Int J STTT (2001) 3: 394–404 / Digital Object Identifier (DOI) 10.1007/s100090100058

Executable Petri net models for the analysis of metabolic pathways

Hartmann Genrich, Robert Küffner, Klaus Voss

GMD – German National Research Center for Information Technology, Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Published online: 24 August 2001 –

Abstract. Computer-assisted simulation of biochemical processes is a means to augment the knowledge about the control mechanisms of such processes in particular organisms. This knowledge can be helpful for the goal-oriented design of drugs. Normally, continuous models (differential equations) are chosen for modelling such processes. The application of discrete event systems such as Petri nets has been restricted in the past to low-level modelling and qualitative analysis. To demonstrate that Petri nets are indeed suitable for simulating metabolic pathways, the glycolysis and citric acid cycle are selected as well-understood examples of enzymatic reaction chains (metabolic pathways). The paper discusses the steps that lead from gaining necessary knowledge about the involved enzymes and substances, to establishing and tuning high-level net models, to performing a series of simulations, and finally to analysing the results. We show that the consistent application of the Petri net view to these tasks has certain advantages, and – using advanced net tools – reasonable simulation times can be achieved.

Keywords: Coloured Petri nets – Executable model – Metabolic pathway – Glycolysis – Citric acid cycle

1 Introduction

Finding promising targets for the development of new drugs very much depends on reliable knowledge about the metabolism (metabolic processes in the human organisms). Then, this knowledge can be exploited to avoid unnecessary, costly, and dangerous experiments and to conduct the remaining unavoidable experiments effectively with the goal of finding drug targets.

Part of this work has been funded by the BMBF under contract TargId (0311615).

Traditionally, the dynamics of metabolic processes is investigated by simulations on the basis of differential equations (e.g., [4, 8, 14]). This is usually done by providing a particular kinetic equation for each reaction of the pathway requiring a considerable number of kinetic constants derived from experimental data. The simulation then proceeds by executing these equations cyclically (and updating the concentrations of the involved substances) in very small timesteps. A prominent example is E-CELL ([16]), a particular software environment for whole-cell simulation. E-CELL offers, among other things, graphical user interfaces to observe the cell's state and manipulate it interactively.

An alternative is the simulation by discrete event systems. Petri nets have been proposed in [13] because of their appropriate semantics (occurrence rule), the inherent precise concurrency notion, their intuitive graphical representation, and their capabilities for (mathematical) analysis. It demonstrates the significance of net abstraction, boundedness, S-invariants, T-invariants, and liveness to draw important "preliminary conclusions about the metabolic pathway". However, the approach in [12] aims at a purely qualitative analysis of biochemical pathways and does not allow simulation of quantitative kinetic effects. This is motivated by the observation that "modelling a complex biochemical system involves data that are incomplete, uncertain or unreliable". Fortunately, the availability of reliable data has improved in recent years although it still remains a serious problem.

To our knowledge, attempts to simulate metabolic pathways by Petri nets so far have been restricted to relatively small reaction chains modelled as low-level nets which are constructed more or less by hand. In [11], so-called "hybrid Petri nets" are chosen to model and simulate a gene regulatory network. This low-level net class contains discrete as well as continuous nodes, where continuous places are marked with real numbers (instead of unit tokens). Section 2 shows, however, that time inter-

vals and real numbers can be handled easily in timed high-level nets and, hence, there is no obvious need to leave the standard Petri net classes with their advanced theory and tools.

This paper describes a first step towards the (automatic) creation and implementation of high-level Petri net models (see [6]) which allow the simulation and quantitative study of networks of metabolic processes, and demonstrates this by use of a well-known example. In contrast to most systems based on differential equations, our approach: (a) aims at automating as many tasks as possible, notably the pathway construction, data collection, and model creation; (b) relies on standard non-specialized software tools and modest hardware equipment, which has to be paid for by less efficient simulations; (c) represents the models as intuitive graphical Petri net graphs which can be understood easily by experts in biochemistry, but does not contain a graphical interface to observe or influence the simulation progress (like E-CELL); and (d) admits not only a simulation and quantitative study of the net model but also the application of qualitative analysis methods from Petri net theory to the same kind of model[1] (work in progress).
Hence, it constitutes an alternative that hopefully will stimulate research and development in the field of metabolic systems.

The paper is organized as follows. Section 2 introduces the basic notions and concepts used for representing a metabolic reaction as a Petri net, and it discusses our choice of the kinetic reaction function. Section 3 deals with the problem of systematically constructing pathways for metabolic processes and of assembling the reactions and metabolic constants for the chosen pathways from the databases. Section 4 then explains the most prominent features of the Petri net models that have been developed for investigating the well-known processes contributing to the glycolysis and – in connection with this – the citric acid cycle. Section 5 presents some typical simulation results and performance figures, followed by a short paragraph with conclusions and some suggestions for future work.

The discussed application runs on a Power Macintosh G3, using the software packages Design/CPN [2] (for modelling and simulation), Excel (for plotting), and MacPerl (for data extraction from databases).

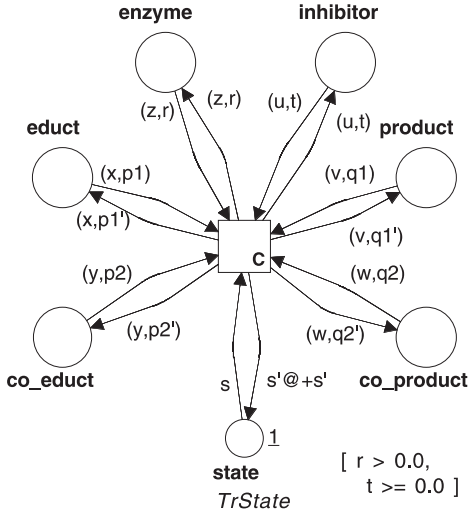
For reading this paper some knowledge of coloured Petri nets (e.g., [6]) and Design/CPN would be helpful.

2 A sample of a reaction and the kinetic function

An enzymatic reaction changes the concentrations of the involved *substrates* (the substances participating in the reaction), catalyzed by a reaction-characteristic enzyme. Some reactions may be slowed down by substances called *inhibitors*. In principle, every enzymatic reaction is reversible; however, most of them have a preferred direction. Therefore, the substrates whose concentrations are decreased in this preferred mode are called *reactants* or *educts*, those with increased concentrations are called *products*. The *speed* of the reaction is the amount of the concentration change within a time unit, $\Delta c / \Delta t$.

An enzymatic reaction can be modelled as a high-level Petri net transition in a straightforward manner. In the coloured Petri net of Fig. 1, the transition in the center models a (sample) biochemical reaction. Every substrate is represented as a place connected through an outgoing and an ingoing arc to the transition. For each of these places, the datatype of its tokens ("colour set") is chosen to be a set of pairs, each consisting of the name and the concentration of a particular substrate. (The purpose of the place *state* is explained in Sect. 4.)

[1] For a qualitative analysis, only the existence or absence of substances is important, not their concentrations. Hence, all parts dealing with the metabolic kinetics can be omitted.




```
input (z,r,x,p1,y,p2,v,q1,w,q2,u,t,s);
output (p1',p2',q1',q2',s');
action
let val (dc,dG,s_) =
   R_kin {enzyme=   (z,r),
          educts=   [(x,p1),(y,p2),(u,t)],
          products= [(v,q1),(w,q2)],
          step=     s }
   val (p1_,p2_,q1_,q2_) = (p1-dc,p2-dc,q1+dc,q2+dc)
in
map UpdateConc [(x,p1_), (y,p2_), (v,q1_), (w,q2_),
                (z^"_dv",dc/CurrSpeed()/real(s)),
                (z^"_dG",dG)];
(p1_,p2_,q1_,q2_,s_)
end;
```

Fig. 1. The model of an enzymatic reaction

Take for example the enzyme *aldolase* with the EC-number[2] 4.1.2.13. It catalyses the reaction[3]

F16P = GA3P + DacP.

Its detailed net model would be that of Fig. 1, with the names

F16P as educt,
GA3P as product, DacP as co_product,
E_4′1′2′13 as enzyme,

and with the co_educt and inhibitor places omitted. The educt place, in this example, is marked by a pair ("F16P", [F16P]), where [F16P] denotes the concentration of F16P at the current state of the net system.

We now return to the general net of Fig. 1. The result of an occurrence of the reaction transition is specified by the functions in its "code region", attached to the transition and executed during simulation. Two functions are called: (a) the kinetic function *R_kin* (discussed in the next paragraph) determines the reaction speed dc ($\equiv \Delta c$), the current free enthalpy dG ($\equiv \Delta G$), and the step width $s\_$; and (b) the function *UpdateConc* whose argument is a list of pairs (n, v) each one denoting a name n and a value v to be written into a plot file. Apart from the names and new concentrations of substrates, the actual reaction speed dc (reduced to step width 1) and dG is also recorded in that file. After the end of the simulation, the plot file serves as input to Excel to produce a plot diagram of (a selection of) these values as a function of time.

For the calculation of the reaction speed we use a general (reaction-independent) kinetic function *R_kin*. It is a reversible *Michaelis-Menten* equation (see e.g., [15, 17]) augmented by an additional term for the free reaction energy, thus combining kinetic and thermodynamic information. The parameters of the kinetic function *R_kin* are the current concentrations of all involved substances. It (essentially) computes the reaction speed, i.e., the decrease/increase Δc (in time unit 1) of the concentrations of the educts/products. The reaction is composed of a forward portion and a backward portion, marked by respective upper arrows in *R_kin*. The concentrations of the enzyme – being a catalyst – and of the inhibitor(s) remain unchanged. *R_kin* only needs a few chemical and enzyme specific constants for its computation. Before starting any simulation, the constants of all relevant enzymes are extracted from the database [1] (see Sect. 3) and collected in a data structure called *enzyme catalog*. Given a reaction (without inhibitors) catalyzed by enzyme e, let S/P denote the set of its educts/products. The concentration of a substance x shall be denoted by $[x]$. Then, a slightly simplified version of *R_kin* reads:

$$\Delta c = (\vec{K} - \overleftarrow{K}) \cdot k_{cat} \cdot [e] \cdot (\vec{K} \cdot \vec{Q} + \overleftarrow{K} \cdot \overleftarrow{Q})$$

with

$$\Delta G = \Delta G0 + R \cdot T \cdot ln \frac{\Pi_{p \in P}\,[p]}{\Pi_{s \in S}\,[s]}$$

$$K = e^{-\Delta G/R \cdot T}\,,\ \vec{K} = \frac{1}{1+K}\,,\ \overleftarrow{K} = \frac{K}{1+K}$$

$$\vec{Q} = Min_{s \in S}\,\frac{[s]}{[s] + Km_s}\,,\ \overleftarrow{Q} = Min_{p \in P}\,\frac{[p]}{[p] + Km_p}$$

where R and T are Nature's constants, k_{cat} and Km_x are enzyme constants contained in the enzyme catalog, and $\Delta G0$ is characteristic of the substances involved.

The feasibility of *R_kin* can be checked by applying it to four particular situations.

1 In the irreversible case it degenerates to the well-known *Michaelis-Menten* equation
$\Delta c = k_{cat} \cdot [e] \cdot [s] \,/\, ([s] + Km_s)$.
2 Always, $\Delta c \le k_{cat} \cdot [e]$ holds.
3 In case $\vec{Q} = \overleftarrow{Q} = 1$, Δc only depends on $\vec{K} - \overleftarrow{K}$.
4 $\Delta G = 0 \Rightarrow \Delta c = 0$,
$\Delta G < 0 \Rightarrow \Delta c > 0$, $\Delta G > 0 \Rightarrow \Delta c < 0$. [4]

As an example, the reaction equation *R_kin* for the enzyme *aldolase* reads:

$$\Delta c = \frac{1-K}{(1+K)^2} \cdot k_{cat} \cdot \left(\frac{[\mathrm{F16P}]}{[\mathrm{F16P}] + Km_{\mathrm{F16P}}} + K \cdot Min\left(\frac{[\mathrm{GA3P}]}{[\mathrm{GA3P}] + Km_{\mathrm{GA3P}}}, \frac{[\mathrm{DacP}]}{[\mathrm{DacP}] + Km_{\mathrm{DacP}}} \right) \right)$$

where $\Delta G = \Delta G0 + 2.424 \cdot \ln \dfrac{[\mathrm{F16P}]}{[\mathrm{GA3P}] \cdot [\mathrm{DacP}]}$,
$K = \mathrm{e}^{-\Delta G/2.424}$, $\Delta G0 = -13.0$, $k_{cat} = 40.0$,
$Km_{\mathrm{F16P}} = 50.0$, $Km_{\mathrm{GA3P}} = 500.0$, $Km_{\mathrm{DacP}} = 500.0$

Differential equations (DEs) normally are processed in discrete time steps, computing iteratively the "rate expressions", i.e., the reaction speeds. In [8], the rate expression for the *aldolase* reaction reads:

$$\Delta c = (\,\mathrm{N}_1 - \mathrm{N}_2\,[\mathrm{GA3P}]\,[\mathrm{DacP}]\,)\,\mathrm{ET} \,/\, (\,\mathrm{D}_+\mathrm{D}_2\,[\mathrm{F16P}] + \mathrm{D}_3\,[\mathrm{GA3P}] + \mathrm{D}_4\,[\mathrm{DacP}] + \mathrm{D}_5\,[\mathrm{F16P}]\,[\mathrm{GA3P}] + \mathrm{D}_6\,[\mathrm{GA3P}]\,[\mathrm{DacP}]\,)$$

where $\mathrm{ET} = 0.37 \cdot 10^{-3}$, $K_{eq} = 0.081$,
$K_1 = 3.85 \cdot 10^7$, $K_2 = 8.388 \cdot 10^5$, $K_3 = 6.84 \cdot 10^6$,
$K_4 = 4.032 \cdot 10^7$, $K_5 = 2.52 \cdot 10^5$, $K_6 = 23.076 \cdot 10^6$,
$N_1 = K_1 K_3 K_5$, $N_2 = N_1 / K_{eq}$,
$D_1 = K_5(K_3 + K_5)$, $D_2 = K_1(K_3 + K_5)$, $D_3 = K_2 K_4$,
$D_4 = K_6(K_3 + K_2)$, $D_5 = K_1 K_4$, $D_6 = K_4 K_6$

[2] The EC-numbers reflect the official classification of the enzymes. The first three numerals in the EC-number hierarchically define the type of the enzymatic function, the fourth numeral increments over different enzymes which catalyze the same function. For syntactical reasons, a place for the enzyme with EC-number i.j.k.l is named E_i′j′k′l in a Design/CPN net.

[3] Enzymes and substrates are written in a shorthand notation. Their full names are listed in the glossary of Appendix A.

[4] Hence, $\Delta G0$, the change of the free enthalpy under standard conditions, determines that constellation of concentrations at which the (reversible) reaction changes its direction, i.e., the sign of Δc.

For a comparison of *R_kin* with DEs, the following aspects are essential:
– For each reaction, a particular DE must be established, whereas *R_kin* is applied uniformly to all reactions.
– According to remark 4 above and footnote 4, in case of applying *R_kin*, the switch of the reaction direction is in accordance with the free enthalpy value $\Delta G0$. Inspecting some of the DEs showed, however, that they do not meet this requirement.
– In general, *R_kin* needs the values of the *Michaelis-Menten* constants Km_x of the substrates x, the maximum reaction velocity $k_{cat} \cdot [e]$ of the enzyme e, and $\Delta G0$. Hence, the number of parameters is *linear* with respect to the number of involved metabolites.
– Usually, DEs require more coefficients than *R_kin*. According to the general kinetic equation in [8], the number of parameters is *quadratic* with respect to the number of metabolites. In addition, in contrast to *R_kin*, these values cannot be found in databases of the internet. "Although some kinetic values can be derived from information available in existing databases, many are unknown. We have assigned values for these parameters by estimations based on available information" [16]. To give an example, many of the equations in [8, 16] require 12–16 parameters, where the corresponding *R_kin* needs 5 or 6.

3 Metabolic pathways

As main sources of information on metabolic pathways the Internet-accessible databases [1], [3], and [7] are used. Entries of these databases describe one enzymatic function each and are indexed via their EC-number. The chemical reaction equations contained in the database entries can be used for two purposes: first, to define transitions of a Petri net, and second, to define a network of enzyme–substrate–enzyme edges via matching and identifying the educts and products of reactions.

The key problem here is the unification of the substrate names, due to different naming conventions. By manually augmenting existent alias lists, detection of typos, etc., the contents of the diverse databases can be compared and compiled into a unified Petri net. In the actual state of the databases the unified Petri net contains about 3 200 EC-entries, 11 300 reactions (transitions) and 12 300 substrates (places) leading to 164 000 enzyme–substrate–enzyme edges.

For the purpose of simulating metabolic pathways derived from the databases, the kinetic enzyme parameters listed at the end of Sect. 2 are needed. Fortunately, BRENDA [1] covers these parameters for a wide range of enzymes, substrates, and organisms, but is far from being complete. With the exception of $\Delta G0$, whose values were taken from [15], all parameters of *R_kin* are contained in BRENDA for the best examined pathways like the glycolysis and citric acid cycle. In case the dynamics are computed using differential equations, the situation is much worse. For one single reaction, sometimes up to 16 values have to be determined to fit the biochemical experiments, see [8, 16].

The next step comprises developing an appropriate language to access the various entries of a database from within the CPN model. This language can then be applied to find the relevant reactions and to compute the necessary metabolic constants. Having checked these data for completeness, they can be inserted into the *enzyme catalogue* which in turn will be inspected by the kinetic function *R_kin* during a simulation to compute the actual reaction speed (see Sect. 2).

A (metabolic) path is a coherent set of enzymatic reactions. The reactions are interconnected via the substrates (educts and products) they act upon. In contrast to naive graphs, Petri nets allow for representing and distinguishing different structural constellations in biochemical networks which is a prerequisite for the systematic construction of pathways in such nets. In particular, the difference between *branching reactions* (one reaction producing more than one product) and *conflicting reactions* (several reactions competing for the same educt) is of substantial importance (see below).

In the following, three rules are defined which shall serve to find sensible and manageable (in size and speed) pathways among the millions of possibilities. A justification for these rules is provided in [10].

First rule. Inspired by the occurrence rule of Petri nets, only those paths – henceforth called *pathways* – shall be considered which are *closed* in the sense that they take care of the availability of all educts and the consumption of all intermediate products. The result of the first rule is that there are no loose ends in such a pathway. An exception from this rule are small molecules like H_2O, NADH, ADP, and CO_2 found in sufficiently large amounts in all organisms, called *ubiquitous* molecules.

The essential task prior to constructing a metabolic Petri net model is the sensible selection of the pathways to be modelled. To start with, the initial and final substrates, i.e., the *source* and the *sink* of the envisaged metabolic process have to be determined.

For example, the *glycolysis* comprises all metabolic processes leading from glucose as the initial to pyruvate as the final substance. An unrestricted search in the database [1] would result in about 500 000 paths (of a length of at most 9; not shown). Applying the above mentioned first rule leads to a pathway with about 80 000 paths. Clearly, the resulting number of pathways is still much too large to be handled.

Hence, a *second rule* is applied which mirrors the observation that very long paths connecting two substances usually contribute much less to the concentration changes of these substrates than short ones. This second rule is depicted in Fig. 2. It cuts off all those paths which exceed a certain length and a certain width (max ST-cut). Delimiting the length of the glycolysis pathways to 9 and their width to 1, leads to Fig. 3 which contains 170

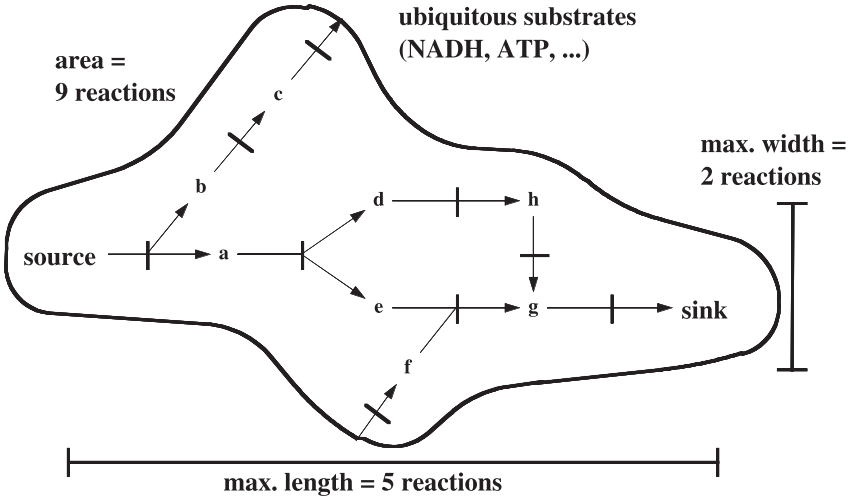


Fig. 2. Pathway reduction principles

pathways.[5] However, to include the simplified glycolysis as presented in most textbooks (Fig. 4), the width has to be at least 2, rendering 541 pathways (not shown).

As a *third rule*, which in practice should not be applied *after* but *before* or *in conjunction with* the former two rules, we restrict those pathways to contain only enzymes which exist in the organism under examination. Most of the mentioned metabolic databases include this information. For the sample organism yeast this finally results in 8 pathways for glycolysis.

4 The Petri net models

Having chosen a set of closed pathways, we are at first interested in finding states for which its execution reaches a dynamic (i.e., flow) equilibrium: given steady sources and steady sinks of a pathway, each substrate concentration must converge. To reach that goal, we model such pathways as a Petri net and then simulate that net with diverse parameters. In particular, by systematically varying the enzyme concentrations (activities), we were seeking for the most influential *key* enzymes and their concentration limits for reaching a flow equilibrium.

Our first attempt to model metabolic pathways consists in simply concatenating the particular reactions by identifying the products of any reaction with the educts of the following one(s). In case of the textbook glycolysis pathway (see, for example, [15, 17]) this leads to the intuitive model $\mathcal{A}$ depicted in Fig. 4.[6]

The model $\mathcal{A}$ is a timed high-level net diagram as expressed by Design/CPN.[7] Every reaction in Fig. 4 is a substitution transition standing for the transition of a substituting net, called "subpage", like that shown in Fig. 1. Of course, these subpages have to fit to the actual numbers of educts, products, and inhibitors. Thus we have one subpage for every possible combination of these numbers (not shown in the paper).

Initially, we assume that all reactions run at the same *step width* of 1, increasing the simulation time after each occurrence by 1. This is achieved by setting the delay expression s' of the arc pointing to the place *state* of Fig. 1. It can be noticed, however, that the speed of certain reactions are rather robust whereas others change quite dramatically in case of slight enzyme or substrate concentration deviations. The latter enzymes – called *key enzymes* – constitute the target of control mechanisms in the organism that regulate the metabolic processes according to the actual situation. This observation is taken advantage of in our model by adapting the step width (i.e., the time increment s') to the current speed of the reaction. If, for example, its current speed is very low then the delay for the next occurrence of this reaction may be increased by more than the actually used number of time units (not exceeding a maximum value of, say, 12), and if the speed is very high then the actual delay may be decreased (with a minimum value of 1). To find out an appropriate function for the speed dependent adjustment of the step width of the individual reactions is a tricky task. We checked out several strategies, but we shall not discuss

[5] Obviously, the graph is almost unreadable. It aims only at giving an impression of the connectivity of its enzymatic reactions and the involved metabolites.

[6] The full names of the enzymes and substrates (and their places) are listed in Appendix A. GL_ is not an enzyme name but denotes a place providing the integral over the amount of glucose flowing from the environment (place Gl) into the glycolysis pathway (place Glucose).

[7] Observe that, in Fig. 4, the arcs connecting the transitions with the substrates places are directed merely to indicate the preferred direction of the reaction (see Sect. 2). In the corresponding subpages they are replaced by a pair of arcs, one pointing to and one from the transition because the substrate concentrations are changed by a transition occurrence. Undirected arcs (between enzyme places and reaction transitions) indicate that the enzyme concentrations stay unchanged but are needed for the kinetic reaction function. A similar remark holds true for Fig. 7.

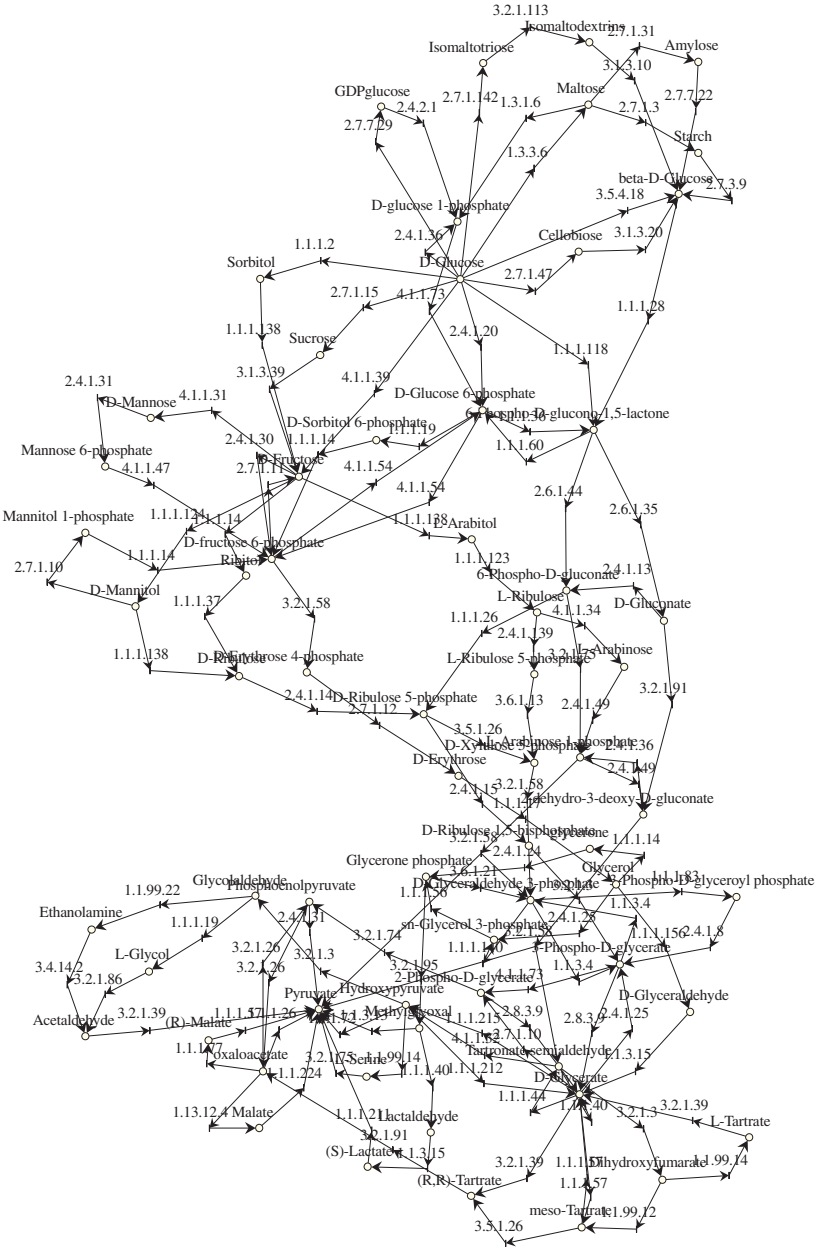


Fig. 3. Glycolysis pathways reduced to length 9 and width 1

this problem further in the paper. The effect of applying this step adjustment function leads to an acceleration of the simulation because, in every time step, only a subset of the reactions have to be executed.

After it became evident that $\mathcal{A}$ constituted a sensible model that could be generalized to arbitrary metabolic pathways, a software package[8] was developed that automatically generates such a model from the data extracted from the databases. The resulting model can be executed immediately in the simulation mode. Not much attention was paid to the graphical layout of the model: the reactions are simply arranged in lines until a page is filled, and then a new page is started. Afterwards, a final manual revision of the diagrams is advisable. The glycolysis model of Fig. 4 was constructed this way.

The simulation performance of the model $\mathcal{A}$ was not satisfactory (see the figures in Sect. 5). Therefore, merely for performance reasons, we built a modified model $\mathcal{B}$ in the following way.

In model $\mathcal{A}$, all substrate and enzyme places are always marked. This means that all reaction transitions

[8] Standard ML was used as programming language because of its flexibility, its integration in Design/CPN and its use for the annotations in CPN models. The package consists of: 1) a program system to extract from the BRENDA database all constants needed for the reactions of a given pathway and to build the enzyme catalogue (18 pages SML code); and 2) an adaption of an earlier general program system (available at our group) for saving/loading coloured nets to/from a textual form (16 pages of code).

are, in principle, always enabled. What changes are the second elements of the pairs in the substrate places, i.e., the concentrations. Even when an enzyme is transitorily de-activated, which was necessary for the experiments to be conducted, this was achieved by setting its concentration r to zero; the corresponding transition in this case is disabled due to its guard $[r > 0.0, \ldots]$ (Fig. 1).

If we focus on those transitions which are not enabled permanently, we can omit all reaction transitions or replace them all by only one transition. The current substrate concentrations can be stored in a particular *concentrations* record. This leads to the new model $\mathcal{B}$, in which there exists only one transition *setconc* comprising all reactions (Fig. 5). Model $\mathcal{B}$ is behaviourally equivalent to model $\mathcal{A}$.[9] However, the intuitive pathway diagram (Fig. 4) is no longer needed, and the automatic construction of model $\mathcal{B}$ degenerates to the generation of the enzyme catalogue and Fig. 5. This figure shall now be discussed briefly.

The colour set *TrState* consists of one list in which the necessary variables for the enzymes and the step width

[9] A formal proof would not be obvious. Its core would be to apply the equivalence transformation rules of high-level nets from [5] and then to show that, in one time step, executing the reactions along the transitions structure of net $\mathcal{A}$ (Fig. 4) yields the same reaction speeds as executing them along the concentrations record *enzlst* by the function *exec_step* in $\mathcal{B}$ (Fig. 5), because the identical kinetic function *R_kin* is applied.

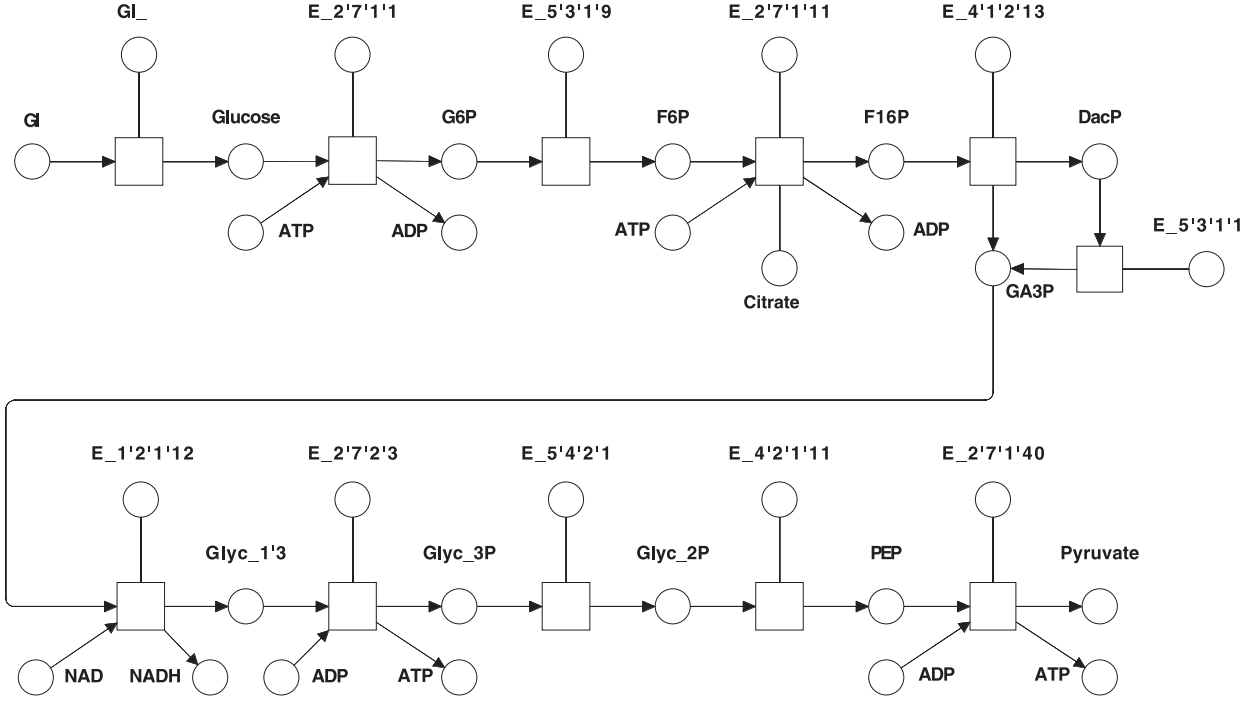
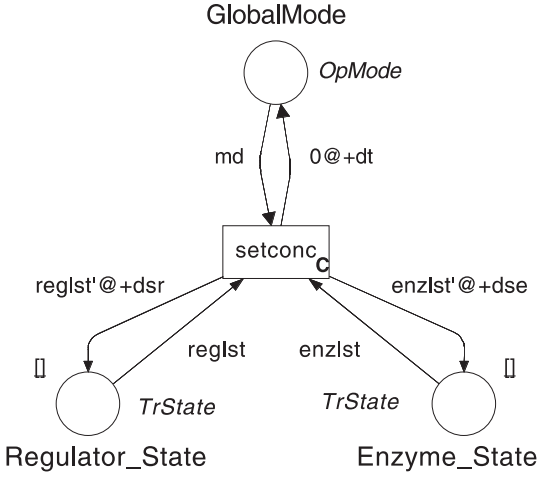


Fig. 4. The model $\mathcal{A}$ of the textbook glycolysis




```
input (enzlst,reglst,md);
output (enzlst',dse,reglst',dsr,dt);
action
let
 val ((enl,dse),(rgl,dsr),dt) =
   if md=0 then          (* execute reactions *)
     (exec_step(ENZYM,enzlst),
      exec_step(ENVIR,reglst),0)
   else if md=(~1) then  (* initialize lists *)
     (init_list(!pEnz_List,md),
      init_list(!pReg_List,md),1)
   else (*md>0*)                (* update lists *)
     (upd_list(!pEnz_List,enzlst,md),
      upd_list(!pReg_List,reglst,md),1)
in (enl,dse,rgl,dsr,dt)
end;
```

Fig. 5. The kernel page of the net model $\mathcal{B}$

are encoded. To be more specific, this colour set is a list of pairs (time t, $sublist(t)$) whose first member, t, determines the simulation time at which those reactions included in the $sublist(t)$ have to occur next. Thus, at each simulation time t, only the $sublist(t)$ has to be inspected for the reactions to be executed. The members of each sublist are quadruples containing the enzyme name (characterizing the reaction), its concentration (constant), the last step width and the last speed of the reaction.

The place *Enzyme_State* stores all information for the proper execution of the enzymatic reactions through the function *R_kin*. As mentioned above, *R_kin* takes the metabolic constants of the enzyme from the enzyme catalogue and the current concentrations of the substrates from the concentrations record.

The place *Regulator_State* deals with metabolic processes that are attributed to the environment of the modelled (glycolysis) pathway. Such reactions are necessary, e.g., to regulate particular ubiquitous molecules or to provide for steady sources and sinks of the entire pathway (see Sect. 3). Normally this kind of reaction is not catalyzed by an enzyme, rather its speed is controlled by a (properly chosen) factor which – in this context – can be treated like an enzyme.

The transition *setconc* serves three purposes, distinguished by the value of md, the *global mode*:

1. At the beginning of the simulation ($md = -1$) the lists in *Enzyme_State* and *Regulator_State* are initialized via the function *init_list*.

2. Later, during the simulation ($md = 0$), the function *exec_step* is applied, which invokes *R_kin* for all enzymatic and environment reactions that are due at the current time, say tc, i.e., are a member of the $sublist(tc)$. These reactions are executed in a random sequence to take into account their inherent concurrency. After the execution of a reaction, yielding a new step width dsx, an updated entry for the reaction is inserted into the appropriate $sublist(tc + dsx)$. In this manner, the new lists *enzlst'* and *reglst'* are generated which, at the end of this process, replace the old lists in the places *Enzyme_State* and *Regulator_State*.

Provision is taken in our metabolic net models that the set of enzymes (and hence of the enzymatic reactions) can be altered during the simulation, splitting the simulation into several so-called *simulation intervals*.[10] Prior to running the simulation, these intervals have to be specified. Each of them is characterized by one specific global mode value md with $md \in \{-1, 1, 2, 3, \dots\}$. For each interval, the corresponding enzymes and substrates together with their initial (with respect to the interval) concentrations have to be specified in a particular mode file. Moreover, for every interval, a reaction speed factor sp and the model time te of its end has to be chosen in advance. A typical interval definition could read $(md = -1, sp = 0.2, te = 600)$. In this case, every reaction speed dc computed by *R_kin* is multiplied by the factor sp. This finer granularity of the reactions is mainly needed to cope with substrate concentrations near zero to avoid meaningless negative concentrations.[11] As a result, if the total model time te equals 600 and $sp = 0.2$, then the total simulation time[12] would become $t_{real} = te/sp = 3000$.

3. Whereas the global mode value $md = -1$ is reserved for the initialization and $md = 0$ for the "normal" processing (see 1. and 2. above), modes with $md > 0$ are used to cope with a switch between intervals. To perform such a switch the function *upd_list* is applied. It updates the concentrations of the involved substances and initializes the lists in *Enzyme_State* and *Regulator_State* for the new interval to be encountered.

The entire net model $\mathcal{B}$ of the textbook glycolysis pathway has been partitioned into three sub-nets which are interconnected via shared places (called "fusion places") to yield the compound net. The most prominent subnet, modelling the metabolic processes, is shown in Fig. 5. The other subnets shall only be mentioned here.

One of them deals with the quite simple initialization of different variables needed for the simulation. The last subnet controls the management of the global mode values and the writing (of values collected during simulation) into the plot file from which – after the end of the simulation – a variety of plot diagrams can be constructed (by use of Excel) which depict the development of the concentrations and of other values of interest as a function of time.

5 Simulation results and performance

The result of each simulation is usually represented as a graph that shows the concentration/time curves for the substrates involved. These diagrams rely on the values of the substrate concentration assembled during a simulation run, and are drawn by use of Microsoft Excel with some VBA macros. A typical example of such a plot diagram is Fig. 6 showing the concentration curves[13] of the substrates participating – during a first interval of 1,000 model time steps – in the (textbook) glycolysis, followed – in a second interval also of 1,000 time steps – by the gluconeogenesis. As can be observed, towards the end of the

[10] Obviously, it would make no sense to choose totally divergent sets of enzymes (and substrates) for the intervals. Rather the intervals should, taken together, constitute again a closed metabolic pathway.

[11] A negative concentration is the result of a too wide extrapolation, an "over-reaction". Anyhow, to be on the safe side, a resulting negative concentration is always cut off to zero.

[12] The simulation time must not be confused with the real (clock) time that a simulation run takes.

[13] The curves on the negative side of the y-axis do not represent concentrations but rather integrals over the supply of glucose at the source and the consumption of pyruvate at the sink of the pathway. These integrals are shown as negative values to separate them graphically from the concentration curves.

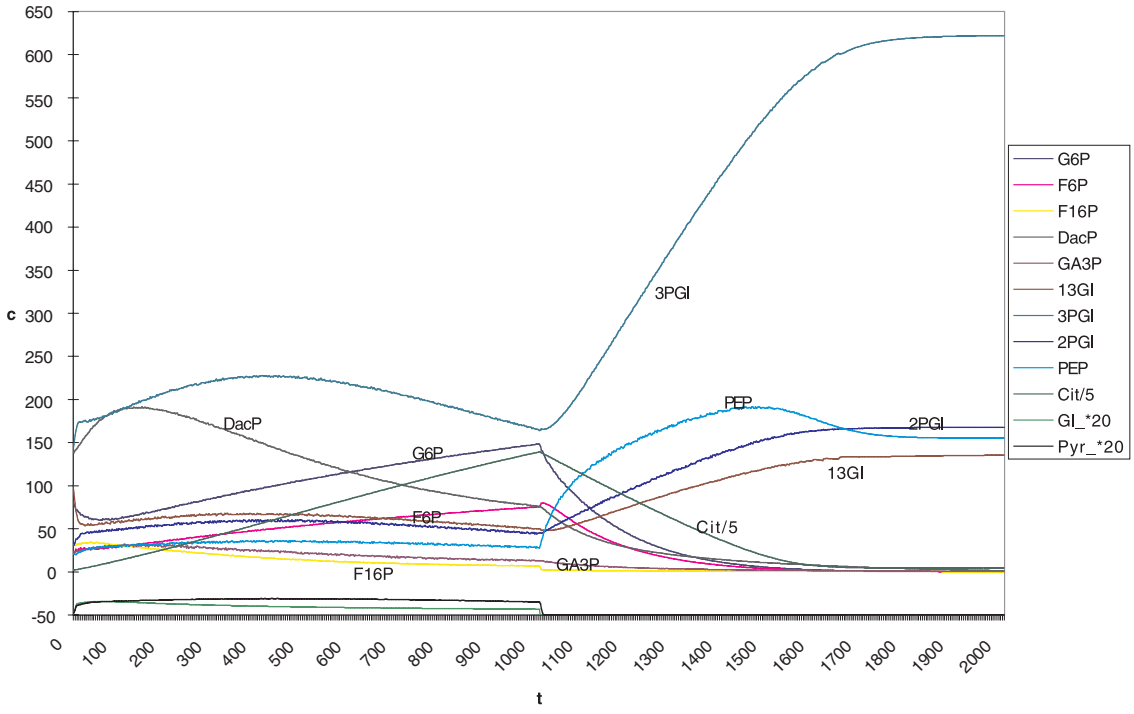


Fig. 6. A concentration diagram for the glycolysis followed by gluconeogenesis

entire simulation, each of the substrate concentrations converges, i.e., a flow equilibrium is reached, that corresponds to the results reported in textbooks like [15, 17].

For analysing metabolic processes, the temporal development of other values, e.g., of the reaction speed *dc*, may also be of interest. This allows model users to identify, most often after a number of experimental steps, the cause of a deviation from the expected equilibria and to alter the responsible values. Such a series of experimental simulation runs can lead to identifying the key enzymes which most efficiently control and influence the entire pathway processes.

Before presenting performance figures of a few typical simulation runs, it should be emphasized that the simulations have been performed on a Power Macintosh G3 under OS 8 using Design/CPN 3.0.5. Hence, we neither profited from more powerful computers nor from the faster simulation engine offered by Design/CPN nor from specialized or tuned software packages.

Table 1 originates in the simulation of three different pathways:

- (textbook) glycolysis,
- glycolysis, then the combination of glycolysis and gluconeogenesis, then again glycolysis,
- combined glycolysis and citric acid cycle.

Each pathway has been simulated, under identical conditions, with the models $\mathcal{A}$ and $\mathcal{B}$.

As mentioned in the previous Sect. 4, before a simulation run is started, among other parameters, the maximal *simulation time* has to be set. In principle, at every simulation time instance, several transition occurrences, i.e., *steps*, may happen.

In the model $\mathcal{A}$, each reaction is represented by one separate transition. Hence, the total number of steps is always much greater than the maximum simulation time. In model $\mathcal{B}$, however, one single transition stands for the

Table 1. Performance figures for models $\mathcal{A}$ and $\mathcal{B}$

Pathway	Model	Simulation time	Number of steps	Real time (s)
Glycolysis	$\mathcal{A}$	6051	45669	1127
	$\mathcal{B}$	6050	6174	70
Glycolysis, Glyc. & Gluconeogen., Glycolysis	$\mathcal{A}$	7551	59524	1579
	$\mathcal{B}$	7551	7705	78
Glycolysis & Citric acid cycle	$\mathcal{A}$	5051	42351	1070
	$\mathcal{B}$	5050	5154	69

set of all reactions. Thus, the number of steps should be equal to the simulation time.[14]

A considerable part of the *real time* consumed for a simulation run is used for calculating the enabled transitions. As a consequence, the simulations with model $\mathcal{B}$ are much faster than with model $\mathcal{A}$ (factors 16.1, 20.2, 15.5, respectively). As can be seen from the Table 1, instead of about 20 min, a typical experiment with model $\mathcal{B}$ now takes a bit more than 1 min.

6 Conclusions

Differential equations (DEs) presenting non-trivial metabolic networks usually are too complex to have closed solutions. They are solved numerically by computing iteratively, in (small) discrete time steps, the concentration changes for the involved substrates. The Petri net approach presented in this paper, in principle, follows the same line. However, a net model like $\mathcal{A}$ (Fig. 4) visualizes the structure of the modelled pathway which is hidden or not easily accessible in a DE system. The execution of such a Petri net model fully respects the inherent concurrency of the reactions, i.e., reactions having no substrates in common are executed *in one step*. Using Design/CPN or comparable tools, the net can be simulated step by step, enabling the designer to observe the details of this process.

Given a metabolic pathway, our approach offers the means to construct the corresponding net diagram (almost) automatically. Because the result is a "standard" coloured net, it is amenable to established analysis methods from the Petri net theory.

Moreover, instead of having to develop a specific kinetic equation for each reaction, we use *one* general function based on the *Michaelis-Menten* equation. The required metabolic parameters are extracted automatically from the BRENDA [1] database.

High-level Petri nets offer a substantial flexibility in constructing net models that serve different intentions but behave equivalently (see [5]). The models $\mathcal{A}$ and $\mathcal{B}$ represent the same metabolic pathways and their execution renders identical results, although their graphical appearance and their performance differs strongly. The merits of model $\mathcal{A}$ lies in its graphical structure which directly reflects the connections among the biochemical reactions. Not surprisingly, this appeal of intuition is lost in the net $\mathcal{B}$ (Fig. 5) as it constitutes a more abstract and more concise model whose only purpose is to enhance the simulation performance.

Yet, the simulation speed is one of the crucial shortcomings of our net models as compared to differential equation models. The main reasons for that can be found in the modest hardware equipment used, and in the application of a general, non-specialized software tool, which includes time-consuming graphics components. Accordingly, the approach presented in this paper is not intended to compete with the established methods in biochemistry to conduct large-scale simulations.

Rather, we argue that a Petri net view may contribute additional and complementary aspects to the computer-based study of metabolic systems. Hence, our approach is to present an alternative that we hope will stimulate research and development in this important area.

In this paper we restrict ourselves to purely metabolic networks and leave out so-called regulatory mechanisms[15] (i.e., enzymes directly activating or inactivating other enzymes by modifying them). The kinetic mechanism of regulatory enzymatic reactions is much less understood compared to metabolic ones. Modelling regulatory enzyme relations graphically as a Petri net would create no major difficulties. The problem lies in finding appropriate kinetic functions and their parameters. A sample net is shown in Fig. 7 (see footnote 7).

Prospective future work in the area of metabolism is mainly determined by the needs and plans of a project in biochemistry we are collaborating with at GMD–SCAI. One favoured research direction will make use of annotated sequence databases and organism-specific databases to complement the metabolic information which then would lead to a more complete coverage of the functions coded in a genome.

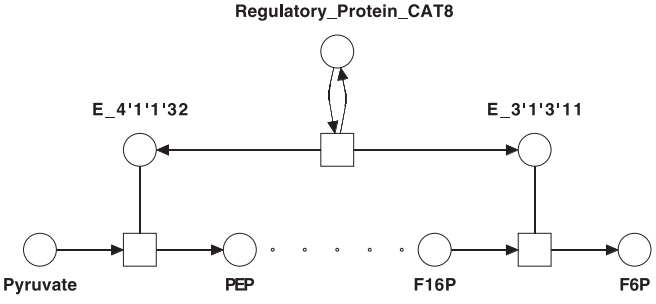


Fig. 7. The model of a regulated pathway (gluconeogenesis)

Acknowledgements. We are indebted to Kurt Jensen for his suggestions concerning the enhancement of the model performance. We also would like to thank the anonymous reviewers for their valuable remarks and hints.

[14] The number of steps is slightly greater because the transitions that are in charge of writing the actual values into the plot file have to be added.

[15] Regulatory mechanisms include
(1) Transcription control: control of biosynthesis of enzyme proteins through regulator proteins,
(2) Interconversion: switching enzyme from active to inactive, and vice versa, through (in-)activating enzymes by signals of e.g., hormones via Second Messengers,
(3) Modulation by ligands, e.g., by coenzymes or diverse inhibitors.

References

1. BRENDA. The Comprehensive Enzyme Information System. http://www.brenda.uni-koeln.de:80/. See: Schomburg, D., Salzmann, D., Stephan, D.: Enzyme handbook, classes 1–6. Berlin, Heidelberg, New York: Springer-Verlag, 1990–1995
2. Design/CPN. http://www.daimi.au.dk/designCPN/
3. ENZYME. Enzyme nomenclature database. http://www.expasy.ch/enzyme/. See: Bairoch, A.: The ENZYME data bank in 1999. Nucleic Acids Res 27(1): 310–311, 1999
4. Franco, R., Canela, E.: Computer simulation of purine metabolism. Eur J Biochem 144: 305–315, 1984
5. Genrich, H.J.: Equivalence transformations of PrT-Nets. In: LNCS 424. Berlin, Heidelberg, New York: Springer-Verlag, 1989, pp. 179–208
6. Jensen, K.: Coloured Petri nets. Basic concepts, analysis methods and practical use. Vol. 1–3. Monographs in Theoretical Computer Science. Berlin, Heidelberg, New York: Springer-Verlag, 1992–1997
7. KEGG. Kyoto encyclopedia of genes and genomes. http://www.genome.ad.jp/kegg/. See: Ogata, H., et al.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res 27: 29–34
8. Lee, I.-D., Palsson, B.O.: A Macintosh software package for simulation of human red blood cell metabolism. Computer methods and programs. In: Biomedicine 38: 195–226, 1992
9. Hofestädt, R.: A Petri net application of metabolic processes. J Syst Anal Modelling Simulation 16: 113–122, 1994
10. Küffner, R., Zimmer, R., Lengauer, T.: Pathway analysis in metabolic databases via differential metabolic display (DMD). Bioinformatics 2000 (submitted)
11. Matsuno, H., Doi, A., Nagasaki, M., Miyano, S.: Hybrid Petri net representation of gene regulatory network. In: Proc. 5th Pacific Symposium on Biocomputing. World Scientific, Singapore, 2000, pp. 338–349
12. Reddy, V.N., Liebman, M.N., Mavrovouniotis, M.L.: Qualitative analysis of biochemical reaction systems. Comput Biol Med 26(1): 9–24, 1996
13. Reddy, V.N., Mavrovouniotis, M.L., Liebman, M.N.: Petri Net representation in metabolic pathways. In: Hunter, L., et al. (eds.): Proc. 1st Intern. Conf. on Intelligent Systems for Molecular Biology, AAAI, Menlo Park, 1993, pp. 328–336
14. Schuster, R., Holzhütter, H-G.: Use of mathematical models for predicting the metabolic effect of large-scale enzyme activity alterations. Application to enzyme deficiencies of red blood cells. Eur J Biochem 229: 403–418, 1995
15. Stryer, L.: Biochemistry. W.H. Freeman, New York, 1996
16. Tomita, M., et al.: E-CELL: software environment for whole-cell simulation. Bioinformatics 15(1): 72–84, 1999
17. Voet, D., Voet, J.G.: Biochemistry. Wiley, New York, 1990

Appendix A: Glossary

Metabolites

ADP	Adenosine diphosphate
ATP	Adenosine triphosphate
DacP	Dihydroxyacetone phosphate
F16P	Fructose biphosphate
F6P	Fructose-6-phosphate
G6P	Glucose-6-phosphate
GA3P	Glyceraldehyde-3-phosphate
Gl	Glucose reservoir
Gluc	Glucose
Glyc_1′3	1,3-Biphosphoglycerate
Glyc_2P	2-Phosphoglycerate
Glyc_3P	3-Phosphoglycerate
NAD	Nicotinamide adenine dinucleotide
	... reduced form, NAD^+
NADH	... oxidized form
PEP	Phosphoenolpyruvate
Pyr	Pyruvate

Enzymes

EC number	Enzyme name
1.2.1.12	GA3P dehydrogenase
2.7.1.1	Hexokinase
2.7.1.11	Phosphofructokinase
2.7.1.40	Pyruvate kinase
2.7.2.3	Phosphoglycerate kinase
4.1.2.13	Aldolase
4.2.1.11	Enolase
5.3.1.1	Triosephosphate isomerase
5.3.1.9	Phosphoglucose isomerase
5.4.2.1	Phosphoglycerate mutase