between these substances in a diverse number of metabolic pathways.

5.3 The BioChem library

Most substances and reactions, respectively, have some common basic features. For instance, all substances must have a concentration and all reactions must have at least one substrate and one product. The design objective behind the `BioChem` library is to collect these basic features of substances and reactions along with units, compartment properties, and other attributes that are commonly used in these kinds of systems in a general-purpose biological and biochemical Modelica library.

```

package BioChem
  package Units
    "Units used in sub-packages of BioChem"
  end Units;
  package CompartmentProperties
    "Properties for compartments used in sub-libraries"
  end CompartmentProperties;
  package Substances
    "Basic components for reaction nodes in the package"
  end Substances;
  package Reactions
    "Basic components for reaction in the package "
  end Reactions;
end BioChem;

```

Figure 6. Structure of the `BioChem` library.

In order to avoid recreating model code for the basic features of substances and reactions for each new Modelica library for biological or biochemical systems these features can instead be collected in one library. Along with substances and reactions it is also practical to define a default environmental container in which the substances are contained and where the reactions can occur. From the visualization's point of view it is also practical to define some default interfaces and icons which later might be replaced in each sub-library. Not only the icons and interfaces are designed to be easily changed and/or replaced, most of the classes in `BioChem` are designed in such a way that they easily can be extended, and some parameters can also be replaced. Due to the design of `BioChem` some restrictions on the types of systems that `BioChem` can be used for arise. The systems that the classes in `BioChem` can be used for are only those biological and biochemical systems that contain mathematical definable chemical reactions. Only for those systems fully functional models that can be used for simulation can be specified. The structure of the package in Modelica code is shown in Figure 6.

5.4 The BioChem Sub-Packages

A simple reaction consists of some substrate(s) that are transformed into some product(s) in a known environment. The basics of these three components, i.e., reactions, substances, and environments are described in `BioChem` along with the some types that are needed to specify the math behind the reaction.

The basics for substances and reactions in biological and biochemical systems are provided in the `BioChem.Substances` package and the `BioChem.Reactions` package respectively. In the `BioChem.CompartmentProperties` package the basics of restricted screened-off containers where the reactions can take place are provided. For all reactions that are placed in the same container all the basic physical properties, e.g. volume and temperature, are the same.

A number of physical types that are needed in order to be able to declare most parameters and variables in the `BioChem` package are collected in the `BioChem.Units` package. Some of the types are SI types and are hence imported from the `Modelica.SIunits` library in order to avoid long name paths. Other types are non-SI types and thus need to be fully declared.

5.5 The Metabolic library

Most classes in the `Metabolic` library extend one or more classes in the `BioChem` library. Generally the partial models specified in `BioChem` are extended, and with only a few additions, turned into fully functional models. As mentioned earlier, many of the reactions that occur in metabolic pathways are more or less the same in all cells no matter what species is considered. This is utilized in `Metabolic` to create a collection of partial models of different metabolic pathways that through small changes and/or additions are turned into fully functional species-specific metabolic pathways. The structure of the `Metabolic` package in Modelica code is shown in Figure 7.

5.6 Metabolic Sub-Packages

In order to be able to run a simulation of a model all substances, reactions, and other constructs in the model must be placed within a compartment model. The `Metabolic.Compartments` package contains models for some of the different types of containers that can be found in cells when dealing with modeling and simulation of metabolic pathways. The partial compartment models in `BioChem.CompartmentProperties` are extended in order to obtain the basic properties of a compartment. Reactions and substances that require different properties than the ones provided by the main-compartment can be placed in new compartments within or adjacent to the main-compartment.

```

within BioChem;
package Metabolic
  "Package for metabolic cellular reactions"
  package Units
    "Units used in the package"
  end Units;
  package Compartments
    "Different types of compartments used in the package"
  end Compartments;
  package Icons
    "Icons used in the package"
  end Icons;
  package ConnectionPoints
    "Connector interfaces used in sub-libraries"
  end ConnectionPoints;
  package Substances
    "Reaction nodes"
  end Substances;
  package Reactions
    "Reaction edges"
    package Kinetics
      "Kinetic reactions"
      package UniUni
        "A->B kinetic reactions"
      end UniUni;
      package UniBi
        "A->B+C kinetic reactions"
      end UniBi;
      package UniTri
        "A->B+C+D kinetic reactions"
      end UniTri;
      package BiUni
        "A+B->C kinetic reactions"
      end BiUni;
      package BiBi
        "A+B->C+D kinetic reactions"
      end BiBi;
      package BiTri
        "A+B->C+D+E kinetic reactions"
      end BiTri;
      package TriUni
        "A+B+C->D kinetic reactions"
      end TriUni;
      package TriBi
        "A+B+C->D+E kinetic reactions"
      end TriBi;
      package TriTri
        "A+B+C->D+E+F kinetic reactions"
      end TriTri;
    end Kinetics;
  end Reactions;
  package SBML
    "Reactions pre-defined in SBML"
    package MichaelisMenten
      "Michaelis-Menten kinetics reactions"
    end MichaelisMenten;
    package Hill
      "Hill kinetics reactions"
    end Hill;
    package Activation
      "Activation kinetics reactions"
    end Activation;
    package Inhibition
      "Inhibition kinetics reactions"
    end Inhibition;
    package Modifier
      "Modifier kinetics reactions"
    end Modifier;
    package Misc
      "Miscellaneous SBML-defined reactions"
    end Misc;
  end SBML;
end Reactions;
end Metabolic;

```

Figure 7. Structure of the `Metabolic` library.

The package `Metabolic.Substances` contains different types of nodes needed for representing a substance in a metabolic pathway. The substance models are specified by extending the partial models of substance nodes in `BioChem.Substances` and adding some additional attributes and equations. Thus both normal substance nodes and nodes with different types of restrictions, e.g. on the concentration of the substance, can be specified.

`Metabolic.Reactions` contains a collection of models for different types of reactions that can take place in metabolic pathway systems. The reactions are built up in two steps. First, different partial reaction types are specified in `Metabolic.Reactions.ReactionTypes`. Extending these basic reaction types and then adding equations for the relation between the reaction rate and the participating substances, i.e., substrates, products, and interacting enzymes, gives the different reaction models that can be used for modeling

and simulation. In addition the predefined reaction types in SBML are also included in `Metabolic.Reactions` in order to facilitate the translation of SBML-models into Modelica, and vice versa. The translation of models is performed with a two-way Modelica-SBML parser [50].

The package `Metabolic.Icons` contains icons used in the drag-and-drop interface of the Model Editor in MathModelica. Since the substances only come in a few flavors there is one icon, i.e., a sphere, for each type of node. The reactions on the other hand come in many different variations. Instead of creating one arrow icon for each type of reaction the final graphical interface for a reaction is built out of several partial icons. Enzymes that affect reactions are represented by a small arrow and an enzyme sign. The sign represent sthe type of effect that the enzyme have on the reaction, i.e., inhibition, activation, or a combination of both, and are indicated with a $-$, $+$, and **M** respectively.

In order to connect the graphical interface to the underlying models, connecting points are needed. The `Metabolic.ConnectionPoints` package contains connectors and several partial connector-models that relate to the graphical interface of at least one icon in `Metabolic.Icons` (Not more than one icon at a time though.). For the reaction arrows, connectors are placed at each intended connectable end. For the enzymes regulating the reactions the connectors are placed at the enzyme signs. Finally for substances, eight connectors are placed on the rim of the shere that represents the node of substance.

6. Example: Malate Dehydrogenase

The enzyme Malate dehydrogenase catalyzes, amongst other reactions, the transformation of S-Malate into Oxaloacetate. The reaction is part of the Citrate Cycle (Figure 1) and the overall reaction is:

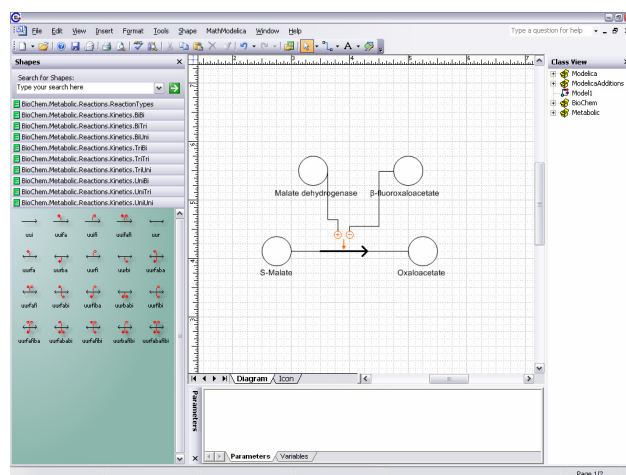


Figure 8. The Malate dehydrogenase catalyzed reaction in the MathModelica environment.

Although energetically unfavorable ($\Delta +6.7$ kcal/mol) the reaction goes forward because NADH is oxidized rapidly via the respiratory chain and oxaloacetate goes on to react with another acetyl-CoA molecule. Apart from Malate dehydrogenase, NAD⁺ also participates in the reaction as a coenzyme. The substance β -fluoroxyaloacetate might inhibit the reaction.

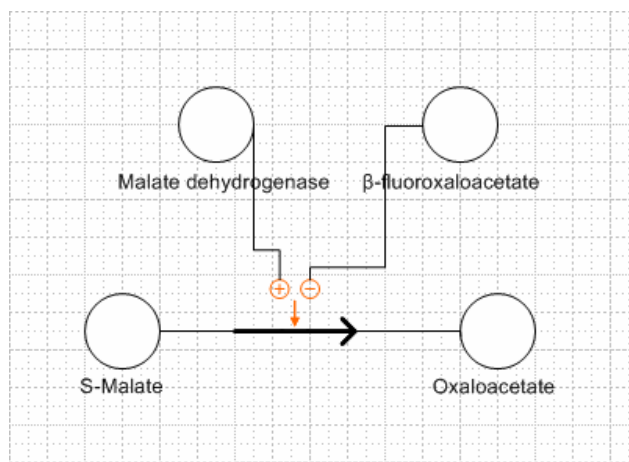


Figure 9. The actual graphical model of the Malate dehydrogenase catalyzed reaction.

The MathModelica graphical model of the Malate dehydrogenase catalyzed reaction in the MathModelica environment is shown in Figure 8. The model consists of four substances spheres connected to one reaction arrow Figure 9. NADH, NAD⁺, and H⁺ are not shown in the model since they are part of the container model. In that way the concentrations of substances like NADH/NADH⁺, ATP/ADP/AMP, and H⁺ are assured to be the same in the whole container.

The variables and parameters for the participating substances and reactions can be reviewed and changed in the pane below the graphical model (Figure 8) and a simulation of the reaction can be performed by choosing the option *Simulation* in the *MathModelica* menu (Figure 10). The result of a simulation is shown a graph.

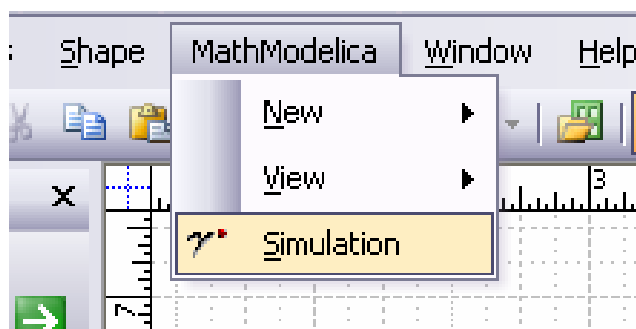


Figure 10. The MathModelica menu.

7. Related Work

7.1 Modeling Tools

The idea of using object-oriented frameworks for metabolic modeling was first proposed in the mid 1990s [51, 52]. Many of the commercially available tools for modeling, simulation, and visualization of metabolic pathways available today use an object-oriented framework in the sense that they are implemented in an object-oriented language with reuse of code at component level. This being said, very few attempts so far have made full use of object-orientation throughout the whole construction of a modeling and simulation environment for metabolic pathways, i.e., applied in the design of the framework, in the implementation, and in the creation of model templates.

During the last couple of years many commercial tools for quantitative simulation of metabolic pathways based on numerical integration of rate equations have been released. With software like GEPASI [53, 54], KINSIM [55], MIST [56], METAMODEL [57], SCAMP [58], and PathwayLab [59], modeling and simulation of metabolic pathways can also be done in a graphical environment. Some tools contain a collection of metabolic pathways, while others rely on import of models from different databases. However, only a few of the available commercial tools are designed for reconstructions at several levels within the cell system. The two-leveled MetaCore [60] is one such system. Linking the gaps between the various levels of the cell's process hierarchy is an extremely challenging problem that has yet to be adequately addressed. The introduction of the E-CELL [61] system is a significant step in that direction.

7.2 Object-Oriented Languages

Many of the common object-oriented languages, e.g. C++ and Java, have been used in several systems for metabolic modeling and simulation. However, there is one object-oriented language specially developed for biomedical applications, OOB-SML, the Object-Oriented Biomedical Systems Modeling Language [62]. OOB-SML [63, 64] was developed with the aim to model and simulate continuous biomedical systems in an interactive knowledge-based environment.

7.3 XML-languages

In order to be able to store and interchange entire models between different pathway tools, several XML-languages have been developed. The non biology-specific MathML [65] is commonly used along with other more biology-specific languages such as SBML [66, 67] and CellML [68].

8. Conclusions

Apart from the single-level limitation of many systems, there are several other limitations in many of the tools. Most of the tools must rely on the pathway data available in metabolic maps found in databases, with the previously mentioned limitations. One other limitation is that most of the tools mainly deal with bacterial metabolism, which is far less complicated than eukaryotic, and especially mammalian/human, metabolism. The graphical user interface is also an obstacle. Most graphical editors have their own layout, which can lead to confusion when switching between different tools.

Different XML-languages have been proposed to make it possible to interchange models in-between different modeling and simulation tools. One limitation of the se XML-models is that the visual appearances of reactions and substances are not included in the models.

During the work with the BioChem and the Metabolic libraries some limitations of the Modelica language has forced us to re-design the libraries' structure at several points. The original BioChem library [10, 11] was at a point divided into two libraries, i.e., BioChem and the Metabolic, which made a significant improvement of the library design and hence the underlying library structure. The design that is presented in this paper is currently being extensively tested and has not shown any major shortcomings this far.

9. Future Work

The BioChem package will probably have few additions of classes and models in the future, while there will surely be more packages added. As mentioned before, the main purpose of BioChem is to serve as a general-purpose package for biological and biochemical Modelica-packages.

The construction of a library with metabolic pathway templates will also continue. The idea is that these model templates can easily be extended and adapted to concrete models. The concrete models can then be used in standalone and connected simulations. For all of the above tasks, the data contained in the different resources mentioned in Section 2.3 will be useful.

10. Acknowledgements

The authors would like to thank MathCore Engineering AB for supplying the MathModelica tool and Andreas Idebrant for providing essential software support. Morgan Ericsson, Växjö universitet, has provided valuable feedback on the text. Emma Larsson Nilsson was in part funded by the Swedish National Graduate School in Computer Science (CUGS).

References

1. Nicholson, J.K., et al., *Metabonomics: a platform for studying drug toxicity and gene function*. Nat. Rev. Drug Discov., 2002. 1(2): p. 153-161.
2. Klopman, G., M. Dimayuga, and J. Talafous, *META. 1. A program for the evaluation of metabolic transformation of chemicals*. J Chem Inf Comput Sci, 1994. 34(6): p. 1320-5.
3. Talafous, J., et al., *META. 2. A dictionary model of mammalian xenobiotic metabolism*. J Chem Inf Comput Sci, 1994. 34(6): p. 1326-33.
4. Darvas, F. and G. Dormán, *High-throughput ADMETox estimation : in vitro and in silico approaches*. 2002, Westborough, MA: Eaton Pub. ; BioTechniques Press. x, 89.
5. Greene, N., et al., *Knowledge-based expert systems for toxicity and metabolism prediction: DEREK, StAR and METEOR*. SAR QSAR Environ Res, 1999. 10(2-3): p. 299-314.
6. Elmqvist, H., S.E. Mattsson, and M. Otter. *Modelica - A Language for Physical System Modeling, Visualization and Interaction*. in *1999 IEEE Symposium on Computer-Aided Control System Design*. 1999. Hawaii, USA.
7. Fritzson, P. and P. Bunus. *Modelica - A General Object-Oriented Language for Continuous and Discrete-Event System Modeling and Simulation*. in *The 35th Annual Simulation Symposium*. 2002. San Diego, California, USA: IEEE.
8. Fritzson, P., *Principles of Object-Oriented Modeling and Simulation with Modelica*. 2003: IEEE Press and Wiley.
9. ModelicaAssociation, *Modelica webpage*, www.modelica.org. 2005.
10. Larsson Nilsson, E. *Simulation of Biological Pathways using Modelica*. in *The Huntsville Simulation Conference 2003*. 2003. Huntsville (AL), USA.
11. Larsson Nilsson, E. and P. Fritzson. *BioChem - A Biological and Chemical Library for Modelica*. in *The 3rd International Modelica Conference*. 2003. Linköping, Sweden.
12. Katzung, B.G., *Basic & clinical pharmacology*. 9. ed. Lange medical books,. 2004, New York: Lange Medical Books/Mcgraw-Hill. xiv, 1202.
13. Ansoerge, W., H. Voss, and J. Zimmermann, *DNA sequencing strategies : automated and advanced approaches*. 1997, New York and Heidelberg: Wiley/Spektrum. xiv, 202 p.
14. Martin, W.J., et al., *Automation of DNA Sequencing: A System to Perform the Sanger Dideoxysequencing Reactions*. 1985. 3(10): p. 911-915.
15. Sanger, F., et al., *Nucleotide sequence of bacteriophage [λ] DNA*. Journal of Molecular Biology, 1982. 162(4): p. 729-773.
16. van Hal, N.L.W., et al., *The application of DNA microarrays in gene expression analysis*. Journal of Biotechnology, 2000. 78(3): p. 271-280.
17. Watson, A., et al., *Technology for microarray analysis of gene expression*. Current Opinion in Biotechnology, 1998. 9(6): p. 609-614.
18. Laurell, T., J. Nilsson, and G. Marko-Varga, *Proteomics-protein profiling technology: the trend towards a microfabricated toolbox concept*. TrAC Trends in Analytical Chemistry, 2001. 20(5): p. 225-231.
19. McKusick, V.A., *Genomics: structural and functional studies of genomes*. Genomics, 1997. 45(2): p. 244-9.
20. Garlow, S.J., *And now, transcriptomics*. Neuron, 2002. 34(3): p. 327-8.
21. Blackstock, W.P. and M.P. Weir, *Proteomics: quantitative and physical mapping of cellular proteins*. Trends Biotechnol, 1999. 17(3): p. 121-7.
22. Greenberg, D.M., H.J. Vogel, and L.E. Hokin, *Metabolic pathways*. 3d ed. 1967, New York: Academic Press. v.

23. Dagley, S. and D.E. Nicholson, *An introduction to metabolic pathways*. 1970, Oxford,: Blackwell Scientific Publications. xi, 343 p.
24. Michal, G., *Biochemical pathways : an atlas of biochemistry and molecular biology*. 1999, New York Heidelberg: Wiley ; Spektrum. xi, 277 p.
25. Barredo, J.-L., *Microbial enzymes and biotransformations*. Methods in biotechnology. 2005, Totowa, N.J.: Humana Press. xi, 319 p.
26. Jeanteur, P., *Molecular and cellular enzymology*. Progress in molecular and subcellular biology ; 13. 1994, Berlin ; New York: Springer-Verlag. ix, 150 p.
27. Kanehisa, M., *The KEGG database*. Novartis Found Symp, 2002. 247: p. 91-101; discussion 101-3, 119-28, 244-52.
28. BioCarta, *Proteomic Pathway Project*, <http://www.biocarta.com>. 2004.
29. Schomburg, I., et al., *BRENDA, the enzyme database: updates and major new developments*. Nucleic Acids Res, 2004. 32 Database issue: p. D431-3.
30. Selkov, E., et al., *The metabolic pathway collection from EMP: the enzymes and metabolic pathways database*. Nucleic Acids Res, 1996. 24(1): p. 26-8.
31. Selkov, E., Jr., et al., *MPW: the Metabolic Pathways Database*. Nucleic Acids Res, 1998. 26(1): p. 43-5.
32. Karp, P.D., et al., *The EcoCyc and MetaCyc databases*. Nucleic Acids Res, 2000. 28(1): p. 56-9.
33. Fujibuchi, W., et al., *DBGET/LinkDB: an integrated database retrieval system*. Pac Symp Biocomput, 1998: p. 683-94.
34. Bugrim, A., T. Nikolskaya, and Y. Nikolsky, *Early prediction of drug metabolism and toxicity: systems biology approach and modeling*. Drug Discov Today, 2004. 9(3): p. 127-35.
35. Bock, G. and J. Goode, *'In silico' simulation of biological processes*. Novartis Foundation symposium ; 247. 2002, New York: John Wiley. viii, 262 p.
36. Kitano, H., *Foundations of systems biology*. 2001, Cambridge, Mass.: MIT Press. 297 p.
37. Gaasterland, T. and E. Selkov, *Reconstruction of metabolic networks using incomplete information*. Proc Int Conf Intell Syst Mol Biol, 1995. 3: p. 127-35.
38. Overbeek, R., et al., *WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction*. Nucleic Acids Res, 2000. 28(1): p. 123-5.
39. Overbeek, R., et al., *The ERGO genome analysis and discovery system*. Nucleic Acids Res, 2003. 31(1): p. 164-71.
40. Covert, M.W., et al., *Metabolic modeling of microbial strains in silico*. Trends Biochem Sci, 2001. 26(3): p. 179-86.
41. Rives, A.W. and T. Galitski, *Modular organization of cellular networks*. Proc Natl Acad Sci U S A, 2003. 100(3): p. 1128-33.
42. Snel, B., P. Bork, and M.A. Huynen, *The identification of functional modules from the genomic association of genes*. Proc Natl Acad Sci U S A, 2002. 99(9): p. 5890-5.
43. Spirin, V. and L.A. Mirny, *Protein complexes and functional modules in molecular networks*. Proc Natl Acad Sci U S A, 2003. 100(21): p. 12123-8.
44. Bunus, P., *Debugging techniques for equation-based languages*. 2004, Linköping: Univ. xii, 243.
45. Fritzson, P., J. Gunnarsson, and M. Jirstrand. *MathModelica - An Extensible Modeling and Simulation Environment with Integrated Graphics and Literate Programming*. in *The 2nd International Modelica Conference*. 2002. Oberpfaffenhofen, Germany.
46. MathCoreAB, *MathModelica website*, www.mathcore.com.
47. DynasimAB, *Dymola website*, www.dynasim.se/dymola.htm.
48. Wolfram, S., *The Mathematica Book*. 2003: Wolfram Media.
49. Microsoft, *Visio website*, <http://office.microsoft.com/en-us/FX010857981033.aspx>.
50. Vollmar, D., *A two-way SBML translator and graphical icons for BioChem*, in *Department of Computer and Information Science*. 2004, Linköpings universitet: Linköping.
51. Breuel, G. and E.D. Gilles. *Towards an Object-Oriented Framework for the Modeling of Integrated Metabolic Processes*. in *German Conference on Bioinformatics 1996*. 1996.
52. Breuel, G., A. Kremling, and E.D. Gilles. *An object-oriented approach to the modeling of bacterial metabolism*. in *SAMS*. 1995.
53. Mendes, P., *GEPASI: a software package for modeling the dynamics, steady states and control of biochemical and other systems*. Comput. Applic. Biosci., 1993. 9: p. 563-571.
54. Mendes, P., *Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3*. Trends Biochem. Sci., 1997. 22: p. 361-363.
55. Barshop, B.A., R.F. Wrenn, and C. Frieden, *Analysis of numerical methods for computer simulation of kinetic processes: development of KINSIM-a flexible, portable system*. Anal. Biochem., 1983. 130: p. 134-145.
56. Ehld, M. and G. Zacchi, *MIST: a user-friendly metabolic simulator*. Comput. Applic. Biosci., 1995. 11: p. 201-207.
57. Cornish-Bowden, A. and J.H. Hofmeyr, *MetaModel: a program for modeling and control analysis of metabolic pathways on the IBM PC and compatibles*. Comput. Applic. Biosci., 1991. 7: p. 89-93.
58. Sauro, H.M., *SCAMP: a general-purpose simulator and metabolic control analysis program*. Comput. Applic. Biosci., 1993. 9: p. 441-450.
59. InneticsAB, *PathwayLab webpage*, <http://innetics.com/>. 2005.
60. GeneGo, *MetaCore*, <http://www.genego.com/>. 2004.
61. Tomita, M., et al., *E-CELL: software environment for whole-cell simulation*. Bioinformatics, 1999: p. 72-84.
62. Hakman, M., *Methods and tools for object-oriented modelling and knowledge-based simulation of complex biomedical systems*. Comprehensive summaries of Uppsala dissertations from the Faculty of Medicine, 913. 2000, Uppsala: Acta Universitatis Upsaliensis : Univ.-bibl. distributör. 54 ;.
63. Hakman, M. and T. Groth, *Object-oriented biomedical system modelling -- the language*. Computer Methods and Programs in Biomedicine, 1999. 60(3): p. 153-181.
64. Hakman, M. and T. Groth, *Object-oriented biomedical system modeling--The Rationale*. Computer Methods and Programs in Biomedicine, 1999. 59(1): p. 1-17.
65. Ausbrooks, R., et al., *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*. 2003.
66. Finney, A. and M. Hucka, *Systems biology markup language: Level 2 and beyond*. Biochem Soc Trans, 2003. 31(Pt 6): p. 1472-3.
67. Hucka, M., et al., *The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models*. Bioinformatics, 2003. 19(4): p. 524-31.
68. Lloyd, C.M., M.D. Halstead, and P.F. Nielsen, *CellML: its future, present and past*. Prog Biophys Mol Biol, 2004. 85(2-3): p. 433-50.