The Living Cell as a Multi-Agent Organisation: A Compositional Organisation Model of Intracellular Dynamics

C.M. Jonker[1], J.L. Snoep[2,4], J. Treur[1],
H.V. Westerhoff[3,4] and W.C.A. Wijngaards[1]

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, NL-1081 HV Amsterdam, The Netherlands, EU,
{jonker, treur, wouterw}@cs.vu.nl

[2]University of Stellenbosch, Department of Biochemistry, Private Bag X1,
Matieland 7602, Stellenbosch, South Africa, jls@maties.sun.ac.za

[3]Stellenbosch Institute for Advanced Study, South Africa and [4]BioCentrum
Amsterdam, Department of Molecular Cell Physiology, De Boelelaan 1087, NL-
1081 HV Amsterdam, The Netherlands, EU, hw@bio.vu.nl

Abstract
Within the areas of Computational Organisation Theory and Artificial Intelligence, techniques have been developed to simulate and analyse dynamics within organisations in society. Usually these modelling techniques are applied to factories and to the internal organisation of their process flows, thus obtaining models of complex organisations at various levels of aggregation. The dynamics in living cells are often interpreted in terms of well-organised processes, a bacterium being considered a (micro)factory. This suggests that organisation modelling techniques may also benefit their analysis. Using the example of *Escherichia coli* it is shown how indeed agent-based organisational modelling techniques can be used to simulate and analyse *E.coli*'s intracellular dynamics. Exploiting the abstraction levels entailed by this perspective, a concise model is obtained that is readily simulated and analysed at the various levels of aggregation, yet shows the cell's essential dynamic patterns.

Keywords
organisational modeling, intracellular, dynamics, modular control analysis, regulation and control

1 Introduction

In the area of modelling intracellular processes, the most widely used approach is based on differential equations, which are integrated numerically (Westerhoff, 2001; Stuart and Humphries, 1996). For some small unicellular organisms, a few isolated chemical pathways are understood in sufficient kinetic detail to obtain a description of their import and primary processing of nutrients; e.g., for *Escherichia coli*, blood cells and yeast (Rohwer et al., 2000, Teusink *et al*., 2000; Wang *et al*., 2001, Ben-Jacob *et al*., 1997; Rizzi et al., 1997; Takahashi, Ishikawa, Sadamoto, Sasamoto, Ohta, Shiozawa, Miyoshi, Naito, Nakayama and Tomita, 2003; Snoep, 2005; Jamshidi and Palsson, 2006). However, this approach has difficulties when tackling larger cellular systems. First, hundreds or more reaction

parameters are needed, for which reliable values are rarely available (Teusink *et al*., 2000; Kholodenko *et al*., 1999). This can seriously compromise the feasibility of the general model. Second, actual behaviour of intracellular pathways may be much less complex than is theoretically possible on the basis of the complexity of the chemical processes (e.g., Rotterdam *et al*., 2002). At best, and only if all system parameters and internal connections are known and sufficiently tuned, the traditional approach delivers a computer replica of (part of) the living cell. If this replica functions correctly, then it can be seen as a validation of the system parameters and of the knowledge obtained for the different pathways. However, if the replica does not function correctly, the huge number of parameters makes localizing aberrations practically impossible. Furthermore, the replica is almost as remote from human understanding as the target system itself. This is because the modelling approach requires a description that is complete, inherently low-level, detailed and complex. In contrast, the human mind operates by abstraction in order to understand an essence.

Conceptual analysis of the cell's internal functioning from a biological perspective often leads to descriptions where specific processes function together according to some form of organisation. For example, viewed from a global perspective (top-down), sub-processes of the overall process are distinguished such as transcription, translation, and metabolism, and these sub processes interact with each other according to a structured pattern. Viewed from a more local perspective (bottom-up), groups of specific biochemical reactions are 'lumped' together to form functional units. This type of approach recognizes that some conglomerates of biochemical processes act as functional units such as "metabolic pathway", "catabolism", "transcriptome" and "regulon". Some of these concepts have been or are being defined formally (Kahn & Westerhoff, 1991; Rohwer *et al.*, 1996; Schilling *et al.*, 2000), but implementation is still in its infancy. This perspective involves modelling the overall process at different levels of aggregation based on functional units. Crucial challenges are:

(1) how to describe the functionality of such a functional unit,

(2) how to describe the manner in which multiple functional units interact and co-operate to obtain a well-organised overall process, and

(3) how to implement software support for simulation and analysis based on these different aggregation levels

To manage complex dynamics in human societies, organisational structures are also often exploited. Within the area of Computational Organisation Theory and Artificial Intelligence, in particular Agent Systems, organisation modelling techniques have been developed to simulate and analyse dynamics within organisations in society. For example, Pajares et al. (2003) present an agent-based organisation model of industry; representing firms by agents that make strategic decisions on investments, product innovation and whether to stay or leave the industry. The manageability of the dynamics emerging from multiple agents in a society depends on some form of organisational structure. An organisational model for a multi-agent system provides a structuring of the processes in such a manner that an agent involved can function appropriately. The dynamics within a given organisational structure is much more dependable than in an entirely unstructured situation. Usually these organisation modelling techniques are applied to factories and the

internal organisation of their process flows, obtaining high-level models of complex organisations at different levels of aggregation.

Dynamics at the different levels of aggregation within an organisation are related to each other. In particular, the dynamics of the whole process depends on dynamics of processes at lower levels of aggregation, i.e. more specific processes. Lomi and Larsen (2001) emphasize the importance of such interlevel relationships. Organisations can be seen as adaptive complex information processing systems of (bounded) rational agents, and as tools for control; central questions are (Lomi and Larsen, 2001):

- from the first view: 'given a set of assumptions about (different forms of) individual behaviour, how can the aggregate properties of a system be determined (or predicted) that are generated by the repeated interaction among those individual units?'
- from the second view: 'given observable regularities in the behaviour of a composite system, which rules and procedures - if adopted by the individual units - induce and sustain these regularities?'.

Both views and problems require means to express relationships between dynamics at different levels of aggregation. In our approach logical relationships between dynamic properties at different aggregation levels provide a manner to express such interlevel relationships (mathematically).

Literature on Organisation Theory is largely informal or semi-formal (see for example Mintzberg, 1979). The idea of using simulation as a formal technique to research organisational dynamics stems already from the 1950s. However, the power of computers then restricted the applicability of the simulations of those times. Although several results based on those simulations were frequently cited in the literature, simulation did not become a popular tool in sociological or biological research.

Recently formal and computational modelling techniques have received more attention within Organisation Theory. Modellers base themselves on recent developments within Organisation Theory, involving concepts like organizational behaviour and adaptation, organizational embeddedness, organizational ecology, and competitive survival. Examples of this formalisation trend can be observed in books such as (Prietula, Gasser, and Carley, 1997; Lomi and Larsen, 2001), and in recently created journals such as Computational and Mathematical Organisation Theory (e.g., Moss et al., 1998). Executable models serve as a basis for simulation experiments. These can be used, for example, in evaluating sample behaviours of (real or simulated) organisations. A language for executable models should be formal, and not too complex, to avoid computational complexity. Software tools to support such a language serve as *simulation environment* (e.g. Moss, Gaylard, Wallis and Edmonds, 1998; Prietula, Gasser, and Carley, 1997).

Within the agent systems area a number of organisation modelling approaches have been developed. One of them is the Agent-Group-Role-approach (AGR) introduced in (Ferber and Gutknecht, 1998), and extended with a dynamic modelling language in (Ferber et al., 2001). Within this approach the organisational structure is the specification of a specific multi-agent organisation based on a definition of groups, roles and their relationships within the organisation. An organisation as a whole is composed of a number of groups (e.g., divisions or departments). A group structure identifies the roles and

(intragroup) role interaction within a group, and the transfers between roles needed for such interactions. In addition, intergroup role relations between roles of different groups specify the connectivity of groups within an organisation. In such an organisation model, a number of descriptions are basic for the dynamics: roles fulfilled by agents, groups consisting of a number of roles, and of interactions between roles and between groups. A limitation of the AGR approach is that only three aggregation levels can be modelled. To smoothly model an organisation comprising an arbitrary number of aggregation levels, and as an alternative to the AGR approach, the compositional organisation modelling approach is introduced and exploited to model organisational structures.

An organisation structure model by itself provides no dynamics. In a sense it abstracts from the dynamics. However, to be able to analyse and/or simulate dynamics within an organisation, as part of an organisation model, also some specification of *dynamic properties* is required. Dynamic properties relate states of the organisation over time. Usually one particular dynamic property refers not to the whole state but to a limited set of specific elements or aspects of these states. This set can be viewed as the scope of a dynamic property. Depending on the property at hand, this scope can be broad (or global), or narrow (or local). For example, global properties of an organisation as a whole may refer to a number of different aspects (sometimes such a property is called *integrative*), whereas properties of the interaction between two specific roles within an organisation will refer only to aspects related to these roles (e.g., a role interaction protocol). Thus to obtain an adequate description of the complex interacting processes at different levels of aggregation, descriptions in terms of dynamic properties are used: an *organisation dynamics model*. Section 9 addresses the software environment used for simulation and analysis. Section 10 shows how well known numerical integration methods for differential equations can be incorporated in the simulation framework used here; Section 11 is a discussion. Appendix A contains more details on the organisation model of *E. coli*.

In Nature several forms of organisational structure have evolved. Examples include insect organisations such as ant-hills, beehives, wasp-hives, as wells as herds, wolf-packs, the coordinated processes of organs in vertebrates, and last but not least the living cell itself. For some of these, simulations can be found in literature; however, these do not invoke structures developed in Organisation Theory. In this paper it is investigated whether and how organisation modelling techniques can be used to model the complex dynamics in living cells.

2 Modelling Approach: Organisation Structure

Within the Agent-Group-Role or AGR organisation modelling approach (Ferber and Gutknecht, 1998), an organisation structure is described at three aggregation levels: the *organisation* consists of a set of *groups*, and each group consists of the *roles* in that group. Furthermore, *connections* between roles and between groups are possible; see Figure 1. Here the smaller ovals indicate roles and bigger ovals groups. Connections are indicated by the two types of arrows (dashed indicates an intergroup interaction, not dashed indicates a transfer). To indicate which role belongs to which group is depicted by drawing the smaller role oval within the bigger group oval. Moreover the organisation is *realized* by *agents* fulfilling roles (not depicted). The main concepts are briefly described as follows:

- The *agents*. The model places no constraints on the internal architecture of agents. An agent is only specified as an active communicating entity which plays roles within groups. This agent definition is intentionally general to allow agent designers to adopt the most accurate definition of agent-hood relative to their application.
    - A *group* is defined as an atomic set of agent roles. Each agent takes part by fulfilling roles in one or more groups. In its most basic form, the group is only a way to name a set of roles.
    - A *role* is an abstract representation of an agent function or service within a group. Each agent can handle multiple roles, and each role handled by an agent is local to a group.

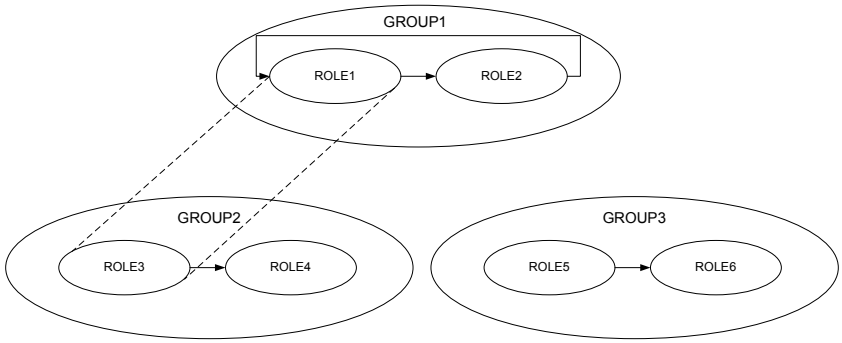


Figure 1. Example organization modeled within AGR

The use of a compositional modelling strategy is common in Computer Science, Artificial Intelligence and Organisation Theory. It has been instantiated in different forms, as functional design, modularized design, task analysis or decomposition, object oriented design, component-based design and agent oriented design, among others (cf. Tanenbaum, 1976; Knuth, 1981; Booch, 1991; Brazier, Jonker and Treur, 2002; Ferber and Gutknecht, 1998). The main point is that the system is too complex to be understood when presented in a direct, flat manner, thus some elements of the model have to be grouped together, and these groups can then be further grouped. The differences between the approaches mentioned are in the choice of grouping and possible combinations of these groups and elements. A key design issue then of compositional modelling is the criterion of grouping processes together. Where, and when can a (process) component be separated from another component, and what interactions the components can have is determined. Note that in this paper the word *component* indicates a *process component*.

Compositionality provides means for *information hiding*. When a component contains several elements that are not visible to the components interacting with the component as a whole, these elements are said to be hidden; some information in the model is not visible to the other components. This is important, as it allows the amount of information at higher levels in the component structure that is visible, to be less than the total amount of information of all the parts and thus give a more manageable view of what is happening. Of

course, an item can only be hidden if it does not affect anything it is hidden from, except through its non-hidden encompassing component (cf. Rohwer et al, 1996).

Compositional modelling proves most effective when building the model. Both in directing effort in breadth - as the abstract overviews give a means to see if everything is covered - and in depth, when at a specific point more detail is needed, the compositional model allows (at a certain aggregation level) the composition of a role (*parent role*) out of several sub-roles (*child roles*) to effectuate the detailing. Parts of the system may be modelled in increasing detail, in a so-called *refinement*. In the example above, first the factory role can be modelled according to its division roles, a division role can be refined according to its department roles and a department role can be further refined to unit roles within the department.

The organisation of the cell as a whole consists of the environment and the cell. The cell is in interaction with the environment. In Figure 2 the hierarchical structure behind the highest aggregation levels of the organisation model is depicted. In this picture the right hand side nodes connected to a node are called the children of the latter node, which itself is called a parent node for those children. For example, the node Cell is the parent node of the nodes Transcription, Translation, and Metabolism. The latter nodes are children of Cell.

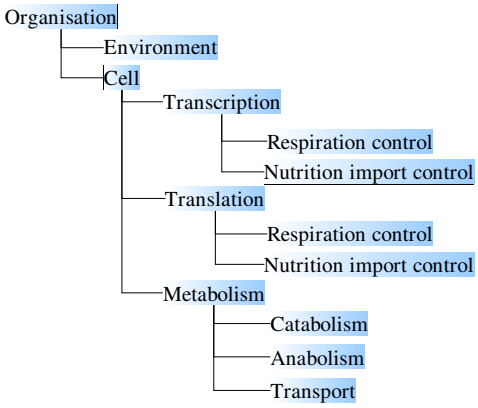


Figure 2. Overview of the hierarchy behind the highest aggregation levels of the organisation model.

In Figure 3 the organisation structure of the highest aggregation levels of the model is depicted. 'Roles' within a group are depicted by small ovals within a bigger oval (cf. Fig. 1). In particular, the cell's functioning is based on three roles: Transcription, Translation, and Metabolism. Within Biochemistry, what is indicated by the modelling concept 'role' is commonly called a 'process'. This distinction will be made in the remainder of the paper: for the modelling entity the word 'role' is used, for the entity in the context of the biological domain the word 'process'.

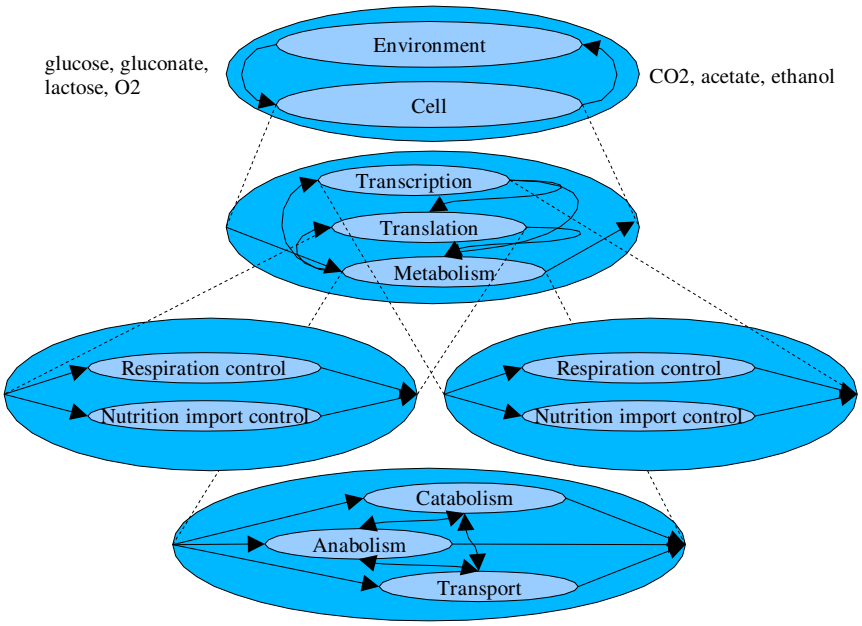


Figure 3. The highest aggregation levels of the organisation structure

Each of these roles can again be viewed as a group, as indicated by the dotted lines connecting to a large oval below. Arrows between small ovals within a bigger oval indicate information transfer between roles. An arrow between a small oval and the encompassing larger oval indicates (interlevel) interaction between the role indicated by the small oval (the child role) and the group indicated by the bigger oval (relating to the parent role of the latter). Given the dotted lines, this entails (interlevel) interaction between the child role and the parent role.

Both the Transcription group and the Translation group include the roles Respiration Control and Nutrition Import Control. The Metabolism group includes the roles Catabolism, Anabolism and Transport.

3 Modelling Approach: Organisation Dynamics

To be able to simulate or analyse dynamics within an organisation, in addition to the static organisation structure specification discussed above, as part of an organisation model also a specification of the dynamics within the organisation is needed. To this end, in the specification of an organisation model different types of specifications of *dynamic properties* are distinguished. These properties serve as constraints on the dynamics of the respective roles and interactions. In (Jonker, Treur, and Wijngaards, 2002) an executable temporal language was introduced to specify the different types of dynamic properties.

After a more general introduction, this temporal language to specify dynamic properties is introduced.

Within a compositional organisation model dynamic properties can be specified for each of the *roles* at the different aggregation levels. Furthermore, dynamic properties can be specified for *transfer* between roles at a given aggregation level. Moreover, *interlevel interaction dynamics* can be specified in the form of dynamic properties relating output of a child role to output of its parent role, or relating input of the parent role to input of a child role.

If specified appropriately, the dynamic properties of a role at a higher aggregation level are related to the dynamic properties of its child roles. The general pattern is the following logical implication:

dynamic properties for children roles &
dynamic properties for transfer between children roles &
dynamic properties for interlevel interaction between parent role and children roles
$\Rightarrow$ dynamic properties for parent role

In more mathematical notation this can be expressed as follows. Here P denotes a parent role with (a conjunction of) dynamic properties DP(P), and with children roles C1, C2, C3, with dynamic properties DP(C1), DP(C2), DP(C3), respectively. Moreover, TRD(R) denotes the dynamic properties for transfer between children roles of role R, and IID(R) the dynamic properties of interlevel interaction between role R and its children roles. Using these notations, the pattern above can be expressed as:

$$DP(C1) \,\&\, DP(C2) \,\&\, DP(C3) \,\&\, TRD(P) \,\&\, IID(P) \quad \Rightarrow \quad DP(P)$$

as before, & means conjunction, and $\Rightarrow$ implication, i.e., if all dynamic properties on the left hand side hold, then the right hand side holds.

Notice that this can be iterated if each (or some) of the children roles are themselves parent role for other children roles. Given such an implication, if a particular trace of organisation dynamics satisfies the properties of the child roles, and the transfer and interlevel interaction properties, then it will also satisfy the dynamic properties of the parent role. Applied in a recursive manner, this implies that properties of the organisation as a whole can be obtained from (or realised by) properties at lower aggregation levels.

In Biology also specific forms of organisation have been developed and exploited. In particular, for the processes in the cell the following main categories or functional units are distinguished: metabolism, translation and transcription (cf. Wijker et al., 1995). These are the main parts of the regulation and control cycle of a cell. The metabolism expands to catabolism, anabolism and transport. The catabolism is the category of processes that decompose substances and extract free energy from them. In the anabolism the processes reside that utilize this free energy to create more and more complex substances reside. The transport processes move substances across the cell membrane.

Section 4 shows in more detail how a compositional organisation modelling approach helps to manage the complexity of intracellular processes. The reasons for grouping certain processes together will be in accordance with biological knowledge. The executable

temporal language used is a temporal language extending the paradigm of Executable Temporal Logic (Barringer et al., 1996; Fisher, 1994, 2005) with real-valued time. Roughly spoken, in this executable language it can only be expressed that if a certain state property holds for a certain time interval, then after some delay another state property should hold for at least a certain time interval; see also (Bosse, Jonker, Mey, and Treur, 2007). The *LEADSTO* language enables one to model direct temporal dependencies between two state properties in successive states. A specification of dynamic properties in LEADSTO format has as advantages that it is executable and that it can often easily be depicted graphically. For the approach described in this paper, the choice has been made to consider time as continuous, described by real values, but for state properties, both quantitative and qualitative variants can be used. The approach subsumes approaches based on simulation of differential or difference equations, and discrete qualitative modelling approaches, but also combines them. For example, it is possible to model the exact (real-valued) time interval for which some qualitative property holds. Moreover, the relationships between states over time are described by either logical or mathematical means, or a combination thereof. This will be explained below in more detail.

Dynamics is considered as evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of properties that do or do not hold at a certain point in time. Ontologies are specified as signatures in order-sorted predicate logic, i.e., sets of sorts and subsort relations, constants in sorts, functions and predicates over sorts.

Definition (State Properties)

Let Ont be a given ontology Ont.

a) The set of *state atoms* (or *atomic state properties*) based on Ont is denoted by APROP(Ont), and the set of *state ground atoms* by GAPROP(Ont).

b) The set of *state properties* STATPROP(Ont) based on Ont consists of the propositions that can be made (using conjunction, negation, disjunction, implication) from the atoms. Moreover, GSTATPROP(Ont) is the subset of *ground state properties*, based on ground atoms. A subset of the set of state properties is the set CONLIT(Ont) of *conjunctions of literals (atoms or negations of atoms)*.

The textual LEADSTO format is defined as follows.

Definition (LEADSTO format)

Let a state ontology Ont be given.

Any expression for Ont of the form

$$\forall x_1, ..., x_n \quad \alpha \rightarrow\!\!\!\!\rightarrow_{e, f, g, h} \beta$$

where α (the *antecedent*) and β (the *consequent*) are state properties in CONLIT(Ont), with variables among x_1, .., x_n, and e, f, g, h non-negative real numbers, is a LEADSTO *expression*. When no variables nor quantifiers occur in this expression, it is called a LEADSTO *ground expression*.

Informally, for the case without variables, a LEADSTO expression $\alpha \rightarrow\!\!\!\!\rightarrow_{e, f, g, h} \beta$ means (also see Figure 4):

If state property α *holds for a certain time interval with duration g, then after some delay (between e and f) state property* β *will hold for a certain time interval of length h.*

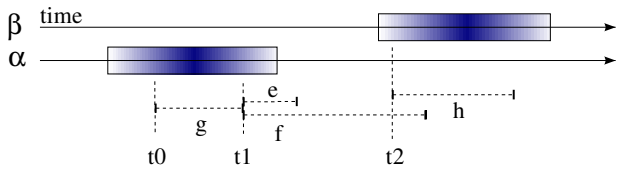


Figure 4. The timing relationships

Within the LEADSTO language it is possible to use sorts, variables over sorts, real numbers, and mathematical operations, such as in the property (where x is a constant):

$$\forall v \; \text{has\_value}(x, v) \rightarrow\!\!\!\!\rightarrow_{e, f, g, h} \text{has\_value}(x, v*0.25)$$

This property expresses the fact that, if has_value(x, v) holds during g time units, then after a delay between e and f time units, has_value(x, v*0.25) will hold during h time units.

The definition of the relationships as given above, will also manage situations where the sources hold for longer than the minimum interval length g. The total duration that the source holds, is also added to the duration that the result will hold, provided $e + h \geq f$. This is because under the given constraint the definition can be applied at each subinterval where α holds, resulting in many overlapping intervals of β. The end result is that the additional duration also extends the duration that the resulting notion β holds. Below we shall use seconds for the unit of time for e, f, g, and h.

The dynamics of each of the roles in the cell model is specified in LEADSTO format. For each of the roles, first it is indicated what are inputs and outputs for the role, and next what characterises its dynamics. The environment for the cell can be used to specify environmental conditions over time. In the model it can be used to specify as output of the environment (to be used by the input of the cell) environmental conditions concerning the presence of glucose, gluconate, lactose, O_2, N, P, S. Moreover, depending on the functioning of the cell, the environment receives as input the presence of some substances produced by the cell, i.e., the presence of CO_2, ethanol and acetate. This view on the environment is that it is a component whose output provides the cell with input, and whose input is fed by the output of the cell. The current model is specified for an experimental setup where a bacterial culture is bubbled through with oxygen and nitrogen, quickly flushing out CO_2. It is also possible to model CO_2 or acetate as staying present in the environment once exported, simply by adding a property specifying that the presence of the substance continues once it has arrived.

Input: present(CO2_outside), present(ethanol_outside), present(acetate_outside)
Output: present(glucose_outside), present(gluconate_outside), present(lactose_outside), present(O2_outside), present(N_outside), present(P_outside), present(S_outside)

An example of a 'follows' relation to specify environmental conditions is the following, the property ED1 (for Environmental Dynamics):

ED1
true $\rightarrow\!\!\!\!\rightarrow_{1;5;10;10}$ output: present(glucose_outside) & present(O2_outside) & present(N_outside) & present(P_outside) & present(S_outside)

4 The Cell and its Three Main Roles

From the top-level perspective the cell can be viewed as a single role that interacts with its environment. Within the cell this role is organised according to three main roles. Dynamic properties in 'leads to' format characterising the dynamics of these roles are specified in more detail below. Note that the chosen model corresponds closely to the way instructors teach their students about the behaviour and functions of the cell. The model is at a high level of abstraction, precisely for this reason. It is easy to talk about the cell in this way. One of the added values of this paper is that the runtime behaviour of the model shows that such a high level of abstraction can be used and still have behaviour that is correct at that level of abstraction. Furthermore, the abstractions are not created on the verbal accounts of the instructors only, but for each abstraction a mapping exists to the underlying concepts in the cell down to the level of mmol/liter of substances. The mapping was made using detailed knowledge in the literature and otherwise based on our own expertise in this area. From a philosophy of science perspective, our approach corresponds to a qualitative model based on causal relations.

4.1 Cell

The cell can use as input from the environment the presence of glucose, gluconate, lactose, O_2, N, P, S. It may produce CO_2, ethanol and acetate (apart from cell growth).

Input: present(glucose_outside), present(gluconate_outside), present(lactose_outside), present(O2_outside), present(N_outside), present(P_outside), present(S_outside)
Output: present(CO2_outside), present(acetate_outside), present(ethanol_outside)

Viewed at the highest aggregation level, the cell's dynamics can be described by a number of temporal input-output relations, in 'leads to' language. These dynamic properties specify under which environmental conditions the cell produces what particular output for the environment. For example, properties CD1, CD2, CD3 (here CD stands for Cell Dynamics) specify that if O_2 is available, as well as at least one of the nutrients glucose, lactose, gluconate, and resources, then the cell produces CO_2. The other properties specify the anaerobic case. The conjunction of all of these dynamic properties is denoted by DP(Cell).

CD1
input: present(O2_outside) & present(glucose_outside) & present(N_outside) & present(P_outside) & present(S_outside) $\rightarrow\!\!\!\!\rightarrow_{72;216;4;4}$ output: present(CO2_outside)

CD2
input: present(O2_outside) & present(lactose_outside) & present(N_outside) & present(P_outside) & present(S_outside) & not present(glucose_outside) $\twoheadrightarrow_{72;216;4;4}$ output: present(CO2_outside)

CD3
input: present(O2_outside) & present(gluconate_outside) & present(N_outside) & present(P_outside) & present(S_outside) $\twoheadrightarrow_{72;216;4;4}$ output: present(CO2_outside)

CD4
input: not present(O2_outside) & present(glucose_outside) & present(N_outside) & present(P_outside) & present(S_outside)
$\twoheadrightarrow_{72;216;4;4}$ output: present(acetate_outside) & present(ethanol_outside)

CD5
input: not present(O2_outside) & present(lactose_outside) & present(N_outside) & present(P_outside) & present(S_outside)
$\twoheadrightarrow_{72;216;4;4}$ output: present(acetate_outside) & present(ethanol_outside)

CD6
input: not present(O2_outside) & present(gluconate_outside) & present(N_outside) & present(P_outside) & present(S_outside)
$\twoheadrightarrow_{72;216;4;4}$ output: present(acetate_outside) & present(ethanol_outside)

4.2 Metabolism

The role Metabolism, which includes import and export, can use substances present outside the cell, but also substances produced by Translation, or Transcription (ADP, P). The enzymes produced by translation function as catalysts in metabolism. It can produce substances that are exported to the environment, as well as substances used by the other two roles, e.g., amino acids for Translation, and nucleotides for Transcription.

Input: present(glucose_outside), present(gluconate_outside), present(lactose_outside), present(O2_outside), present(N_outside), present(P_outside), present(S_outside), present(fermentation_enzymes), present(respiration_enzymes), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes), present(ADP), present(P)

Output: present(ATP), present(CO2_outside), present(acetate_outside), present(ethanol_outside), present(nucleotides), present(aminoacids), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount), present(ArcB_P)

The following dynamic properties specify that under appropriate environmental conditions the Metabolism will produce ATP, amino acids and nucleotides (apart from cell growth).

The timing is given as 72;216;4;4 which means that e=72, f=216, g=4 and h=4. Thus after 4 seconds have passed that the antecedents hold, then a delay between 72 and 216 seconds passes, after which the consequent holds for 4 seconds. Note that continued holding of the antecedent for more than 4 seconds will lead to continued holding of the consequent. Also note that after the consequent duration has passed, the property does not specify whether the consequent will then hold or not, it could be true or false. If other properties affect the same consequent, this consequent may continue to hold even though the current property does not affect it any more.

MD1
input:not present(glucose_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output: present(CRPcAMP).

MD2
input: present(lactose_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(allolactose).
MD3
input: present(gluconate_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output present(gluconate6P_observation_amount).
MD4
input: present(glucose_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & present(O2_outside) & present(glucose_import_enzymes) & present(respiration_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output:present(CO2_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD5
input: present(lactose_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & present(O2_outside) & present(lactose_import_enzymes) & present(respiration_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output:present(CO2_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD6
input: present(gluconate_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & present(O2_outside) & present(gluconate_import_enzymes) & present(respiration_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output:present(CO2_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD7
input: present(glucose_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & not present(O2_outside) & present(glucose_import_enzymes) & present(fermentation_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output: present(acetate_outside) & present(ethanol_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD8
input: present(lactose_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & not present(O2_outside) & present(lactose_import_enzymes) & present(fermentation_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output: present(acetate_outside) & present(ethanol_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD9
input: present(gluconate_outside) & present(N_outside) & present(P_outside) & present(S_outside) & present(ADP) & present(P) & not present(O2_outside) & present(gluconate_import_enzymes) & present(fermentation_enzymes)
$\twoheadrightarrow_{4;40;4;4}$ output: present(acetate_outside) & present(ethanol_outside) & present(ATP) & present(nucleotides) & present(aminoacids).
MD10
input: present(O2_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output: present(ArcB_P).

The conjunction of these properties is denoted by DP(Metabolism).

4.3 Translation

Translation involves amino acids, ATP, and particular types of mRNA. It can produce particular enzymes, ADP, and P.

Input: present(aminoacids), present(ATP), present(respiration_mRNA),

present(fermentation_mRNA), present(glucose_import_mRNA),
present(lactose_import_mRNA), present(gluconate_import_mRNA)

Output: present(ADP) present(P), present(respiration_enzymes) present(fermentation_enzymes), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)

The following dynamic properties specify under which circumstances which particular output will be generated, i.e., the enzyme(s) is (are) produced for which the associated mRNA is present. The conjunction of all of these properties is denoted by DP(Translation).

TaD1
input: present(aminoacids) & present(ATP) & present(respiration_mRNA)
$\twoheadrightarrow_{0;0;60;600}$ output: present(ADP) & present(P) & present(respiration_enzymes)

TaD2
input: present(aminoacids) & present(ATP) & present(fermentation_mRNA)
$\twoheadrightarrow_{0;0;60;600}$ output: present(ADP) & present(P) & present(fermentation_enzymes)

TaD3
input: present(aminoacids) & present(ATP) & present(glucose_import_mRNA)
$\twoheadrightarrow_{0;0;60;600}$ output: present(ADP) & present(P) & present(glucose_import_enzymes)

TaD4
input: present(aminoacids) & present(ATP) & present(lactose_import_mRNA)
$\twoheadrightarrow_{0;0;60;600}$ output: present(ADP) & present(P) & present(lactose_import_enzymes)

TaD5
input: present(aminoacids) & present(ATP) & present(gluconate_import_mRNA)
$\twoheadrightarrow_{0;0;60;600}$ output: present(ADP) & present(P) & present(gluconate_import_enzymes)

4.4 Transcription

Transcription can use nucleotides, ATP, ArcB_P, allolactose, and gluconate6P observation amount; moreover for some functionality it depends on the presence of CRPcAMP. The gluconate6P observation amount refers to an amount of gluconate6P in the cell that signals the presence of gluconate in the environment. This amount is smaller than the amount of gluconate6P that would be present when gluconate is actively imported. The presence of appropriate DNA is not mentioned here as a condition, since it is assumed to be present internally, not as input. Depending on circumstances it can produce particular forms of mRNA, besides ADP and P.

Input: present(nucleotides), present(ATP), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount), present(ArcB_P)

Output: present(respiration_mRNA), present(fermentation_mRNA), present(glucose_import_mRNA), present(lactose_import_mRNA), present(gluconate_import_mRNA), present(ADP), present(P)

The following dynamic properties specify under which circumstances which output is generated; i.e., here the decisions are made which mRNA(s) will be generated for given circumstances. For example, always glucose import mRNA is generated (assuming ATP and nucleotides present), respiration and fermentation mRNA, are generated if ArcB_P is present or absent, respectively, and the production of lactose import mRNA and gluconate

each depend on specific other conditions (allolactose, CRPcAMP, resp. gluconate6P observation amount, CRPcAMP). The conjunction of all of these properties is denoted by DP(Transcription).

TcD1
input: present(ArcB_P) & present(nucleotides) & present(ATP)
$\rightarrow\!\!\!\!\rightarrow_{60;60;1;40}$ output: present(ADP) & present(P) & present(respiration_mRNA)
TcD2
input: not present(ArcB_P) & present(nucleotides) & present(ATP)
$\rightarrow\!\!\!\!\rightarrow_{60;60;1;40}$ output: present(ADP) & present(P) & present(fermentation_mRNA)

TcD3
input: present(nucleotides) & present(ATP)
$\rightarrow\!\!\!\!\rightarrow_{60;60;1;40}$ output: present(ADP) & present(P) & present(glucose_import_mRNA)
TcD4
input: present(allolactose) & present(CRPcAMP) & present(nucleotides) & present(ATP)
$\rightarrow\!\!\!\!\rightarrow_{60;60;1;40}$ output: present(ADP) & present(P) & present(lactose_import_mRNA)
TcD5
input: present(gluconate6P_observation_amount) & present(CRPcAMP) &
present(nucleotides) & input:present(ATP)
$\rightarrow\!\!\!\!\rightarrow_{60;60;1;40}$ output: present(ADP) & output:present(P) & present(gluconate_import_mRNA)

4.5 Transfer properties and interlevel interaction properties

In addition to the role properties, also transfer properties and interlevel role interaction properties have been specified. Both have zero time delay. This is achieved by putting the 'leads to' parameters e, f, g, h according to e = f = -g , and g=h, so that the result will occur simultaneously with the antecedent, and also will have the same length. This is instantaneous transfer, without possibility of loss of communication and without delay. The example model uses the settings –.1,-.1,.1,.1. As the roles take place at the same place, it is reasonable to assume that a generated output is immediately available as input. As an example, the following is a property template for transfer between Transcription and Translation, where p is any state property that belongs both to the output of transcription and the input of translation.

output(transcription): p $\rightarrow\!\!\!\!\rightarrow_{-.1;-.1;.1;.1}$ input(translation): p

This dynamic property relates output of the role Transcription to input of the role Translation at the same aggregation level in an instantaneous manner, i.e., without any time difference. The other transfer properties are similar. The conjunction of all dynamic properties for transfer between children roles of the (parent) Cell role is denoted by TRD(Cell).

An example of a property template for interlevel interaction between Metabolism and Cell is the following (here p belongs to the output of both metabolism and cell):

output(metabolism): p $\rightarrow\!\!\!\!\rightarrow_{-.1;-.1;.1;.1}$ output(cell): p

This dynamic property relates output of a role at a lower level to output of its parent role one aggregation level higher in an instantaneous manner, i.e., without any time difference. Similarly inputs of lower level roles can be related to input of their parent role. The

conjunction of all such dynamic properties for interlevel interaction between children roles and the (parent) Cell role is denoted by IID(Cell). Here, the abbreviation IID stands for the Interlevel Interaction Dynamics.

4.6 Logical interlevel relations between the dynamic properties within Cell

Within a compositional organisation model the dynamic properties of a higher level role are related to the dynamic properties of its child roles. Recall the general pattern from Section 3:

dynamic properties for child roles &
dynamic properties for transfer between child roles &
dynamic properties for interlevel interaction between parent role and children roles
$\Rightarrow$ dynamic properties for parent role

For the Cell as parent role, and Transcription, Translation and Metabolism as children roles this can be made more specific in the following manner:

DP(Transcription) & DP(Translation) & DP(Metabolism) &
TRD(Cell) & IID(Cell) $\Rightarrow$ DP(Cell)

In the above relation, DP stands for Dynamic Properties, TRD stands for the TRansfer Dynamics and IID stands for the Interlevel Interaction Dynamics.

This relation indeed holds. However, it can be made more specific by not involving whole sets of dynamic properties but specific subsets. For example, if the dynamic property CD1 for the cell is considered, this is already entailed by a smaller set of properties, i.e., a subset of the left hand side DP(Transcription) & DP(Translation) & DP(Metabolism) & TRD(Cell) & IID(Cell). Careful investigation of these more specific logical relationships yielded the following ones.

MD0 & MD4 & MD10 &
TcD1 & TcD3 &
TaD1 & TaD3 &
TRD(Cell) & IID(Cell) $\Rightarrow$ CD1

MD0 & MD1 & MD2& MD5 & MD10 &
TcD1 & TcD4 &
TaD1 & TaD4 &
TRD(Cell) & IID(Cell) $\Rightarrow$ CD2

MD0 & MD1 & MD3 & MD6 & MD10 &
TcD1 & TcD5 &
TaD1 & TaD5 &
TRD(Cell) & IID(Cell) $\Rightarrow$ CD3

MD0 & MD7 &
TcD2 & TcD3 &
TaD2 & TaD3 &
TRD(Cell) & IID(Cell) $\Rightarrow$ CD4

MD0 & MD1 & MD2 & MD8 &
TcD2 & TcD4 &
TaD2 & TaD4 &
TRD(Cell) & IID(Cell) ⇒ CD5

MD0 & MD1 & MD3 & MD9 &
TcD2 & TcD5 &
TaD2 & TaD5 &
TRD(Cell) & IID(Cell) ⇒ CD6

Note that for all of the properties of the Cell, the assumption must be met that at the start ATP, nucleotides and amino acids are available for the transcription and translation. This means that the cell should be alive at the start. This property is an initial condition called MD0, and was not covered in the description of the Metabolism earlier, because it has a different status; it is not in the general 'leads to' format. The property is:

MD0

[0:60] output: present(ATP) & present(nucleotides) & present(aminoacids)

The property MD0 is for the initialisation of the model. Because of the dynamics of the cell the substances mentioned in MD0 will be present if present earlier. This indicates that the cell stays prepared to regulate itself and to adapt to changes in its environment.

5 Dynamic Properties at Lower Aggregation Levels

From the three main roles within the cell, further refinement of Transcription and Translation will be addressed in Sections 5.1, 5.2 and 5.3. The main roles within Metabolism (Catabolism, Anabolism, and Transport) will be addressed in Sections 6, 7, and 8, respectively.

5.1 Translation and its two main roles

An overview is given of the input, the output and the dynamic properties of the roles within Translation. Note that the delay, here specified as between 0 and 0, only starts after the duration of the antecedent, so only after 60 seconds have passed will the enzymes start to be present.

Translation – Respiration Control
Input: present(aminoacids), present(ATP), present(respiration_mRNA), present(fermentation_mRNA)
Output: present(ADP), present(P), present(respiration_enzymes) present(fermentation_enzymes)

TaRD1
input:present(respiration_mRNA) & present(aminoacids) & present(ATP)
→→0;0;60;600 output:present(ADP) & present(P) & present(respiration_enzymes)
TaRD2
input:present(fermentation_mRNA) & present(aminoacids) & present(ATP)
→→0;0;60;600 output:present(ADP) & present(P) & present(fermentation_enzymes)

The conjunction of these properties is denoted by DP(Translation-Respiration-Control).

Translation – Nutrition import control

Input: present(aminoacids), present(ATP), present(glucose_import_mRNA), present(lactose_import_mRNA), present(gluconate_import_mRNA)

Output: present(ADP), present(P), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)

TaND1
input:present(glucose_import_mRNA) & present(aminoacids) & input:present(ATP)
→↠0;0;60;600 output:present(ADP) & present(P) & present(glucose_import_enzymes)

TaND2
input:present(lactose_import_mRNA) & present(aminoacids) & present(ATP)
→↠0;0;60;600 output:present(ADP) & present(P) & present(lactose_import_enzymes)

TaND3
input:present(gluconate_import_mRNA) & present(aminoacids) & present(ATP)
→↠0;0;60;600 output:present(ADP) & present(P) & present(gluconate_import_enzymes)

The conjunction of these properties is denoted by DP(Translation-Nutrition-Import-Control).

Logical relationships within Translation

Because

DP(Translation) = DP(Translation-Respiration-Control) & DP(Translation-Nutrition-Import-Control),

it trivially holds that

DP(Translation-Respiration-Control)
& DP(Translation-Nutrition-Import-Control)
& TRD(Translation) & IID(Translation) ⇒ DP(Translation)

5.2 Transcription and its two main roles

An overview is given of the input and output and the dynamic properties of the roles within Transcription.

Transcription – Respiration Control

Input: present(nucleotides), present(ATP), present(ArcB_P)

Output: present(ADP), present(P), present(respiration_mRNA), present(fermentation_mRNA)

TcRD1
input:present(ArcB_P) & present(nucleotides) & present(ATP)
→↠60;60;1;40 output:present(ADP) & present(P) & present(respiration_mRNA)

TcRD2
input: not present(ArcB_P) & present(nucleotides) & present(ATP)
→↠60;60;1;40 output:present(ADP) & present(P) & present(fermentation_mRNA)

The conjunction of these properties is denoted by DP(Transcription-Respiration-Control).

Transcription – Nutrition import control

Input: present(nucleotides), present(ATP), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount)

Output: present(ADP), present(P), present(glucose_import_mRNA), present(lactose_import_mRNA), present(gluconate_import_mRNA)

TcND1

input:present(nucleotides) & present(ATP)
→↠60;60;1;40 output:present(ADP) & present(P) & present(glucose_import_mRNA)

TcND2

input:present(allolactose) & present(CRPcAMP) & present(nucleotides) & present(ATP)
→↠60;60;1;40 output:present(ADP) & present(P) & present(lactose_import_mRNA)

TcND3

input:present(gluconate6P_observation_amount) & present(CRPcAMP) & present(nucleotides) & input:present(ATP)
→↠60;60;1;40 output:present(ADP) & output:present(P) & present(gluconate_import_mRNA)

The conjunction of these properties is denoted by DP(Transcription-Nutrition-Import-Control).

Logical relationships within Transcription

Because,

DP(Transcription) = DP(Transcription-Respiration-Control) & DP(Transcription-Nutrition-Import-Control),

it trivially holds

DP(Transcription-Respiration-Control)
& DP(Transcription-Nutrition-Import-Control)
& TRD(Transcription) & IID(Transcription) ⇒ DP(Transcription)

5.3 Metabolism and its three main roles

An overview is given of the input and output and the dynamic properties of the roles within Metabolism.

Catabolism

Input: present(glucose6P), present(gluconate6P), present(lactose), present(pyruvate), present(ADP), present(P), present(fermentation_enzymes), present(respiration_enzymes), present(O2), present(NAD(P))

Output: present(pyruvate), present(glucose6P), present(PEP), present(ATP), present(CO2), present(acetate), present(ethanol), present(NAD(P)H).

To keep the conserved moiety of NAD(P) and NAD(P)H in continued existance, the NAD(P) is kept present always. Thus the energy-poor version is kept available, while the energy-rich NAD(P)H can fluctuate.

Persistent: input:present(NAD(P)).

CaD1

(input:present(glucose6P) or present(gluconate6P) or present(lactose)) & present(ADP) & present(P) & present(NAD(P)) & present(O2) & present(respiration_enzymes)
$\twoheadrightarrow_{4;12;4;4}$ output:present(pyruvate) & present(glucose6P) & present(ATP) & present(NAD(P)H) & present(PEP) & present(CO2)

CaD2

(input:present(glucose6P) or present(gluconate6P) or present(lactose)) & present(ADP) & present(P) & present(NAD(P)) & present(fermentation_enzymes)
$\twoheadrightarrow_{4;12;4;4}$ output: present(pyruvate) & present(glucose6P) & present(ATP) & present(NAD(P)H) & present(PEP) & present(acetate) & present(ethanol)

Anabolism

Input: present(ATP), present(NAD(P)H), present(glucose6P), present(pyruvate), present(N), present(P), present(S)
Output: present(ADP), present(P), present(NAD(P)), present(aminoacids), present(nucleotides)

AD1

input:present(ATP) & present(NAD(P)H) & present(glucose6P) & present(pyruvate) & present(N) & present(P) & present(S)
$\twoheadrightarrow_{2;6;4;4}$ output:present(ADP) & present(P) & present(NAD(P)) & present(nucleotides) & present(aminoacids)

Transport

Input: present(glucose_outside), present(lactose_outside), present(gluconate_outside), present(O2_outside), present(N_outside), present(P_outside), present(S_outside), present(PEP), present(ATP), present(acetate), present(ethanol), present(CO2), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)
Output: present(glucose6P), present(lactose), present(gluconate6P), present(O2), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount), present(ArcB_P), present(N), present(P), present(S), present(pyruvate), present(ADP), present(P), present(acetate_outside), present(ethanol_outside), present(CO2_outside)

TrD1

input:present(glucose_outside) & present(PEP) & present(glucose_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(glucose6P) & present(pyruvate)

TrD2

input:present(gluconate_outside) & present(ATP) & present(gluconate_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(gluconate6P) & present(ADP) & present(P)

TrD3

input:present(lactose_outside) & present(ATP) & present(lactose_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(lactose) & present(ADP) & present(P)

TrD4

input:present(O2_outside) $\bullet\!\!\rightarrow\!\!\twoheadrightarrow_{0;0;4;4}$ output:present(O2) & present(ArcB_P)

TrD5

input:present(N_outside) & present(ATP)
$\twoheadrightarrow_{0;0;4;4}$ output:present(N) & present(ADP) & present(P)

TrD6

input:present(P_outside) & present(ATP)

$\twoheadrightarrow_{0;0;4;4}$ output:present(P) & present(ADP) & present(P)

TrD7

input:present(S_outside) & present(ATP)
$\twoheadrightarrow_{0;0;4;4}$ output:present(S & present(ADP) & present(P)

TrD8

input:present(acetate) $\twoheadrightarrow_{0;0;4;4}$ output:present(acetate_outside)

TrD9

input:present(ethanol) $\twoheadrightarrow_{0;0;4;4}$ output:present(ethanol_outside)

TrD10

input:present(CO2) $\twoheadrightarrow_{0;0;4;4}$ output:present(CO2_outside)

TrD11

input: not present(glucose_outside)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(CRPcAMP)

TrD12

input:present(lactose_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(allolactose)

TrD13

input:present(gluconate_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(gluconate6P_observation_amount)

Logical relationships within Metabolism

Also in this case the following logical relationship holds:

DP(Catabolism) & DP(Anabolism) & DP(Transport) &
TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ DP(Metabolism)

The more specific logical relationships are as follows.

TrD11 & TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD1

TrD12 & TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD2

TrD13 & TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD3

TrD1 & TrD4 & TrD5 & TrD6 & TrD7 & TrD10 &
CaD0 & CaD1 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD4

TrD3 & TrD4 & TrD5 & TrD6 & TrD7 & TrD10 &
CaD0 & CaD1 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD5

TrD2 & TrD4 & TrD5 & TrD6 & TrD7 & TrD10 &
CaD0 & CaD1 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD6

TrD1 & TrD5 & TrD6 & TrD7 & TrD8 & TrD9 &
CaD0 & CaD2 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) $\Rightarrow$ MD7

TrD3 & TrD5 & TrD6 & TrD7 & TrD8 & TrD9 &
CaD0 & CaD2 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) ⇒ MD8

TrD2 & TrD5 & TrD6 & TrD7 & TrD8 & TrD9 &
CaD0 & CaD2 &
AD0 & AD1 &
TRD(Metabolism) & IID(Metabolism) ⇒ MD9

TrD4 & TRD(Metabolism) & IID(Metabolism) ⇒ MD10

The initialisation properties here are AD0 and CaD0, given as:

AD0

[0:60] output: present(NAD(P))

CaD0

[0:60] output: present(PEP) & present(ATP)

These properties are used to initialise the simulation as well.

6 Simulation and Analysis Results

For a simulation based approach to be acceptable, the simulation has to fulfill the following criteria (Hollendbeck, 2000): replication, prediction, data availability, and validation. Therefore, the simulation has to be described in sufficient detail, along with the details of initial values. The parameters and variables of the simulation have been identified on the basis of the available data on the cell (and in particular ecoli) in literature. The aim of this last section is to show the validity of the model presented.

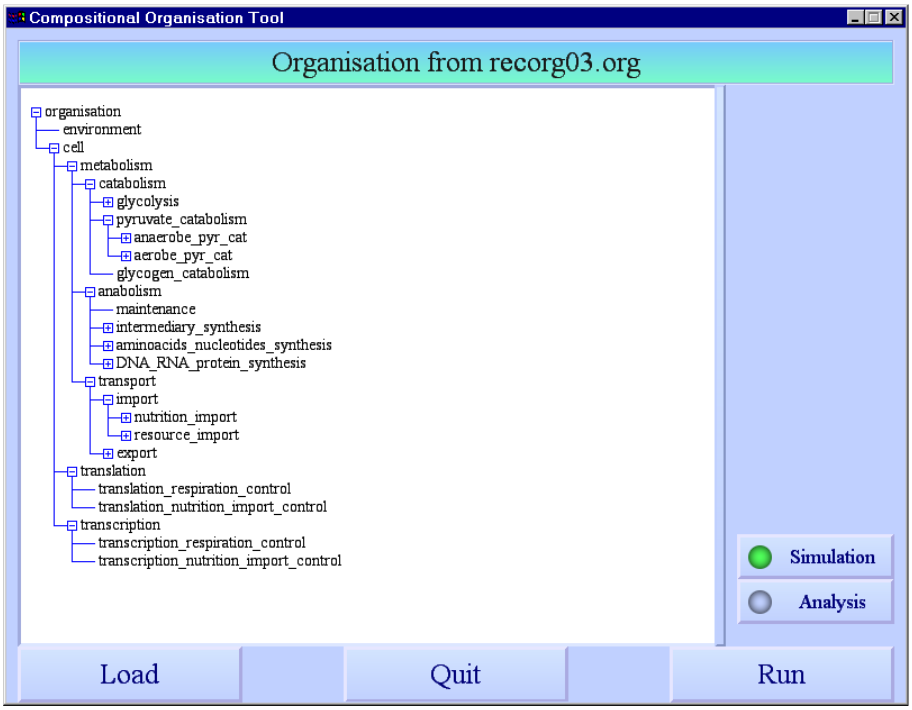


Figure 5. Screenshot of the compositional organisation modelling tool, selecting which level of aggregation to use for simulation or analysis.

A software environment has been built to support both simulation and analysis of organisation models. For a compositional organisation model this software allows the user to select which parts of the hierarchy need to be taken into account, and then to run a simulation or analysis, see Figure 5.

6.1 Simulation

Given the available dynamic properties at each aggregation level, a choice can be made to perform simulation at a high level of aggregation, or at a lower level. Simulation at a higher level of aggregation uses the rougher, less detailed dynamical properties of the higher level roles. Simulation at a lower level of aggregation uses the more detailed dynamic properties of the lower level roles. Because of information hiding, the higher level properties are less in number and less detailed, resulting in more efficient simulation due to the lower complexity. The level of aggregation for simulation can be selected per role. If a particular role is selected for detailed, lower level, simulation, one possibility would be to assume that the part around it is constant. Another possibility, used here, is to simulate the parts around it using them at a higher level of aggregation, using the less detailed properties. Thus it is possible to indicate the level of detail to use for simulation for each part of the tree separately (as depicted in Figure 4), by navigating the tree selecting some parts for more or less detail than other parts. Simulation is performed using the dynamic 'leads to' properties

associated to the leaves of the subtree selected: the end nodes (not indicated by -) in Figure 5. Roughly spoken, the simulation algorithm takes care that if the antecedent of a 'follows' property is true in the trace constructed so far, the trace will be extended by making the consequent true as well. If several rules fire to make the same consequent true, which means some substance has a concentration above a threshold, then the consequent will be true in the conjunction of both intervals. The modeller should make sure that no rules fire to make the same consequent true and false at the same time, as this is an inconsistency. By searching and applicable rules whose consequent has not been added to the trace yet, the trace is extended to further time intervals. The simulations are initialised by setting all inputs to true for a period of time, 0 .. 60 seconds in the examples. If needed, a subset of the input of a component can be specified per component; only that subset will be initiated in that case.

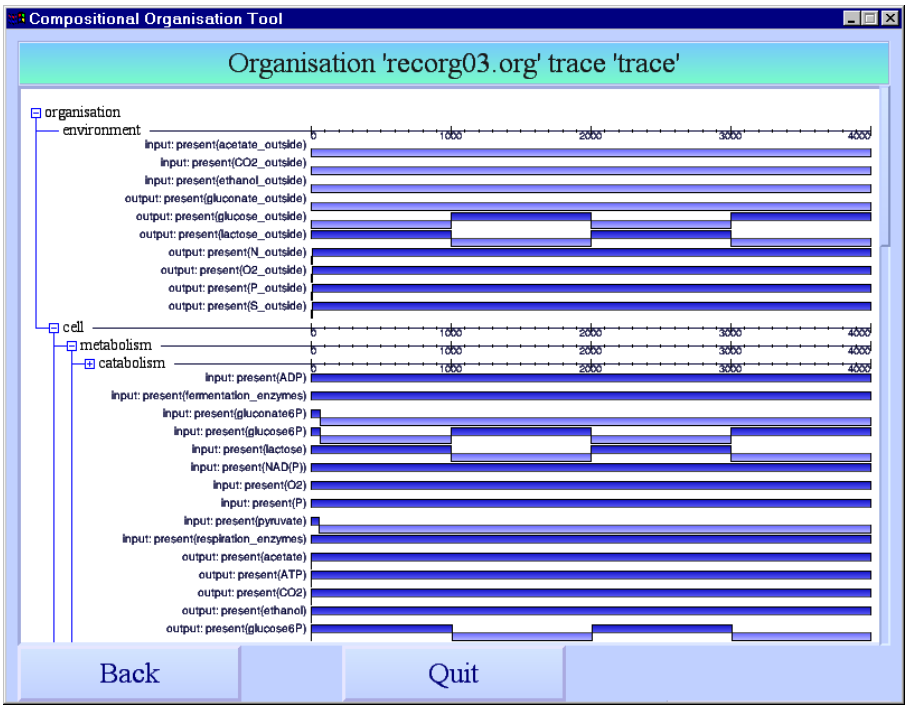


Figure 6. Screenshot of the compositional organisation tool, browsing the results of a simulation. Time flows to the right, in seconds. The roles environment en catabolism have their input and output visible. Dark boxes mean true, light boxes mean false. Lactose and glucose are alternating in the environment in a way that was predefined. All 'environment 'properties were completely defined by the outside world. All other properties obtained an initial value at the beginning but were further determined by the system. The lactose and glucose6P inside the cell are input to the catabolism.

After a simulation the user is presented with the results, where for each part (role with its subroles) the input, internal and output atoms are shown changing over time, see Figure

6. Atoms are the input and output identifiers for the roles. Time flows to the right. When an (atomic) state property is true this is depicted by a dark box above the line, whilst false is depicted by a lighter box below the line. As in the paradigm of Executable Temporal Logic (Barringer et al., 1996), the simulation algorithm achieves that the constructed trace satisfies all the 'follows' properties used for the simulation (assuming that the model does not include inconsistencies). This does not automatically imply that all dynamic properties at higher levels of aggregation will hold for such a constructed trace. This is because during model construction the lower level properties used in simulation might not imply the higher level properties, and analysis would reveal this (validation of the model). However, if the logical relationships between dynamic properties at different levels of aggregation hold as discussed, the higher level dynamic properties will (have to) hold. This is useful for validation of the dynamic properties: For each trace obtained by simulation, it can be checked automatically whether any 'leads to' property holds. If a higher level property does not hold for a given trace, then this indicates incorrectness of some of the lower level properties. This analysis process will be discussed in the next section.

The simulation of Fig. 6, took less than 1.8 hours of computation time, for 533 identifiers and 828 temporal relationships (not all shown in the Figure). In this simulation several changes (imposed by us) on the environment lead to cell dynamics where one steady state after another was encountered, and also states on the way from one steady state to another are derived. Of the 59 components in the actual simulation, for reasons of presentation, only some are shown.

6.2 Analysis

The simulations for two slightly different models can be compared. This can be used to localize the differences between the two models. Different models can be obtained by making different choices for the levels of aggregation to use for simulation or analysis. Note that an analysis model contains not only the 'leads to' properties of the leaves of the tree, but also the dynamical properties of higher level roles. Our analysis software focuses on this option. Another option would be to obtain a trace from a laboratory experiment and check this with respect to the dynamic properties of the organisation model. Whether the trace is obtained by another model or by experiment, the differences between a trace and the analysis model are analysed.

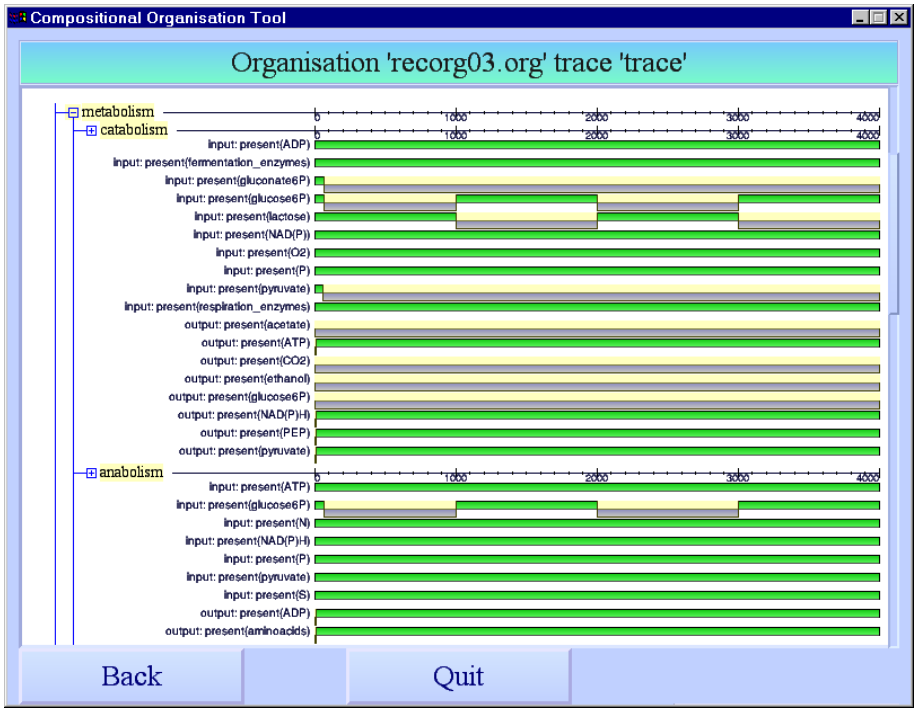


Figure 7. Screenshot showing the results of an analysis of a trace obtained by simulation.

The Analysis Software works by checking for each role whether its action and premissae are related correctly in the given trace with respect to the simulation model. The analysis software changes the colours where the trace differs from the prediction of the model. In the trace the blue colour is changed to green in undisputed spots, whilst deviations are flagged yellow (unexpected by the analysis model) or red (expectation unfulfilled for the analysis model). A light yellow colour means that a false interval was not explained, it was not expected. A stronger yellow colour means that a non-false interval was not expected. Yellow means that the 'leads to' properties that are checked do not completely specify the trace. A red colour means that an interval was expected by a 'leads to' property, but did not happen in reality. If this occurs, the 'leads to' property that is checked and the 'leads to' properties used to specify the behaviour do not fit together, since the former predicts something different from what happens in the trace. The analysis results should be interpreted according to circumstances. If the trace has been produced by a simulation, then either the properties used for simulation are wrong or the property checked is wrong. If a "natural", laboratory, trace is analysed, then the property checked has to be wrong. To aid browsing the results, the name of each component is highlighted in yellow or red if any errors have been found in that role's input or output. Also the + and - box next to a component are highlighted if there are any errors in the subtree below a component.

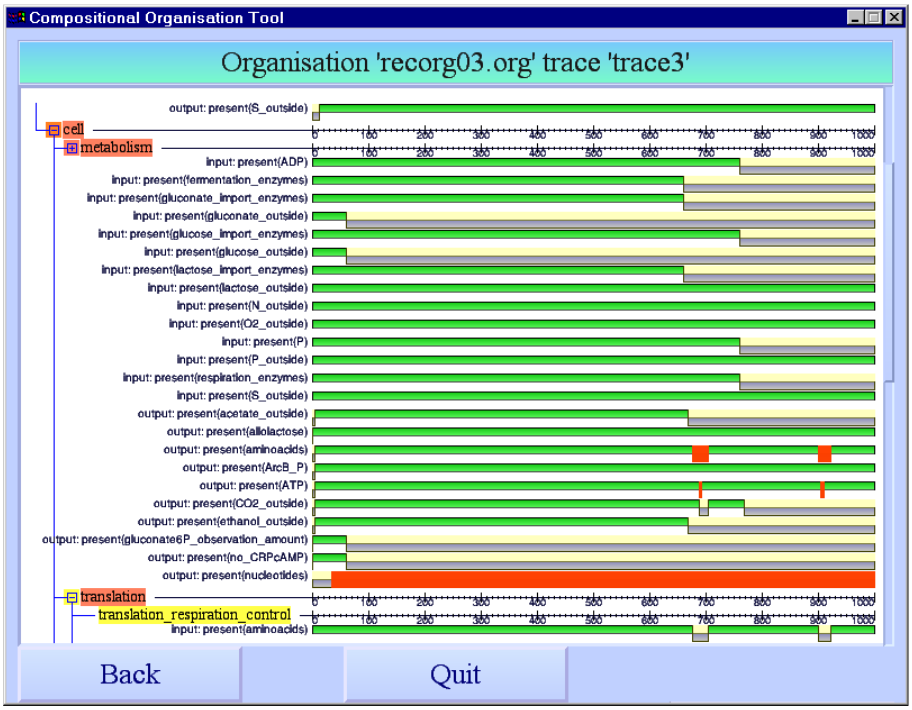


Figure 8. Results of analysis, viewing the metabolism

In Figs 8-11 an example top-down analysis of a trace is shown. In the complete model the component nucleotides synthesis was crippled, by having it no longer generate any nucleotides. We shall now show how the analysis mode of our Software facilitated finding where the bug was. A simulation was run both of the crippled and of the uncrippled model. Then both were submitted to the 'Analysis' routine. Then a check was done at the top level. This check was inspected, see Figure 8. As can be seen many errors occurred because of the crippling. One clearly marked is that the metabolism properties expected nucleotides on the output, but they never appeared, amongst others, later on in the trace.

In Figure 9, the metabolism is further inspected to reveal that the anabolism also expected nucleotides, but they are missing.

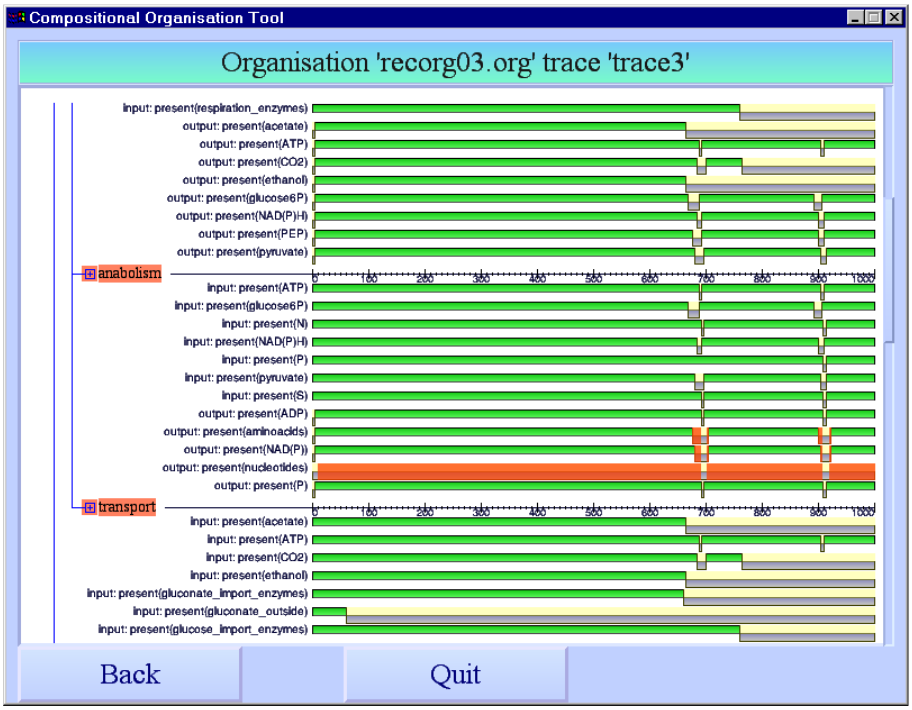


Figure 9. Results of analysis, viewing the anabolism

In Figure 10 the aminoacids nucleotides synthesis component is inspected, showing that nucleotides were again missing.

In Figure 11, the nucleotides synthesis component is shown. It reported no serious errors – since it operated perfectly fine according to the faulty property. Since the nucleotides synthesis role property was sabotaged, and simulation produced a trace conforming to it, during the analysis the property will not show discrepancies with the trace. There are no nucleotides on the output however. This causes the properties of higher level components to show discrepancies with the trace, clearly flagging where to investigate. In this example it can be seen how the properties of the higher level components aid in finding bugs in the specification of the model.

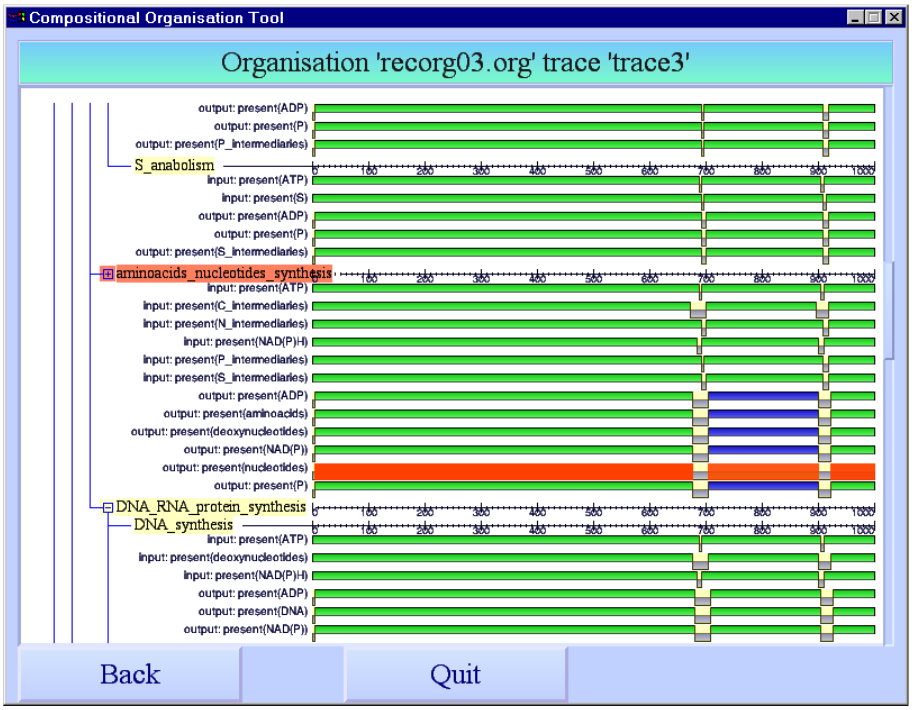


Figure 10. Results of analysis, viewing the aminoacids nucleotides synthesis

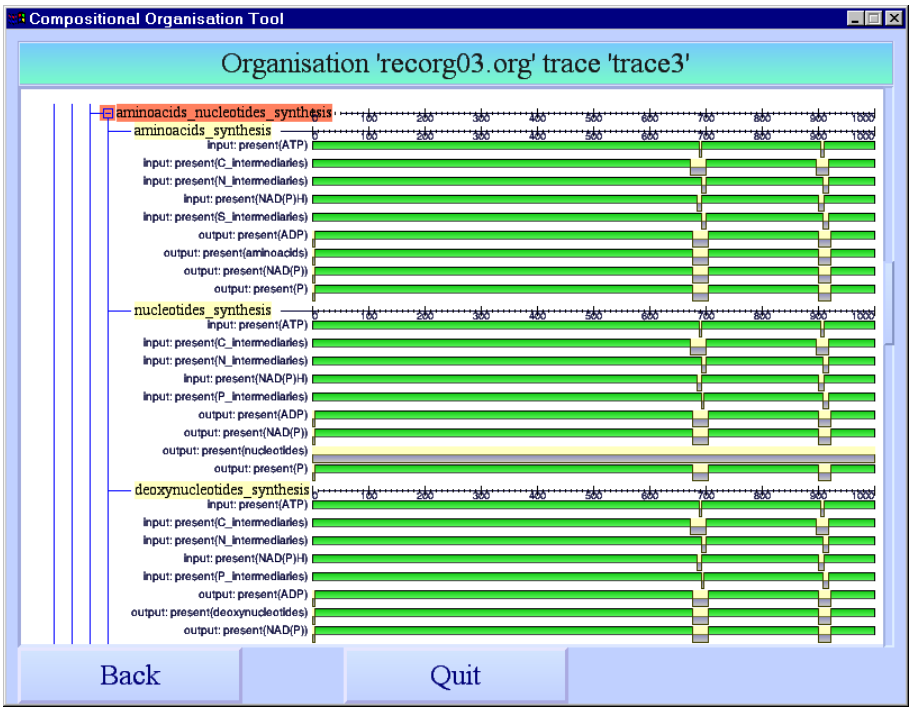


Figure 11. Results of analysis, viewing the nucleotides synthesis

7 Discussion

In this paper it was shown how organisational modelling techniques could be exploited to manage complexity of intracellular processes. The organisational modelling techniques were adopted from the area of Computational Organisation Theory and Artificial Intelligence, where they are used to describe how complex dynamics in human societies can be managed using multi-agent organisation structures. Usually these organisation modelling techniques are applied to organisations such as factories and the internal organisation of their process flows, obtaining high-level models of complex organisations at different levels of aggregation. From the biological perspective complex intracellular dynamics are often interpreted in terms of well-organised processes; sometimes a cell is considered a (micro)factory. Using the example of *Escherichia coli* it was shown how indeed such organisational modelling techniques can be used to simulate and analyse *E. coli*'s intracellular dynamics. Exploiting the abstraction levels entailed by this organisation modelling perspective, a concise model was obtained that is easy to simulate and analyse at different levels of aggregation.

The abstraction was based on a detailed mapping of the abstract notions to detailed information on the concentration levels of the various substrates involved in the processes

of the cell. That way we ensured the downward compatibility of our model to the traditional models of the cell, often presented in terms of differential equations. Upward compatibility is maintained, but of course only with respect to the abstractions made in our model. For example, the continuous nature of the chemical processes in the cell is reduced to more characteristic and abstract notions of the cell process.

The added value of this abstraction approach is that it provides teachers with the justification to explain the overall cell behaviour in such abstract terms. The mappings and the runtime behaviour of the abstract model validate their practice. Their abstractions correspond to the abstraction by the 'leads to' properties based on discretization of states and have the benefits of understandability and efficient simulation. These benefits are enhanced by the introduction of overall properties, again based on biochemical knowledge, that span different levels of aggregation and simplify the analysis of the model. Higher-level properties enable the analysis of traces obtained using simulation of lower level properties and make it easier to find errors in the model. The simulations show the cell's essential dynamic patterns depending on (static or dynamic) environmental conditions, both for reaching a steady state (in case of a static or fast fluctuating environment) and for oscillating dynamics (e.g., in case of a periodically changing environment). As soon as the student is interested in more details of such dynamics, he can zoom in to that level by taking the more standard simulations of the indivual pathways that are based on differential equations.

Technically, it is possible to connect these more detailed models to our simulation in the LEADSTO environment. We have done so successfully for other domains. That way the higher level behaviour would emerge out of the behaviour at the lower levels of abstraction. In this case we chose not to do so, precisely to show that a model at the higher level of abstraction is enough to show the essential dynamic patterns of the cell.

The LEADSTO modelling approach (Bosse et al., 2007) used as a vehicle, has some elements in common with discrete event simulation methods. A difference is that LEADSTO belongs to the family of executable logical languages, and therefore concepts from logic can directly be applied to LEADSTO specifications.

In the method used, any simulation produces a *trace* that satisfies the dynamic properties in the 'leads to' language used for the simulation. As a consequence of the logical relationships between dynamic properties, the higher level properties are implied by the lower level properties. Thus, once these interlevel relationships have been validated, for any trace simulated by dynamic properties at a certain aggregation level, it is guaranteed that the higher-level properties hold. Such an approach can also be applied within the differential equations method if for higher levels of aggregation differential equations are used for lumped reactions. The use of different aggregation levels is a distinguishing element and advantage, as compared to the modelling of intracellular processes based on beliefs, desires and intentions (the socalled BDI-model) presented for the steady state and non-steady state case in (Jonker, Snoep, Treur, Westerhoff and Wijngaards, 2002, 2008). The BDI-modelling approach, however, has the advantage of interpretation of intracellular processes in readily understood intentional terms for internal states, whereas in the current paper internal states are described in biochemical terms that are accessible to biochemists only.

The modeling, or rather calculation (Westerhoff, 2001), of living systems is becoming more and more timely, with the vast amount of experimental data surmounting the possibility of evaluation by the unaided human mind. Detailed models have been the answer until now (e.g. Teusink et al., 2000, Wang et al., 2001; Takahashi, et al., 2003; Snoep, 2005; Jamshidi and Palsson, 2006), but they have been limited to smaller parts of the cell, in part because of the complexity of handling larger parts. Existing whole-cell models have been lacking the true kinetic information of the enzymes or have had to rely on oversimplified aspects thereof (Covert et al., 2001). Ways of simplifying this type of modeling without loss in essential information on the dynamic behavior have been sought after. This paper is not the first attempt at simplification by modeling living cells in terms of hierarchical and modular structures. Heinrich et al. (1976) championed an approach based on the time hierarchy of a system, treating fast relaxing and slowly relaxing subsystems differently. Kahn and Westerhoff (1991) developed a hierarchical approach for metabolic control analysis where they distinguished modules within a cell that do not communicate by material fluxes between them (see also Hofmeyr & Westerhoff, 2001). Westerhoff et al (1983), Schuster et al (1983), Brown et al. (1990) and Rohwer et al. (1996) developed modular approaches to flux-connected biochemical networks. In a way, these approaches mimicked what has been biochemical intuition for qualitative approaches to cell function. All these earlier approaches however missed the possibility to discuss the organization of cell function in much the same way as one discusses the organization of human society.

The high-level of abstraction of the organizational model results in an inherently faster computation compared to a model based on differential equations. This computational advantage mainly stems from the fact that to model dynamic relations between states in a trajectory or trace, in the presented model 'leads to' properties are used instead of differential equations that have to be computed in very small steps. Using 'leads to' properties also flat, unorganised models can be built, but the organisational structure can be used to obtain even more efficiency. An additional computational advantage can be obtained if simulation is performed on the basis of 'leads to' properties at higher aggregation levels, abstracting from processes at lower aggregation levels. However, in that case no information is obtained on the dynamics of these processes at lower levels. It depends on the interest of the simulator whether or not this is a drawback. If only the global dynamics is of interest, a simulation at a higher aggregation level may suffice. We have not engaged in this stategy in the present manuscript but may do so in the future.

There are perhaps a few situations where the present method does differ critically in its quantitative predictions from the method of differential equations. For instance, when delay times are comparable to relaxation times of concentrations, the present method might become less accurate and even produce apparent oscillations where there are none in reality. The oscillations then derive from negative feedback loops over long delays that stem from the modelling method rather than that they are inherent in the system that is being modelled. A second aspect where the method used here, as well as the methods developed by Glass & Kauffman (1973), Jonker et al. (2001) and Jonker & Treur (2002) may be inferior to the method of integrating differential equations, is the tendency to treat concentrations digitally in a binary system. Indeed binary modeling methods such as proposed by Glass and Kauffman (1973) may not be subtle enough for some aspects of cell function (cf. Endy & Brent, 2001). In reality concentrations can assume a wide range of

values and reaction rate are quite sensitive to the concentrations. Metabolic control is in fact based on such subtlety (Hofmeyr, 2002). On the other hand, it does take a while before an entire mRNA or protein molecule has been synthesized and this discrete time effect is rarely simulated in differential equation models, whereas it comes naturally in the method developed here. Some other assumptions made are that the model presented in the paper does not model reactions as consuming substrates. Instead, products have a minimum lifetime, which can be extended but will eventually expire. The approach allows the addition of substrate consumption by specifying this in the consequents of the 'leads to' properties.

With respect to the specific model of *E. coli* developed here, a number of *caveats* are appropriate. First, parts of the biochemistry of this bacterium are insufficiently known for the model to become completely detailed and precise. Paradoxically, of some parts of *E. coli* biochemistry much more is known and in much more detail than what has been implemented here. With respect to those parts however, we have engaged in (over) simplification because of the unavoidable lack of precision in the other parts should not make enhanced precision worthwhile. After all, the present study merely served to illustrate the method of organizational modeling for living cells. In order to obtain an optimal model of *E. coli* a separate study will be needed.

An amelioration of the model could be achieved by using more valuations for substances than present or not present as done now. For example, a worthwhile addition could be the use of a scale of three values: low, medium and high for ATP and NAD(P)H. The approach allows the use an arbitrary number of levels per substance.

Within the model, some state properties (atoms) were assumed persistent. For example, in catabolism input(NAD(P)), and also one in glucose_import2. These state properties (on input, output or internal) will always hold since they start to hold. They are used here to keep the conserved moieties present at all times. The energy-drained versions were kept in existence, whilst the energy-rich versions could fluctuate and possibly disappear. The NAD(P) and IIAglc substances are assumed persistent in the model. The notation NAD(P)H and NAD(P) was used to abstract from NADH, NADPH, NAD and NADP. NADH and NADPH were taken together to avoid the added complexity of transhydrogenase reactions and pathways.

Notwithstanding these *caveats*, the reader may have noted remarkable similarities between the performance of the model and what is known concerning the physiology of *E. coli*. Here and in parallel studies by the same authors, the well known strong regulation of cell physiology by oxygen and glucose in this organism was reproduced. This suggests that organizational modelling may well help to grasp the essence of the physiology of living cells.

Interesting further work would be to investigate further the relationship between the approach in this paper and methods based on differential equations. In this paper logical relations between dynamical properties of components and subcomponents are established. Perhaps something similar can be done for differential equations instead of logical formula to specify dynamic properties.

References

Barringer, H., M. Fisher, D. Gabbay, R. Owens, and M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd. and John Wiley & Sons.

Ben-Jacob, E., Cohen, I., Czirók, A., Vicsek, T., and Gutnick, D.L. (1997). Chemomodulation of Cellular Movement and Collective Formation of Vortices by Swarming Bacteria. In: *Physica A*, Vol. 238, pp. 181-197.

Booch, G. (1991). *Object oriented design with applications*. Benjamins Cummins Publishing Company, Redwood City.

Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., (2007). A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.

Brazier, F.M.T., Jonker, C.M., and Treur, J., (2002). Principles of Component-Based Design of Intelligent Agents. *Data and Knowledge Engineering*, vol. 41, 2002, pp. 1-28.

Brown GC, Hafner RP, and Brand MD. (1990). A 'top-down' approach to the determination of control coefficients in metabolic control theory. *Eur J Biochem*. 188:321-5

Covert, M.W., Schilling, C.H., Famili, I., Edwards, J.S., Goryanin, I.I., Selkov, E., Palsson, B.O. (2001). Metabolic modeling of microbial strains in silico. *Trends Biochem Sci*. 2001 Mar;26(3):179-86.

Dardenne, A., Lamsweerde, A. van, and Fickas, S. (1993). Goal-directed Requirements Acquisition. *Science in Computer Programming*, vol. 20, pp. 3-50.

Dubois, E., Du Bois, P., and Zeippen, J.M. (1995). A Formal Requirements Engineering Method for Real-Time, Concurrent, and Distributed Systems. In: *Proceedings of the Real-Time Systems Conference, RTS'95*.

Endy D & Brent R. (2001) Modelling cellular behavior. *Nature* 409, 391- 395.

Ferber, J., Gutknecht, O., Jonker, C.M., Müller, J.P., and Treur, J., (2001). Organization Models and Behavioural Requirements Specification for Multi-Agent Systems. In: Y. Demazeau, F. Garijo (eds.), *Multi-Agent System Organisations*. *Proceedings of the 10th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'01*, 2001.

Ferber, J., and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. *Third International Conference on Multi-Agent Systems (ICMAS '98) Proceedings.* IEEE Computer Society, 1998.

Fisher, M., (1994). A survey of Concurrent METATEM — the language and its applications. In: D.M. Gabbay, H.J. Ohlbach (eds.), *Temporal Logic — Proceedings of the First International Conference*, Lecture Notes in AI, vol. 827, pp. 480–505.

Fisher, M., (2005). Temporal Development Methods for Agent-Based Systems, *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 10, 2005, pp. 41-66.

Galton, A. (2003). Temporal Logic. *Stanford Encyclopedia of Philosophy*, URL: http://plato.stanford.edu/entries/logic-temporal/#2.

Gear, C.W., (1971). Numerical Initial Value Problems in Ordinary Differential Equations. Englewood Cliffs, NJ: Prentice-Hall.

Glass L and Kauffman SA. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J Theor Biol*.39:103-29.

Heinrich R., Rapoport S.M., and Rapoport, T.A. (1977). Metabolic Regulaiton and Mathematical Models. *Progr. Biophys. Mol. Biol.* 32, 1-82.

Hofmeyr, J.H.S. (2002) Metabolic Control Analysis in a nutshell. *BioComplexity*, in the press.

Hofmeyr J.H.S. and Westerhoff H.V. (2001). Building the cellular puzzle: Control in Multi-level reaction networks. *J. Theor. Biol*. 208, 261-285.

Jamshidi, N., and Palsson, B.O. (2006) Systems biology of the human red blood cell. *Blood Cells, Molecules and Diseases* 36:239-247.

Jonker, C.M., Letia, I.A., and Treur, J., (2002). Diagnosis of the Dynamics within an Organisation by Trace Checking of Behavioural Requirements. In: Wooldridge, M., Weiss, G., and Ciancarini, P. (eds.), *Proc. of the 2nd International Workshop on Agent-Oriented Software Engineering, AOSE'01*. Lecture Notes in Computer Science, vol. 2222. Springer Verlag, 2002, pp. 17-32.

Jonker, C.M., Treur, J., and Wijngaards, W.C.A., (2001). Specification, Analysis and Simulation of the Dynamics within an Organisation. In: *Journal of Applied Intelligence*, vol. 27, 2007, pp. 131-152.

Jonker, C.M., Snoep, J.L., Treur, J., Westerhoff, H.V., and Wijngaards, W.C.A. (2002). Putting Intentions into Cell Biochemistry: An Artificial Intelligence Perspective. *Journal of Theoretical Biology,* vol. 214, pp. 105-134.

Jonker, C.M., Snoep, J.L., Treur, J., Westerhoff, H.V., and Wijngaards, W.C.A. (2008). BDI-Modelling of Complex Intracellular Dynamics. *Journal of Theoretical Biology*, vol. 251, 2008, pp. 1–23.

Jonker, C.M. and Treur, J. (2002). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems*, vol. 11, 2002, pp. 51-92.

Kahn D., and Westerhoff H.V. (1991) Control Theory of Regulatory Cascades. *J. Theor. Biol.* 153, 255 - 285.

Kholodenko, B.N., Demin, O.V., Moehren, G., Hoek, J.B. (1999). Quantification of short term signaling by the epidermal growth factor receptor. *J. Biol Chem*., 274(42):30169-81.

Knuth, D.E. (1981). *The Art of Computer Programming*, Addison-Wesley.

Lomi, A., and Larsen, E.R. (2001). *Dynamics of Organizations: Computational Modeling and Organization Theories*, AAAI Press, Menlo Park.

Mintzberg, H. (1979). *The Structuring of Organisations*, Prentice Hall, Englewood Cliffs, N.J.

Moss, S., Gaylard, H., Wallis, S. & Edmonds, B. (1998). SDML: A Multi-Agent Language for Organizational Modelling, *Computational and Mathematical Organization Theory*, vol. 4, (1), 43-70.

Pajares, J., Lopez, A, and Hernandez, C., 2003. Industry as an Organisation of Agents: Innovation and R&D Management. In: Journal of Artificial Societies and Social Simulation, vol. 6, no. 2, http://jasss.soc.surrey.ac.uk/6/2/7.html.

Press, W. H., Teukolsky, S. A., Vetterling,W. T., and Flannery, B. P., 1992. *Numerical recipes in C: the art of scientific computing*. Cambridge university press, second edition.

Prietula, M., Gasser, L., Carley, K. (1997). *Simulating Organizations*. MIT Press.

Rizzi, M., Baltes, M., Theobald, U. & Reuss, M. (1997). *In vivo* analysis of metabolic dynamics in *Saccharomyces cerevisiae*: II. Mathematical model. *Biotechnol. Bioeng.* **55**, 592–608.

Rohwer J.M., Schuster S. and Westerhoff H.V. (1996). How to recognize monofunctional units in a metabolic system. *J. theor. Biol.* 179, 213-228.

Rohwer, J.M., Meadow, N.D., Roseman, S., Westerhoff, H.V., Postma, P.W. (2000). Understanding glucose transport by the bacterial phosphoenolpyruvate:glycose

phosphotransferase system on the basis of kinetic measurements in vitro. *J. Biol. Chem*., 275(45):34909-21.

Rotterdam, B.J. van, Crielaard, W., van Stokkum, I.H., Hellingwerf, K.J., Westerhoff H.V. (2002). Simplicity in complexity: the photosynthetic reaction center performs as a simple 0.2 V battery. *FEBS Lett*., 510(1-2):105-7.

Snoep, J.L. (2005) The Silicon Cell initiative: working towards a detailed kinetic description at the cellular level. *Curr Opin Biotechnol*. 16:336-43.

Stuart, A.M., and Humphries, A.R. (1996). *Dynamical Systems and Numerical Analysis*. Cambrige University Press.

Takahashi, K., N. Ishikawa, Y. Sadamoto, H. Sasamoto, S. Ohta, A. Shiozawa, F. Miyoshi, Y. Naito, Y. Nakayama and M. Tomita (2003) E-Cell 2: Multi-platform E-Cell simulation system. *Bioinformatics* 19: 1727-1729

Tanenbaum, A.S. (1976). *Structured Computer Organisation*. Prentice-Hall, London.

Teusink, B., Passarge, J., Reijenga, C.A., Esgalhado, E., van der Weijden, C.C., Schepper, M., Walsh, M.C., Bakker, B.M., van Dam, K., Westerhoff, H.V., Snoep, J.L. (2000). Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *Eur J Biochem.,* 267(17):5313-29.

Wang, J., Gilles, E.D., Lengeler, J.W., and Jahreis, K. (2001). Modeling of inducer exclusion and catabolite repression based on a PTS-dependent sucrose and non-PTS-dependent glycerol transport systems in Escherichia Coli K-12 and its experimental verification. *J. Biotechnol.* 2001, Dec 28; 92(2): 133-58.

Weiss, G. (ed.) (1999). *Multiagent Systems*. MIT Press.

Westerhoff, H.V.(2001). The silicon cell, not dead but live! *Metab. Eng.* 2001; 3(3):207-10.

Wijker J.E., Jensen P.R., Vaz Gomes A., De Waal A., Van Workum M., Van Heeswijk W.C., Molenaar D., Wielinga P., Diderich J., Bakker B.M., Teusink B., Hemker M., Rohwer J., Van der Gugten A.A., Kholodenko B.N., and Westerhoff H.V. (1995). Energy, control and DNA structure in the living cell. *Biophys. Chem.* 55, 153-165.

Wooldridge, M., Jennings, N.R. (eds.) (1995). *Intelligent Agents,* Lecture Notes in Artificial Intelligence, Vol. 890, Springer Verlag, Berlin.

Appendix A *E. coli*'s Internal Dynamics: Catabolism

In this section the dynamics of the children roles of Catabolism are discussed in more detail: Glycolysis, Pyruvate-Catabolism, and glycogen catabolism. Descriptions of roles at lower levels of aggregation can be found in the Appendix.

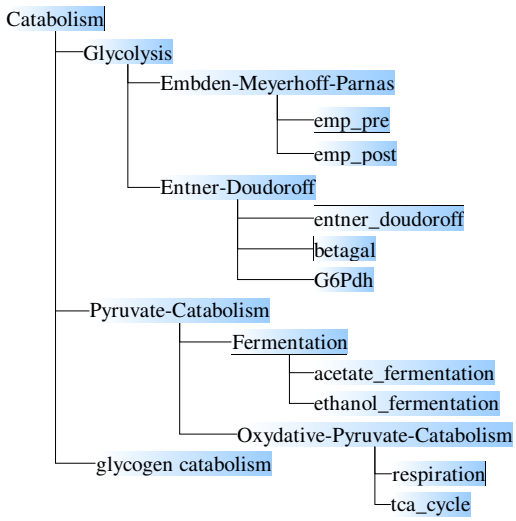


Figure 5. Catabolism Organisation structure.

6.1 Organisation structure for Catabolism

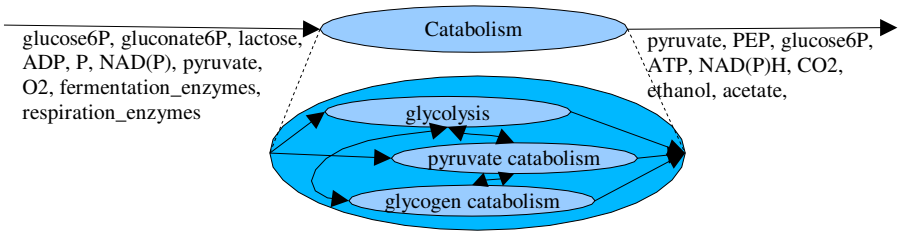


Figure 6. Catabolism organisation structure

6.2 Organisation dynamics within Catabolism

Catabolism

Input: present(glucose6P), present(gluconate6P), present(lactose), present(pyruvate), present(ADP), present(P), present(fermentation_enzymes), present(respiration_enzymes), present(O2), present(NAD(P))

Output: present(pyruvate), present(glucose6P), present(PEP), present(ATP), present(CO2), present(acetate), present(ethanol), present(NAD(P)H).

Persistent: input:present(NAD(P)).

CaD1

(input:present(glucose6P) or present(gluconate6P) or present(lactose)) & present(ADP) & present(P) & present(NAD(P)) & present(O2) & present(respiration_enzymes) $\twoheadrightarrow_{4;12;4;4}$ output:present(pyruvate) & present(glucose6P) & present(ATP) & present(NAD(P)H) & present(PEP) & present(CO2)

CaD2

(input:present(glucose6P) or present(gluconate6P) or present(lactose)) & present(ADP) & present(P) & present(NAD(P)) & present(fermentation_enzymes) $\twoheadrightarrow_{4;12;4;4}$ output:present(pyruvate) & present(glucose6P) & present(ATP) & present(NAD(P)H) & present(PEP) & present(acetate) & present(ethanol)

Glycogen catabolism

Input: present(ATP)
Output: present(glucose6P), present(ADP), present(P).

The glucose6P produced when glycogen is catabolised is further processed by the glycolysis and pyruvate catabolism to release ATP.

SD1

input: not present(ADP) & internal:present(glycogen) $\twoheadrightarrow_{0;0;1;200}$ output:present(glucose6P)

SD2

input:present(ATP) $\twoheadrightarrow_{0;0;200;200}$ internal:present(glycogen) & output:present(ADP) & present(P)

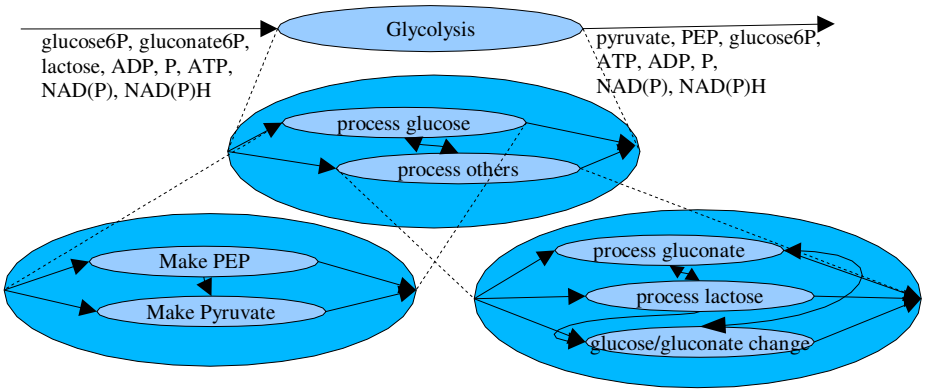


Figure 7. Glycolysis organisation structure. For a desciption of the lower components, see Appendix A

Glycolysis

Input: present(glucose6P), present(gluconate6P), present(lactose), present(ADP), present(P), present(ATP),
present(NAD(P)), present(NAD(P)H)

Output: present(glucose6P), present(pyruvate), present(PEP), present(ATP), present(ADP), present(P), present(NAD(P)),
present(NAD(P)H)

GD1

(input:present(glucose6P) or present(gluconate6P) or present(lactose)) & present(ADP) & present(P) & present(NAD(P))
→→2;6;4;4 output:present(pyruvate) & present(glucose6P) & present(PEP) & present(ATP) & present(NAD(P)H)

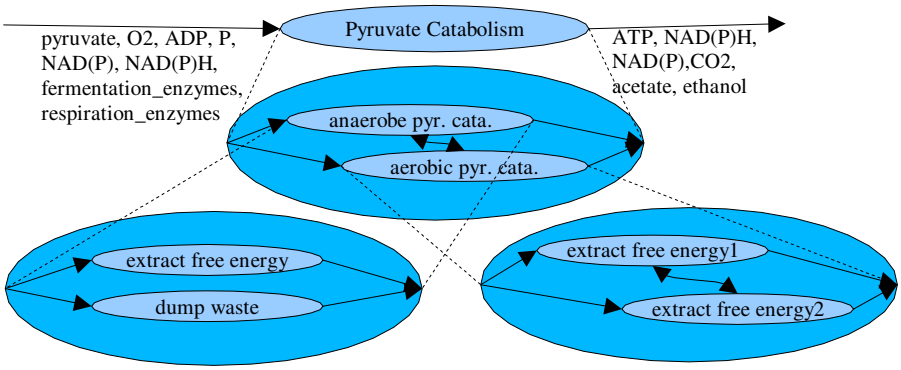


Figure 8. Pyruvate Catabolism organisation structure. For a desciption of the lower components, see Appendix A

Pyruvate Catabolism

Input: present(pyruvate), present(O2), present(ADP), present(P), present(NAD(P)), present(NAD(P)H), present(fermentation_enzymes), present(respiration_enzymes)

Output: present(ATP), present(NAD(P)), present(NAD(P)H), present(CO2), present(acetate), present(ethanol)

PD1

input:present(pyruvate) & present(ADP) & present(P) & present(NAD(P)H) & present(O2) & present(respiration_enzymes)
→→0;0;4;4 output:present(ATP) & present(NAD(P)) & present(CO2).

PD2

input:present(pyruvate) & present(ADP) & present(P) & present(NAD(P)H) & present(fermentation_enzymes)
→→0;0;4;4 output:present(ATP) & present(NAD(P)) & present(acetate) & present(ethanol).

Logical relationships for Catabolism
The general format provides:

DP(Glycolysis) & DP(Pyruvate-Catabolism) & DP(Glycogen-Catabolism)
& TRD(Catabolism) & IID(Catabolism) ⇒ DP(Catabolism)

In more detail, the following relationships hold:

GD1 & PD1 & TRD(Catabolism) & IID(Catabolism) ⇒ CaD1
GD1 & PD2 & TRD(Catabolism) & IID(Catabolism) ⇒ CaD2

Appendix B *E. coli*'s Internal Dynamics: Anabolism

Within Anabolism four children roles occur: Maintenance, Intermediary Synthesis, Aminoacids Synthesis, and DNA/RNA/Protein Synthesis. The dynamics of each of these will be addressed. Lower aggregation levels can be found in the Appendix.

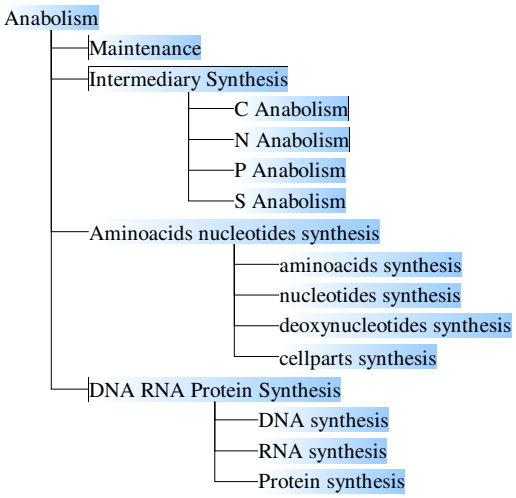


Figure 9. Anabolism hierarchy

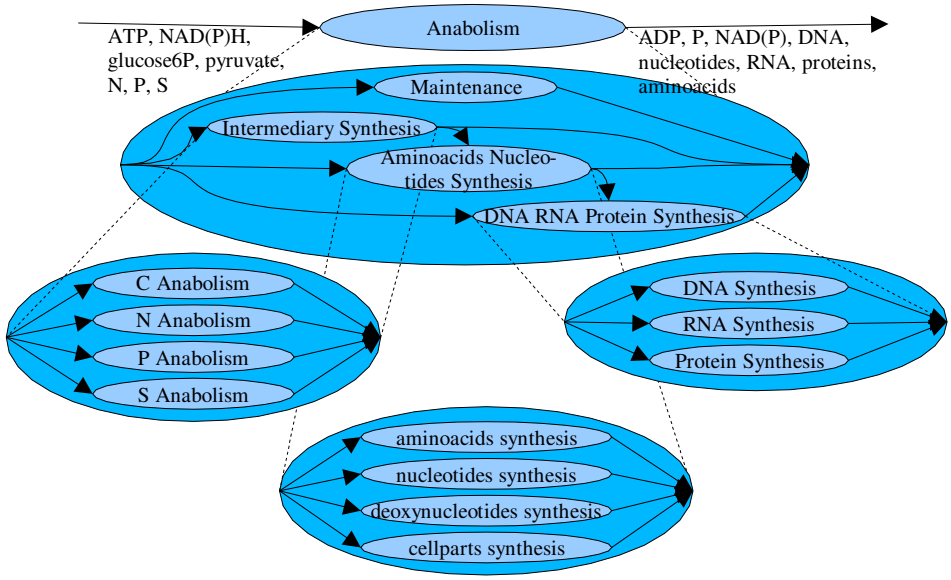


Figure 10. Anabolism organisation structure. For a desciption of the lower components, see Appendix A

Anabolism

Input: present(ATP), present(NAD(P)H), present(glucose6P), present(pyruvate), present(N), present(P), present(S)

Output: present(ADP), present(P), present(NAD(P)), present(aminoacids), present(nucleotides), present(DNA), present(RNA), present(proteins)

AD1

input:present(ATP) & present(NAD(P)H) & present(glucose6P) & present(pyruvate) & present(N) & present(P) & present(S)
$\twoheadrightarrow_{2;6;4;4}$ output:present(ADP) & present(P) & present(NAD(P)) & present(nucleotides) & present(aminoacids)

Maintenance

Input: present(ATP).

Output: present(ADP), present(P).

MaD1

input:present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(ADP) & present(P).

Intermediary Synthesis

Input: present(ATP), present(glucose6P), present(pyruvate), present(N), present(P), present(S).

Output: present(ADP), present(P), present(C_intermediaries), present(N_intermediaries), present(P_intermediaries), present(S_intermediaries).

ID1

input:present(ATP) & present(glucose6P) & present(pyruvate) & present(N) & present(P) & present(S)
$\twoheadrightarrow_{0;0;4;4}$ output:present(ADP) & present(P) & present(C_intermediaries) &

present(N_intermediaries) & present(P_intermediaries) &
present(S_intermediaries).

Aminoacids Nucleotides Synthesis

Input: present(ATP), present(NAD(P)H), present(C_intermediaries), present(N_intermediaries), present(P_intermediaries), present(S_intermediaries)

Output: present(ADP), present(P), present(NAD(P)), present(aminoacids), present(nucleotides), present(deoxynucleotides)

AnD1

input:present(ATP) & present(NAD(P)H) & present(C_intermediaries) &
present(N_intermediaries) & present(P_intermediaries) & present(S_intermediaries)
$\twoheadrightarrow_{0;0;4;4}$ output:present(ADP) & present(P) & present(NAD(P)) & present(aminoacids) &
present(nucleotides) & present(deoxynucleotides)

DNA RNA Protein Synthesis

Input: present(ATP), present(NAD(P)H), present(aminoacids), present(nucleotides), present(deoxynucleotides)

Output: present(ADP), present(P), present(NAD(P)), present(DNA), present(RNA), present(proteins)

DD1

input:present(ATP) & present(NAD(P)H) & present(C_aminoacids) & present(nucleotides) &
present(deoxynucleotides)
$\twoheadrightarrow_{0;0;4;4}$ output:present(ADP) & present(P) & present(NAD(P)) & present(DNA) &
present(RNA) & present(proteins)

Logical relationships for Anabolism

The general pattern provides

DP(Maintenance) & DP(Intermediary-Synthesis) &
DP(Aminoacids-Nucleotides-Synthesis) & DP(DNA-RNA-Protein-Synthesis) &
TRD(Anabolism) & IID(Anabolism) $\Rightarrow$ DP(Anabolism)

In more detail:

MaD1 & ID1 & AnD1 & DD1 & TRD(Anabolism) & IID(Anabolism) $\Rightarrow$ AD1

Here, TRD(Anabolism) are the transfer dynamics within the Anabolism role, and IID(Anabolism) are the interlevel interaction dynamics within the Anabolism role.

Appendix C *E. coli*'s Internal Dynamics: Transport

Within Transport two children roles occur: Import and Export. The dynamics of each of these will be addressed. Furthermore, Import's children Nutrition Import and Resource Import will be described. Lower aggregation levels can be found in the Appendix.

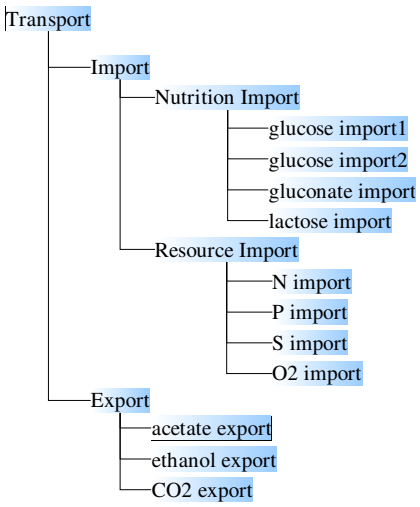


Figure 11. Transport hierarchy

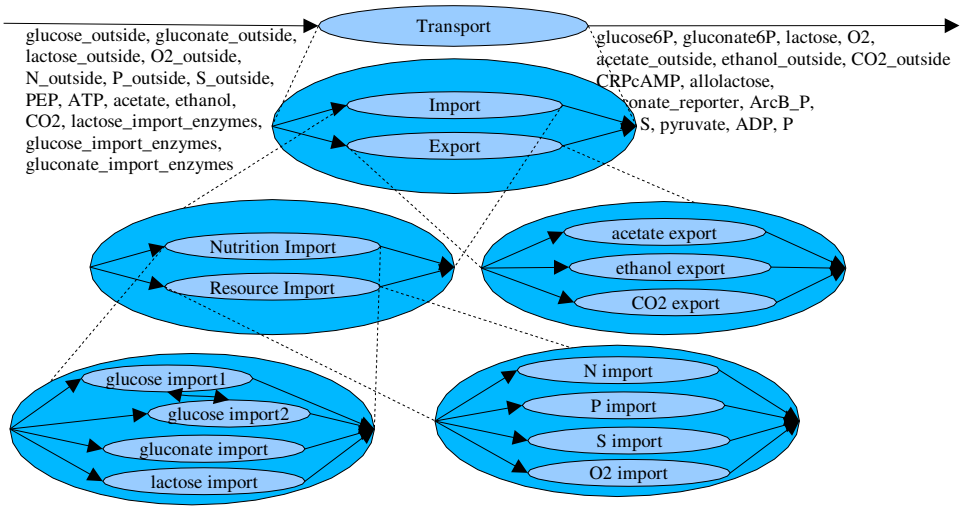


Figure 12. Transport organisation structure. For a desciption of the lower components, see Appendix A

Transport

Input: present(glucose_outside), present(lactose_outside), present(gluconate_outside), present(O2_outside), present(N_outside), present(P_outside), present(S_outside), present(PEP), present(ATP), present(acetate), present(ethanol), present(CO2), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)

Output: present(glucose6P), present(lactose), present(gluconate6P), present(O2), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount),

present(ArcB_P), present(N), present(P), present(S), present(pyruvate),
present(ADP), present(P), present(acetate_outside), present(ethanol_outside),
present(CO2_outside)

The timings here sometimes contain negative values. This is because the delay time only starts after the antecedent duration has passed. For example, timing –4;0;4;4 for TrD2 means that when gluconate becomes present outside at a time t0 for 4 seconds, then the delays will start to count at time t0+4. Thus the effects can happen between t0+4 + –4 seconds and t0+4 + 0 seconds, lasting at least 4 seconds. These timings thus ensure that the effects will happen at or after their causes happen.

Also note that the properties here are the combination of the properties of the Import and Export roles. For example, TrD4 is equal to ImD3. This is because the subroles Import and Export operate 'in parallel'.

TrD1
input:present(glucose_outside) & present(PEP) & present(glucose_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(glucose6P) & present(pyruvate)

TrD2
input:present(gluconate_outside) & present(ATP) & present(gluconate_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(gluconate6P) & present(ADP) & present(P)

TrD3
input:present(lactose_outside) & present(ATP) & present(lactose_import_enzymes)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(lactose) & present(ADP) & present(P)

TrD4
input:present(O2_outside) $\twoheadrightarrow_{0;0;4;4}$ output:present(O2) & present(ArcB_P)

TrD5
input:present(N_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(N) & present(ADP) & present(P)

TrD6
input:present(P_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(P) & present(ADP) & present(P)

TrD7
input:present(S_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(S & present(ADP) & present(P)

TrD8
input:present(acetate) $\twoheadrightarrow_{0;0;4;4}$ output:present(acetate_outside)

TrD9
input:present(ethanol) $\twoheadrightarrow_{0;0;4;4}$ output:present(ethanol_outside)

TrD10
input:present(CO2) $\twoheadrightarrow_{0;0;4;4}$ output:present(CO2_outside)

TrD11
input: not present(glucose_outside)
$\twoheadrightarrow_{-4;0;4;4}$ output:present(CRPcAMP)

TrD12
input:present(lactose_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(allolactose)

TrD13
input:present(gluconate_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(gluconate6P_observation_amount)

Import
Input: present(glucose_outside), present(lactose_outside), present(gluconate_outside),

present(O2_outside), present(N_outside), present(P_outside), present(S_outside), present(PEP), present(ATP), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)

Output: present(glucose6P), present(lactose), present(gluconate6P), present(O2), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount), present(ArcB_P), present(N), present(P), present(S), present(pyruvate), present(ADP), present(P)

ImD1

input:present(glucose_outside) & present(PEP) & present(glucose_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(glucose6P) &present(pyruvate)

ImD2

input:present(gluconate_outside) & present(ATP) & present(gluconate_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(gluconate6P) & present(ADP) & present(P)

ImD3

input:present(lactose_outside) & present(ATP) & present(lactose_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(lactose) & present(ADP) & present(P)

ImD4

input:present(O2_outside) $\twoheadrightarrow_{0;0;4;4}$ output:present(O2) & present(ArcB_P)

ImD5

input:present(N_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(N) & present(ADP) & present(P)

ImD6

input:present(P_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(P) & present(ADP) & present(P)

ImD7

input:present(S_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(S & present(ADP) & present(P)

ImD8

input: not present(glucose_outside) $\twoheadrightarrow_{-4;0;4;4}$ output:present(CRPcAMP)

ImD9

input:present(lactose_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(allolactose)

ImD10

input:present(gluconate_outside) $\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(gluconate6P_observation_amount)

Nutrition Import

Input: present(glucose_outside), present(lactose_outside), present(gluconate_outside), present(PEP), present(ATP), present(lactose_import_enzymes), present(glucose_import_enzymes), present(gluconate_import_enzymes)

Output: present(glucose6P), present(lactose), present(gluconate6P), present(CRPcAMP), present(allolactose), present(gluconate6P_observation_amount), present(pyruvate), present(ADP), present(P)

ND1

input:present(glucose_outside) & present(PEP) & present(glucose_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(glucose6P) & present(pyruvate)

ND2

input:present(gluconate_outside) & present(ATP) & present(gluconate_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(gluconate6P) & present(ADP) & present(P)

ND3

input:present(lactose_outside) & present(ATP) & present(lactose_import_enzymes) $\twoheadrightarrow_{-4;0;4;4}$ output:present(lactose) & present(ADP) & present(P)

ND4

input: not present(glucose_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(CRPcAMP)

ND5
input:present(lactose_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(allolactose)

ND6
input:present(gluconate_outside)
$\twoheadrightarrow_{0;0;0.230;0.230}$ output:present(gluconate6P_observation_amount)

Resource Import

Input: present(O2_outside), present(N_outside), present(P_outside), present(S_outside), present(ATP)
Output: present(O2), present(ArcB_P), present(N), present(P), present(S), present(ADP), present(P)

RD1
input:present(O2_outside) $\twoheadrightarrow_{0;0;4;4}$ output:present(O2) & present(ArcB_P)

RD2
input:present(N_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(N) & present(ADP) & present(P)

RD3
input:present(P_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(P) & present(ADP) & present(P)

RD4
input:present(S_outside) & present(ATP) $\twoheadrightarrow_{0;0;4;4}$ output:present(S & present(ADP) & present(P)

Export

Input: present(acetate) present(ethanol) present(CO2)
Output: present(acetate_outside) present(ethanol_outside) present(CO2_outside)

ED1
input:present(acetate) $\twoheadrightarrow_{0;0;4;4}$ output:present(acetate_outside)

ED2
input:present(ethanol) $\twoheadrightarrow_{0;0;4;4}$ output:present(ethanol_outside)

ED3
input:present(CO2) $\twoheadrightarrow_{0;0;4;4}$ output:present(CO2_outside)

Logical relationship for Transport

Because

DP(Transport) = DP(Import) & DP(Export),

it trivially holds that

DP(Import) & DP(Export) & TRD(Transport) & IID(Transport) $\Rightarrow$ DP(Transport)

Logical relationship for Import

Because

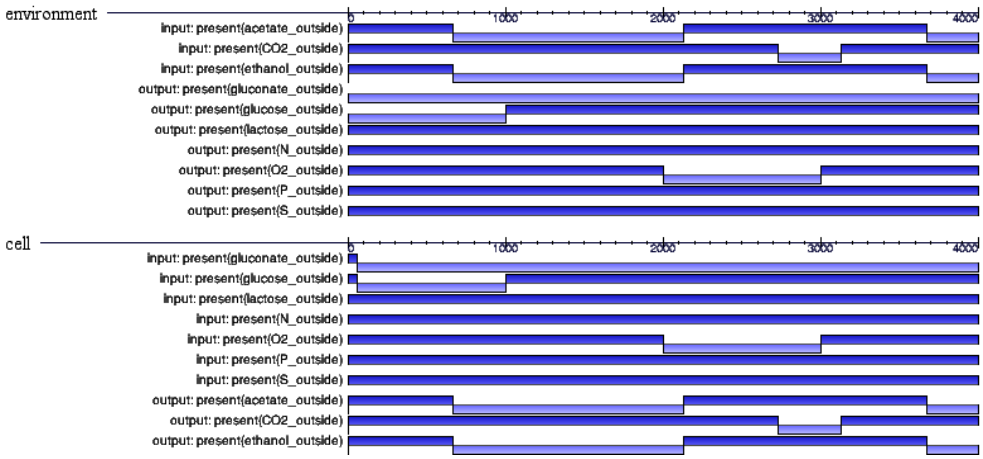
DP(Import) = DP(Nutrition-Import) & DP(Resource-Import),

it trivially holds that

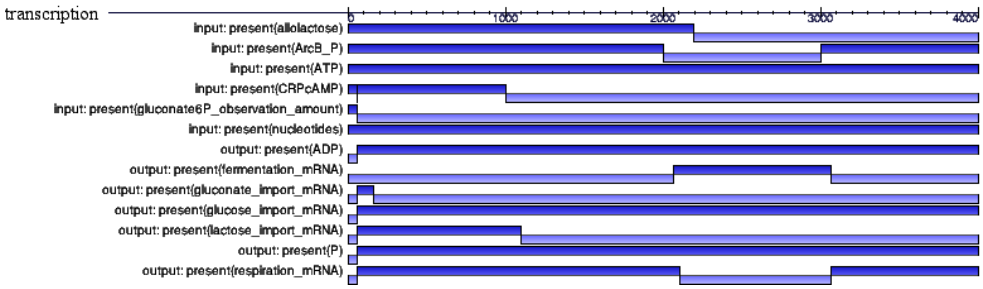
DP(Nutrition-Import) & DP(Resource-Import) & TRD(Import) & IID(Import) ⇒ DP(Import)

Appendix D More details of simulation results

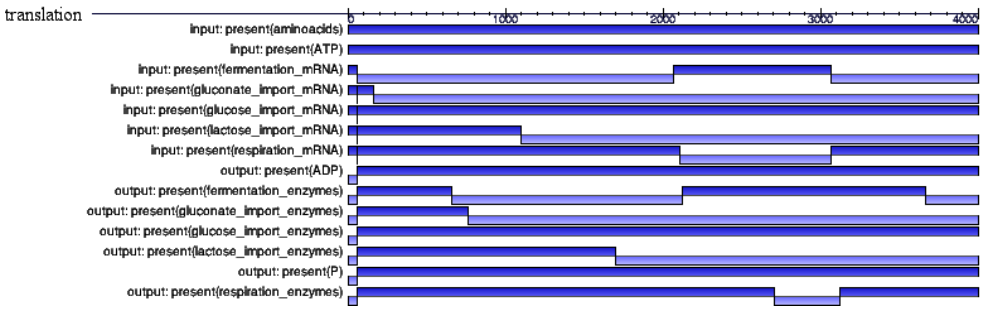
This example simulation follows what is described in Section 6.1.

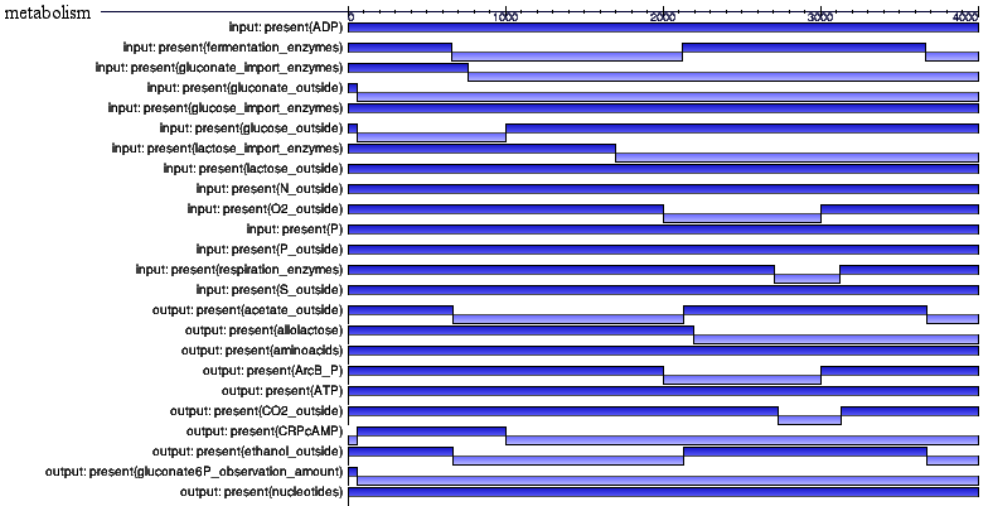


The environment was slowly changing over time. Lactose was always present, as were N, P and S. Gluconate was absent. At the start oxygen was present and glucose was absent. At time 1000 (seconds) glucose was added to the environment. At the start, all values on the inputs were set to true for 60 seconds, including for example gluconate outside. This could be specified in more detail, if another particular initialisation state were desired. At time 2000 oxygen was removed from the environment. The oxygen was added again at time 3000. The outputs of the environment are decided beforehand for the experiment, the outputs of the cell indicate the response of the bacterium. The output of the cell to the environment caused acetate, ethanol and CO_2 production at the very beginning, but as the cell adapted to the situation only CO_2 was produced. As oxygen was removed the CO_2 emissions stopped after a while, and acetate and ethanol were produced instead. After the oxygen was added again the cell adapted by stopping the acetate and ethanol emissions and returning to CO_2 production.



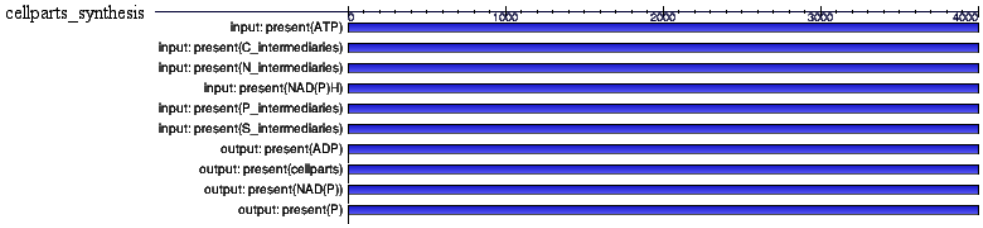
Inside the cell, the metabolism, the translation and the transcription are considered. Note that for example the ATP is not consumed, in general in this model substrates are not consumed, and products have a minimum lifetime assigned. Consumption of ATP could be added by putting the removal of ATP as the consequent of the dynamical properties. The transcription shows the detector molecules allolactose, ArcB-P and CRPcAMP doing their jobs. At first allolactose is present because lactose is present in the environment. Also ArcB-P is present, because there is oxygen. CRPcAMP is present because there is no glucose. There is no internal gluconate (a reporter of the presence of external gluconate) either. The detector molecules then change, observing the changes in the world, with the exception of allolactose, which disappears some time after t=2000 even though there is still lactose in the environment. The presence of glucose has caused CRPcAMP to go down and thereby inhibits the lactose import enzymes to be synthesized. After a while the enzymes have been diluted or turned over causing their concentration to decrease, making observation of lactose impossible. Based on these detectors and using the ATP and nucleotides that are input to the transcription from the metabolism, the mRNA was transcribed. Glucose import mRNA was always synthesized. At the start lactose-import mRNA was created enabling the model cell to feed on the lactose in the absence of glucose. The lactose import mRNA transcription stopped when glucose was added to the medium. Later, fermentation mRNA was created as the oxygen was removed, and respiration-mRNA transcription ceased. Fermentation mRNA instead of respiration mRNA was transcribed when oxygen was added again shortly after t = 3000.





The translation takes the mRNA from the transcription and uses ATP and aminoacids from the metabolism to translate the mRNA to proteins. The enzymes among the latter catalyze the metabolism. The metabolism takes ADP and P from the translation and transcription, as well as input from the environment. It makes free energy available, detects substances in the environment and provides building blocks, such as aminoacids and nucleotides to the translation and transcription. Waste CO_2, acetate and ethanol were output to the environment. In the simulated trace, steady states established as the cell adapted to each externally defined state. When lactose was the only carbon substrate, the lactose was detected, and the the enzymes from translation regulated the metabolism to perform lactose uptake. When glucose was been added the glucose import enzymes were already there, the corresponding gene expression being constitutive. Respiration and fermentation enzyme levels adjusted to the oxygen conditions outside. CO_2 was produced when oxygen was present, but as the latter was removed, the cell quickly turned to fermentation.

More detail can be seen when inspecting components further down the hierarchy, see e.g. the cellparts synthesis:



The cellparts synthesis was in a steady state throughout the entire simulation, even though the environment was changing. This shows the robustness of the model cell towards changes in the environment. A steady supply of ATP, C, N, P and S intermediaries as well

as NAD(P)H was on the input. Cellparts are continually being produced on the output: the cell is growing.