Multiagent and Grid Systems – An International Journal 3 (2007) 199–223
IOS Press

On the use of organisation modelling techniques to address biological organisation

Tibor Bosse[a], Catholijn M. Jonker[b] and Jan Treur[a,*]
[a]*Vrije Universiteit Amsterdam, Agent Systems Research Group, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands*
[b]*Delft University of Technology, Man Machine Interaction Group, Department of Mediametics, Mekelweg 4, 2628 CD Delft, The Netherlands*

Received 10 June 2005 (MAS*BIOMED 2005)
Revised 6 March 2006, 16 October 2006
Accepted 7 November 2006

Abstract. This paper explores how the dynamics of complex biological processes can be modelled and simulated as an organisation of multiple agents. This modelling perspective identifies organisational structure occurring in complex decentralised processes and handles complexity of the analysis of the dynamics by structuring these dynamics according to an organisational structure. More specifically, dynamic properties at different levels of aggregation in the organisational structure are identified, and related to each other according to the organisational structure. The applicability of this organisational modelling approach to address complexity in biological context is illustrated by two case studies: the organisation of intracellular processes and the organisation of the circulatory system.

Keywords: Dynamics, organisation modelling, systems biology

Part I: Problem description

1. Introduction

The area of systems biology aims at a system-level understanding of biological processes. In [23] it is stated that systems biology requires a shift in our notion of "what to look for" in biology: 'While an understanding of genes and proteins continues to be important, the focus is on understanding a system's structure and dynamics. Because a system is not just an assembly of genes and proteins, its properties cannot be fully understood merely by drawing diagrams of their interconnections. Although such a diagram represents an important first step, it is analogous to a static roadmap, whereas what we really seek to know are the traffic patterns, why such traffic patterns emerge, and how we can control them [23, p. 1662].' Analysing the structure and dynamics of complex biological processes is a nontrivial task. In many disciplines, some type of *organisational structure* is exploited to handle such complex (decentralised) dynamics. For example, the dynamics that emerge from multiple interacting *agents* within human society have been studied within Social Sciences in the area of Organisation Theory, and within Artificial Intelligence in the area of Agent Systems; e.g. [10,25,26,29,34,38]. To manage complex, decentralised dynamics in human society, organisational structure is a crucial element: organisation provides a structuring and co-ordination of the processes in such a manner that a process (or agent) involved can function in a more adequate manner. The dynamics shown by a given organisational structure are much more dependable than in an entirely unstructured situation. It is assumed that the organisational structure itself is relatively stable, i.e., the structure may change, but the frequency and scale of change are assumed low compared

*Corresponding author. Tel.: +31 20 598 7763; mobile +31 6 15028916; Fax: +31 20 598 7653; E-mail: treur@cs.vu.nl.

ISSN 1574-1702/07/$17.00

to the more standard dynamics through the structure. Also in Nature several forms of organisational structure have been developed; typical examples are a beehive, the coordinated processes of organs in mammals, and the well-organised biochemistry of a living cell.

By using multi-agent organisation modelling techniques for analysis and simulation, the inherent complexity of the dynamics of multiple interacting processes within a society can be made manageable by choosing the right level of abstraction in describing them. As also discussed in [21], many phenomena in nature (or in the laboratory) have the same characteristic: they also involve complex dynamics of multiple distributed processes and their interaction. Therefore, a natural question is whether a multi-agent-organisation modelling perspective is promising for this domain of biological complexity. This question is addressed in this paper.

Organisations can be viewed in two ways: (1) as adaptive complex information processing systems of (boundedly) rational agents, and (2) as tools for control; central issues are [26]:

- How to identify properties of the whole, given properties of parts; 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?'
- How to identify properties of parts, given desired or required properties of the whole; 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?'

Recently a number of formal and computational modelling techniques have been developed that can be used for simulation or for formal analysis of the dynamics within a multi-agent organisation. Examples of this formalisation trend can be found in books such as [26,34], and in a recently created journal: Computational and Mathematical Organisation Theory; e.g. [31]. For dynamics of an organisation, different levels of aggregation can be identified, from single agent behaviour to the dynamics of the overall organisation. Dynamics can be described in an abstract manner by focusing on one of these levels and specifying dynamic properties for this level. Moreover, interlevel relationships between dynamic properties at different levels can be identified.

One of the organisation modelling approaches that have been developed within the agent systems area is the Agent-Group-Role (AGR) approach, introduced in [11], extended with operational semantics in [12], and with a modelling approach for dynamic properties in [20]. A related dynamic modelling framework for specification, analysis and simulation of AGR-organisation models, and supported by a software environment is described in [22]. This dynamic modelling environment allows to:

- *specify dynamic properties* for the different elements and levels of aggregation within an AGR organisation model
- *relate these dynamic properties* to each other according to the organisational structure
- use dynamic properties in *executable* form as a declarative *specification of a simulation model*
- perform *simulation experiments*
- automatically *check dynamic properties* for simulated or empirical traces

The main goal of this paper is to explore to what extent the AGR approach and the corresponding modelling framework by [22] can be useful for the analysis of complex biological processes from an organisation modelling perspective. To this end, first in Section 2 Ferber and Gutknecht's [11] AGR approach is introduced, with an emphasis on organisational structure. It is illustrated by a model of the organisational structure of intracellular processes within the unicellular organism *Escherichia coli* [32]. Section 3 addresses the dynamics of the organisation, described in terms of dynamics properties expressed in a Temporal Trace Language and in Section 4 relations between different levels of aggregation are discussed, following [20]. Section 5 provides some results of simulation and computer supported verification. In Section 6 it is shown to what extent the modelling approach can be generalised to other biological domains. Section 7 relates the approach to other techniques used in the literature, and Section 8 is a conclusion.

Part II: Approach

2. Organisational structure

The Agent-Group-Role (AGR) organisation modelling approach [11] originates from the area of multi-agent systems. In this section, first a brief introduction of the AGR approach can be found (Section 2.1). Next, Section 2.2 illustrates how the approach can be applied in a biological domain by describing the internal organ-

isational structure of *E. coli*. In this example, which for reasons of presentation is kept limited, the main property to focus on is growth under different environmental circumstances.

2.1. The AGR organisation modelling approach

An AGR organisational structure for an overall process (or organisation) is a specification based on a definition of groups, roles and their relationships. An organisation as a whole is composed of a number of *groups*. A group structure identifies the *roles* and the *intragroup transfers* between roles. In addition, *intergroup role interactions* between roles of different groups specify the connectivity of groups within an organisation. Physical *agents* can be allocated to roles; they realise the organisation. However, the aim of an organisation model is to abstract from any specific agent allocated. Therefore instead of particular agents, roles are used as abstract entities, defining properties agents should have when they are to function in a given role within an organisation.

2.2. Organisational structure of the living cell

In Fig. 1 the aggregation levels of the AGR-organisation model of *E. coli* are 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 Control and Metabolism. The latter nodes are children of Cell. This means that they are the main categories or functional units that are distinguished for the processes in the cell. To be more specific, Metabolism and Control are the main parts of the processes of a cell. At one aggregation level lower, Metabolism expands to Catabolism, Anabolism and Transport. Catabolism is the category of processes that decompose substances and extract free energy from them. In Anabolism the processes that utilise this free energy to create more and more complex substances reside. Transport processes move substances across the cell membrane. Control is decomposed into Transcription and Translation. These processes generate mRNA and enzymes, respectively.

An AGR-model of *E. coli*'s organisational structure is shown in Fig. 2. The functional units Control and Metabolism are depicted as different *groups* here (depicted by the larger ovals). Their children (according to Fig. 1) are depicted in Fig. 2 as *roles* (depicted by smaller ovals) within the groups. The behaviour of these roles, in the next section described by role behaviour properties, is as follows: they receive as input the presence of some substances generated by another role, in order to generate the presence of some new substances as output. The solid arrows represent *intragroup role transfers*, the transfer of substances between roles: they express that a substance produced by one role is used by another role. Notice that each group contains an additional Portal role. The idea is that these roles collect the output substances produced by all other roles within their group, to be able to interact with the other group. The dashed arrows between both portal roles represent *intergroup role interactions*, relating the input of one portal role to the output of the other. Note that the model depicted in Fig. 2 is a simplification of the true living cell. For example, only control at the transcriptional and translational level is included, and 'post translational modifications' (such as phosphorylation) are left out. Nevertheless, it reflects the main aspects of its organisational structure in a way that is understandable.

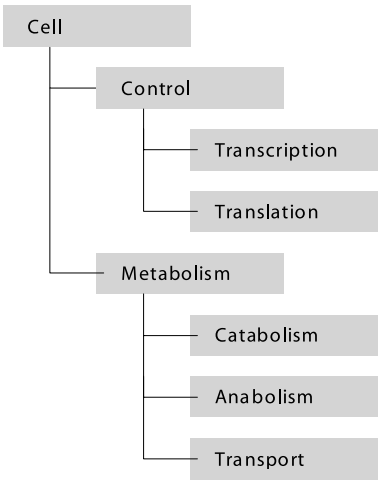

Fig. 1. Overview of the aggregation levels of the organisation model of *E. coli*.

3. Organisation dynamics

The AGR organisation modelling approach was extended with a dynamic modelling approach in [20]. To characterise the dynamics within an organisation, dynamic properties of various types can be formulated. For example, a dynamic property of the organisation as a whole, such as

> If oxygen, resources and some nutrients are externally available, then the cell will produce CO_2.

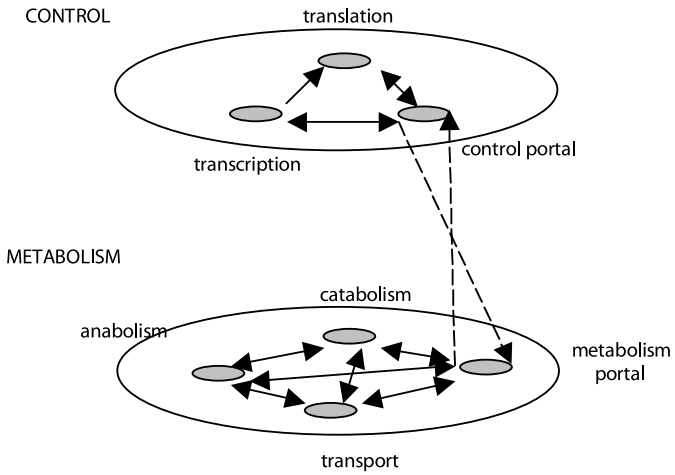

Fig. 2. *E.coli* : groups and interactions.

Other examples are dynamic properties of one specific role within an organisation, or dynamic properties that characterise how two roles cooperate.

An organisational structure provides a basis to distinguish in a systematic manner dynamic properties for different elements and aggregation levels within the organisation. In particular, as an extension of the AGR organisation model dynamic properties can be specified for each of the following aggregation levels within the model:

I. At the (highest) aggregation level of the organisation as a whole
   - dynamic properties for the *organisation* as a whole; the highest aggregation level, relating any roles within the organisation over time;
   - dynamic properties for *intergroup role interaction*, relating the input of one role to the output of a role in another group;

II. At the aggregation level of a group within the organisation
   - dynamic properties at the level of a *group*, relating states of roles within a given group over time;
   - dynamic properties for *transfer* between roles within a group (from output state of the source role to input state of the destination role);

III. At the (lowest) aggregation level of a role within a group
   - dynamic properties at the level of a *role* within a group, relating input and output state (and possibly internal state) of the role;

To describe the dynamics of *E. coli*'s intracellular processes, all types of dynamic properties are used.

3.1. Dynamic properties of the organisation as a whole

The example model for *E. coli*'s dynamics was inspired by the model described in [18], which is based on a different modelling approach: the compositional organisation modelling approach. For the example of the living cell, global properties of the organisation as a whole can be expressed in terms of interaction with an Environment. Note that this environment is not shown in Figs 1 and 2 (since we consider it not being part of the organisation itself), but we assume that the roles in the organisation can interact with it. The cell can use as input from the environment the (external) presence of glucose, gluconate, lactose, O_2, N, P and S. It may export CO_2, ethanol and acetate to the environment. For example, Cell Property 1 (CP1) in Box 1 specifies the property that if O_2 is externally available, as well as resources and at least one of the nutrients glucose, lactose, gluconate, then the cell produces CO_2. Moreover, CP2 specifies an analogue property for the anaerobic case. Note that in addition to d1, w1, also α is a variable, which makes it possible to have different instantiations of one property. For instance, property CP1(d1, w1, α) may be instantiated to CP1(0.3, 0.5, glucose). For all properties, notice that it is explicitly mentioned when interaction with the environment is involved. More specifically, if by transport a substance is emitted to the environment, this is phrased as 'exports to the Environment', and if a substance is available for transport (i.e., import) within the environment, this

is phrased as 'is present within the Environment'. In contrast, the internal exchange of the presence of substances within the organisation model is indicated by the words *generates* and *receives*. For α ranging over {glucose, lactose, gluconate}, the properties shown in Box 1 characterise the cell-environment dynamics.

Within Computer Science and Artificial Intelligence a number of high-level specification languages have been developed to specify dynamic properties with mathematical precision, thereby allowing qualitative and (sometimes) quantitative aspects. To formally express the properties presented in this paper, the high-level Temporal Trace Language (TTL) has been chosen, see [4,19], to model and analyse the internal and external dynamics of agents, and of multi-agent organisations.

A *trace* or *trajectory* in the state space is a sequence of states indexed over time. *States* are characterised by *state properties* indicating, for example, values of certain variables. *Dynamic properties* are properties of traces, i.e., properties that relate states over time. To express dynamic properties the sorted predicate logic temporal trace language TTL is used. This language is built on atoms referring to, e.g., a *trace* γ, a *time point* t and a *state property* p, such as 'in trace γ at time point t state property p holds', formalised by state(γ, t) |= p.

As an example, formalising dynamic property CP1 from Box 1 in TTL yields the following:

CP1(d1, w1, α**)** $\equiv$
$\forall$t1,t2
[[t1 < t2 & $\forall$t' [t1 $\leqslant$ t' $\leq$ t2 $\Rightarrow$
state(γ, t') |= in_environment(α)$\wedge$
state(γ, t') |= in_environment(O_2)$\wedge$
state(γ, t') |= in_environment(N) $\wedge$
state(γ, t') |= in_environment(P) $\wedge$
state(γ, t') |= in_environment(S)]]
$\Rightarrow$ $\exists$t3 [t2 + d1 $\leqslant$ t3 $\leqslant$ t2 + w1 & state(γ, t3) |= cell_exports (CO_2)]

The Temporal Trace Language TTL can play a useful role in modelling complex phenomena from an agent-oriented perspective in the following manners:

- it provides a way to obtain well-defined and mathematically formalisable *specifications* of *dynamic properties* of externally observable agent behaviour, their internal processes, and their organisation; such dynamic properties can be specified at any level of precision as desired.
- for further *analysis* it supports the identification of formalised relationships between different dynamic properties, for example between properties of an agent's externally observable behaviour and its internal processes, or between properties of externally observable agent behaviour and properties of an organisation in which they function.
- it offers possibilities to specify and execute *simulation models* in a high level language, for example simulation of an agent's externally observable behaviour on the basis of its internal processes, or simulation of an organisation on the basis of given or assumed properties of externally observable behaviour of the agents involved.

Throughout the remainder of this paper, dynamic properties will not be formally expressed, but in the semi-formal format presented earlier, to enhance readability. Within this format, each property always holds for all traces γ over the ontology, but γ is not mentioned explicitly to keep the notation simple.

3.2. Intergroup role interaction properties

Within the AGR organisation modelling approach, intergroup role interaction properties model connections between groups by specifying how the input state of a role in one group can be (temporally) related to the output state of another role in a different group. Within the current example, the intergroup role interaction properties take care of the exchange of substances between both groups. This is done by relating the input of the portal role of one group to the output of the portal role of the other group. The properties expressing this are shown in Box 2. The delay parameters in these intergroup role interaction properties can be used to model some form of mobility of molecules produced by one process before they are used in another process. However, for simplicity we assume the exchange to be instantaneous, all delays (ci's and ri's) are 0 in this example, i.e. t' = t in the properties above.

3.3. Dynamic properties of the metabolism and control group

For each of the groups, dynamic properties are considered that contribute to the properties of the organisation as a whole. A group property is specified in terms of temporal relationships between input and output states of roles within this group.

Within the group Metabolism, which includes transportation through the cell's membrane (import and export), substances present outside the cell, but also substances produced by Control can be used. Likewise, it can produce substances that are exported to the environment, as well as substances used by Control. The

Box 1. Dynamic properties of the cell as a whole

CP1(d1, w1, α) **CO_2 production**
For any two time points t1 and t2
if between t1 and t2 the substances α, O_2, N, P and S are present within the Environment
then there exists a time point t3 with $t2+d1 \leq t3 \leq t2+w1$ such that at t3 the cell exports CO_2 to the Environment

CP2(d2, w2, α) **Acetate and ethanol production**
For any two time points t1 and t2
if between t1 and t2 the substances α, N, P and S are present within the Environment
and the substance O_2 is not present within the Environment
then there exists a time point t3 with $t2+d2 \leq t3 \leq t2+w2$ such that at t3 the cell exports acetate and ethanol to the Environment

Box 2. Dynamic properties for Intergroup Role Interaction

IGIP1(c1, r1) **Control Portal-Metabolism Portal Intergroup Role Interaction**
At any point in time t, for all substances β,
if Control Portal receives a substance β
then there exists a time point t' with $t+c1 \leq t' \leq t+r1$ such that at t' Metabolism Portal generates the substance β

IGIP2(c2, r2) **Metabolism Portal-Control Portal Intergroup Role Interaction**
At any point in time t, for all substances β
if Metabolism Portal receives a substance β
then there exists a time point t' with $t+c2 \leq t' \leq t+r2$ such that at t' Control Portal generates the substance β

exchange of substances to and from Control goes via the Metabolism Portal role. Metabolism property MP4 is an example of a complex property that has input from and output to both the environment and Control. For α ranging over {glucose, lactose, gluconate}, the dynamic properties in Box 3 characterise the Metabolism dynamics. As opposed to Metabolism, the group Control does not interact with the environment. Via its role Control Portal certain substances produced by Metabolism are available, and (abstracting from intermediate steps) it can itself produce particular enzymes, ADP, and P. In Box 4 the dynamic properties for the Control group are shown.

3.4. Transfer properties

Transfer properties are assumed to have a generic pattern: that every substance presence generated (in its output state) by any role r1 for any role r2 is received (in its input state) by role r2. In the example, for transfer properties similar assumptions are used as for intergroup role interaction properties, namely instantaneous transfer of all substances (i.e., no time durations taken into account for molecule mobility between chemical processes; all gi's and hi's are 0). All solid arrows in Fig. 2 refer to transfer properties. Since they all look the same, only two examples are shown in Box 5. Furthermore, notice that there is a transfer from Transcription to Translation, but not vice versa. For all other combination of roles, there is transfer in two directions.

3.5. Role behaviour properties

Dynamic properties for a role characterise how the role behaves, given its input. Such a dynamic property typically is expressed in terms of a temporal relationship between input state and output state of the role.

3.6. Roles within the Control group

The role Transcription can receive the substances nucleotides, ATP, ArcB_P, allolactose, CRPcAMP, and gluconate6P observation amount, all coming (via the Control Portal) from the Metabolism group. Depending on certain circumstances it will produce partic-

Box 3. Dynamic properties for the Metabolism group

MP0(m_{init}) Metabolism Initialisation
there exists a time point t with $0 \leq t \leq m_{init}$ such that at t
Metabolism Portal receives the substances ATP, nucleotides and aminoacids

MP1(p1, q1) Metabolism CRPcCAMP production
At any point in time t
if within the Environment the substance glucose is not present
then there exists a time point t' with $t+p1 \leq t' \leq t+q1$ such that at t'
Metabolism Portal receives the substance CRPcAMP

MP2(p2, q2) Metabolism allolactose production
At any point in time t
if within the Environment the substance lactose is present
then there exists a time point t' with $t+p2 \leq t' \leq t+q2$ such that at t'
Metabolism Portal receives the substance allolactose

MP3(p3, q3) Metabolism gluconate_6P production
At any point in time t
if within the Environment the substance gluconate is present
then there exists a time point t' with $t+p3 \leq t' \leq t+q3$ such that at t'
Metabolism Portal receives the substance gluconate_6P_observation_amount

MP4(p4, q4, α) Metabolism ATP-nucleotides-aminoacids production and CO_2 export
At any point in time t
if within the Environment the substances α, N, P, S and O_2 are present
and Metabolism Portal generates the substances ADP, P, respiration_enzymes and import_enzymes for α
then there exists a time point t' with $t+p4 \leq t' \leq t+q4$ such that at t'
Metabolism Portal receives the substances ATP, nucleotides and aminoacids
and the cell exports CO_2 to the Environment

MP5(p5, q5, α) Metabolism ATP-nucleotides-aminoacids production/acetate-ethanol export
At any point in time t
if within the Environment the substances α, N, P and S are present
and within the Environment the substance O_2 is not present
and Metabolism Portal generates the substances ADP, P, fermentation_enzymes and import_enzymes for α
then there exists a time point t' with $t+p5 \leq t' \leq t+q5$ such that at t'
Metabolism Portal receives the substances ATP, nucleotides and aminoacids
and the cell exports acetate and ethanol to the Environment

MP6(p6, q6) Metabolism ArcB_P production
At any point in time t
if within the Environment the substance O_2 is present
then there exists a time point t' with $t+p6 \leq t' \leq t+q6$ such that at t'
Metabolism Portal receives the substance ArcB_P

ular forms of mRNA, ADP, and P. See Box 6 for the dynamic properties of Transcription. The role Translation's input is amino acids and ATP (both produced by the Metabolism group), and a particular type of mRNA (produced by the Transcription role). It can produce ADP, P, and a particular enzyme, corresponding to the type of mRNA. For η ranging over {respiration, fermentation, glucose_import, lactose_import, gluconate_import}, the property in Box 7 characterises the Translation dynamics.

3.7. Roles within the Metabolism group

Within the Metabolism group, the role Catabolism receives the presence of several substances. Some of them are provided by the other roles, Anabolism and Transport, some others are provided (via the Metabolism Portal) by the Control group. Likewise, the substances it produces are also used by the roles Anabolism and Transport, and (via the Metabolism Portal) by the group Control. For δ ranging over {glucose6P,

Box 4. Dynamic properties for the Control group

CoP1(u1, v1) **Glucose_import_enzymes production**
At any point in time t
if Control Portal generates the substances nucleotides, ATP and aminoacids
then there exists a time point t' with t+u1 ≤ t' ≤ t+v1 such that at t'
Control Portal receives the substances ADP, P and glucose_import_enzymes

CoP2(u2, v2) **Respiration_enzymes production**
At any point in time t
if Control Portal generates the substances ArcB_P, nucleotides, ATP and aminoacids
then there exists a time point t' with t+u2 ≤ t' ≤ t+v2 such that at t'
Control Portal receives the substances ADP, P and respiration_enzymes

CoP3(u3, v3) **Fermentation_enzymes production**
At any point in time t
if Control Portal generates the substances nucleotides, ATP and aminoacids
and Control Portal does not generate the substance ArcB_P
then there exists a time point t' with t+u3 ≤ t' ≤ t+v3 such that at t'
Control Portal receives the substances ADP, P and fermentation_enzymes

CoP4(u4, v4) **Lactose_import_enzymes production**
At any point in time t
if Control Portal generates the substances allolactose, CRPcAMP, nucleotides, ATP and aminoacids
then there exists a time point t' with t+u4 ≤ t' ≤ t+v4 such that at t'
Control Portal receives the substances ADP, P and lactose_import_enzymes

CoP5(u5, v5) **Gluconate_import_enzymes production**
At any point in time t
if Control Portal generates the substances gluconate_6P_observation_amount, CRPcAMP, nucleotides, ATP and aminoacids
then there exists a time point t' with t+u5 ≤ t' ≤ t+v5 such that at t'
Control Portal receives the substances ADP, P and gluconate_import_enzymes

Box 5. Dynamic properties for transfer within the Control group

TP1(g1, h1) **Transcription-Translation Transfer**
At any point in time t, for all substances β
if Transcription generates a substance β
then there exists a time point t' with t+g1 ≤ t' ≤ t+h1 such that at t'
Translation receives the substance β

TP2(g2, h2) **Transcription-Control Portal Transfer**
At any point in time t, for all substances β
if Transcription generates a substance β
then there exists a time point t' with t+g2 ≤ t' ≤ t+h2 such that at t'
Control Portal receives the substance β

gluconate6P, lactose}, the dynamic properties shown in Box 8 characterise the Catabolism dynamics.

Like Catabolism, the role Anabolism also interacts with several roles in its own group. Apart from initialisation, its dynamics can be described by one single property; see Box 9. The role Transport is the only role within the organisation that interacts with the environment. Furthermore, it also interacts with roles in its own group, including the Metabolism Portal role, which serves as a portal to the Control group. Recall that the words *generates* and *receives* indicate that the substances are exchanged within the organisation model *internally*. In contrast, if by transport a substance is emitted to the environment, this is phrased as 'exports to the Environment', and if a substance is available for transport (i.e., import) within the environment, this is phrased as 'is present within the Environment'. For ε ranging over {N, P, S} and ζ ranging over {acetate,

Box 6. Dynamic properties for Transcription

TcP1(k1, l1) **Glucose_import_mRNA production**
At any point in time t
if Transcription receives the substances nucleotides and ATP
then there exists a time point t' with t+k1 ≤ t' ≤ t+l1 such that at t'
Transcription generates the substances ADP, P and glucose_import_mRNA

TcP2(k2, l2) **Respiration_mRNA production**
At any point in time t
if Transcription receives the substances ArcB_P, nucleotides and ATP
then there exists a time point t' with t+k2 ≤ t' ≤ t+l2 such that at t'
Transcription generates the substances ADP, P and respiration_mRNA

TcP3(k3, l3) **Fermentation_mRNA production**
At any point in time t
if Transcription receives the substances nucleotides and ATP
and Transcription does not receive the substance ArcB_P
then there exists a time point t' with t+k3 ≤ t' ≤ t+l3 such that at t'
Transcription generates the substances ADP, P and fermentation_mRNA

TcP4(k4, l4) **Lactose_import_mRNA production**
At any point in time t
if Transcription receives the substances allolactose, CRPcAMP, nucleotides and ATP
then there exists a time point t' with t+k4 ≤ t' ≤ t+l4 such that at t'
Transcription generates the substances ADP, P and lactose_import_mRNA

TcP5(k5, l5) **Gluconate_import_mRNA production**
At any point in time t
if Transcription receives the substances gluconate6P_observation_amount, CRPcAMP, nucleotides and ATP
then there exists a time point t' with t+k5 ≤ t' ≤ t+l5 such that at t'
Transcription generates the substances ADP, P and gluconate_import_mRNA

Box 7. Dynamic properties for Translation

TlP1(e1, f1, η) **Enzymes production**
At any point in time t
if Translation receives the substances aminoacids, ATP and mRNA for η
then there exists a time point t' with t+e1 ≤ t' ≤ t+f1 such that at t'
Translation generates the substances ADP, P and enzymes for η

ethanol, CO2}, the properties shown in Box 10 characterise the Transport dynamics.

4. Interlevel relations

The idea of expressing dynamic properties at different levels of aggregation is that certain logical interlevel relationships can be identified between properties at the different levels. Typically, dynamics of the whole organised (multi-agent) system can be related to dynamic group properties and intergroup interaction properties via the following pattern:

dynamic properties for the groups &
dynamic properties for intergroup role interaction
⇒ dynamic properties for the organisation

This implication (which also can be expressed as *logical entailment*) should be understood as follows: 'for any organisation, if for any trace the group properties and intergroup role interaction properties hold, then the general properties for the organisation also hold'. Likewise, dynamic properties of groups can be related to dynamic properties of roles in the following way:

dynamic properties for roles &
dynamic properties for transfer between roles ⇒
dynamic properties for a group

Box 8. Dynamic properties for Catabolism

CaP0(c_{init}) Catabolism Initialisation
there exists a time point t with $0 \leq t \leq c_{init}$ such that at t
Catabolism generates the substance NAD(P)

CaP1(i1, j1, δ) Catabolism Dynamics 1
At any point in time t
if Catabolism receives substances δ, ADP, P, NAD(P), O_2 and respiration_enzymes
then there exists a time point t' with $t+i1 \leq t' \leq t+j1$ such that at t'
Catabolism generates the substances glucose6P, pyruvate, ATP, NAD(P)H, PEP and CO_2

CaP2(i2, j2, δ) Catabolism Dynamics 2
At any point in time t
if Catabolism receives the substances δ, ADP, P, NAD(P) and fermentation_enzymes
then there exists a time point t' with $t+i2 \leq t' \leq t+j2$ such that at t'
Catabolism generates the substances glucose6P, pyruvate, ATP, NAD(P)H, PEP, acetate and ethanol

Box 9. Dynamic properties for Anabolism

AP0(a_{init}) Anabolism Initialisation
there exists a time point t with $0 \leq t \leq a_{init}$ such that at t
Anabolism generates the substances PEP and ATP

AP1(m1, n1) Anabolism Dynamics 1
At any point in time t
if Anabolism receives substances ATP, NAD(P)H, glucose6P, pyruvate, N, P and S
then there exists a time point t' with $t+m1 \leq t' \leq t+n1$ such that at t'
Anabolism generates the substances ADP, P, NAD(P), nucleotides and aminoacids

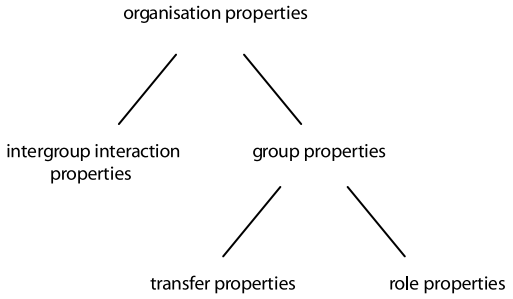

Fig. 3. Overview of interlevel relationships between dynamic properties within an organisation model.

A general overview of the interlevel relationships between dynamic properties at different aggregation levels is depicted as an AND-tree in Fig. 3.

The next sections will describe the interlevel relationships between dynamic properties within the example of the living cell.

4.1. Interlevel relations for overall properties of the cell dynamics

Global property CP1(glucose) states that the cell will produce CO_2 if the substances O_2, glucose, N, P and S are available within the environment. Careful investigation of the group properties and intergroup role interaction properties yields the interlevel relationship depicted in Fig. 4.

The interlevel relationship between Global Property CP1(lactose) and the properties it depends on is depicted in Fig. 5. This property states that the cell will produce CO_2 if the substances O_2, lactose, N, P and S are available within the environment. However, nothing is said about the availability of glucose. An argumentation of the dependencies shown could therefore be obtained by reasoning by cases: suppose all lower level properties of Fig. 5 hold. Then, if glucose is present within the environment, this will be used in order to export CO_2, according to properties MP0, MP6, IGIP2, CoP1, CoP2, IGIP1, and MP4(glucose). But if glucose is not present and lactose is present within the environment, then lactose will be used, according to properties MP0, MP1, MP2, MP6, IGIP2, CoP2, CoP4, IGIP1, and MP4(lactose). Hence, if all lower level properties hold, then CO_2 will always be exported, making use of either glucose or lactose from the environment. It may thus be concluded that CP1(lactose) holds.

Box 10. Dynamic properties for Transport

TpP1(s1, t1) **Transport CRPcAMP production**
At any point in time t
if within the Environment the substance glucose is not present
then there exists a time point t' with t+s1 ≤ t' ≤ t+t1 such that at t'
Transport generates the substance CRPcAMP

TpP2(s2, t2) **Transport allolactose production**
At any point in time t
if within the Environment the substance lactose is present
then there exists a time point t' with t+s2 ≤ t' ≤ t+t2 such that at t'
Transport generates the substance allolactose

TpP3(s3, t3) **Transport gluconate_6P_observation_amount production**
At any point in time t
if within the Environment the substance gluconate is present
then there exists a time point t' with t+s3 ≤ t' ≤ t+t3 such that at t'
Transport generates the substance gluconate6P_observation_amount

TpP4(s4, t4) **Transport glucose6P production**
At any point in time t
if Transport receives the substances PEP and glucose_import_enzymes
and within the Environment the substance glucose is present
then there exists a time point t' with t+s4 ≤ t' ≤ t+t4 such that at t'
Transport generates the substances glucose6P and pyruvate

TpP5(s5, t5) **Transport gluconate6P production**
At any point in time t
if Transport receives the substances ATP and gluconate_import_enzymes
and within the Environment the substance gluconate is present
then there exists a time point t' with t+s5 ≤ t' ≤ t+t5 such that at t'
Transport generates the substances gluconate6P, ADP and P

TpP6(s6, t6) **Transport lactose production**
At any point in time t
if Transport receives the substances ATP and lactose_import_enzymes
and within the Environment the substance lactose is present
then there exists a time point t' with t+s6 ≤ t' ≤ t+t6 such that at t'
Transport generates the substances lactose, ADP and P

TpP7(s7, t7) **Transport O_2 production**
At any point in time t
if within the Environment the substance O_2 is present
then there exists a time point t' with t+s7 ≤ t' ≤ t+t7 such that at t'
Transport generates the substances O_2 and ArcB_P

TpP8(s8, t8, ε) **Transport resources production**
At any point in time t
if Transport receives the substance ATP
and within the Environment the substance ε is present
then there exists a time point t' with t+s8 ≤ t' ≤ t+t8 such that at t'
Transport generates the substances ε, ADP and P

TpP9(s9, t9, ξ) **Transport environment export**
At any point in time t
if Transport receives the substance ξ
then there exists a time point t' with t+s9 ≤ t' ≤ t+t9 such that at t'
the cell exports ξ to the Environment

A complete specification of the interlevel relations for the global properties is given in Box 11. In addition to the relationships between the properties, dependencies between corresponding parameters are given. Recall that in this example the delays of the intergroup role interaction properties (all ci's and ri's) are assumed

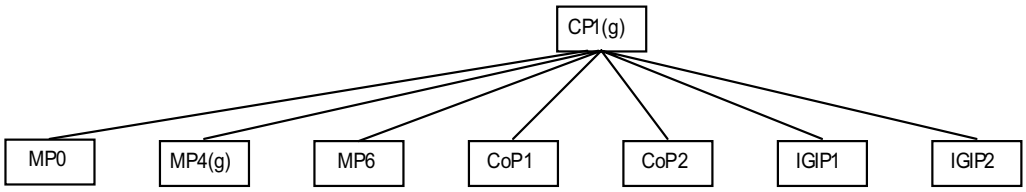

Fig. 4. Property CP1(g) related to group properties and intergroup role interaction properties.

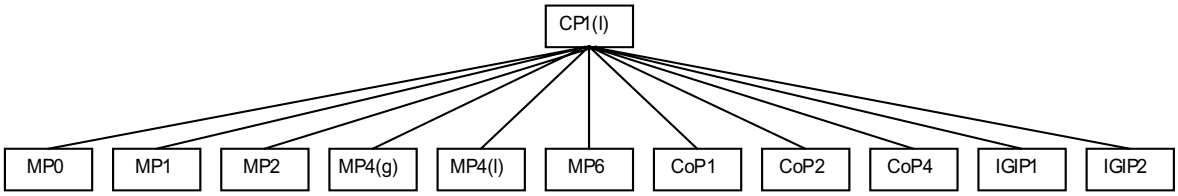

Fig. 5. Property CP1(l) related to group properties and intergroup role interaction properties.

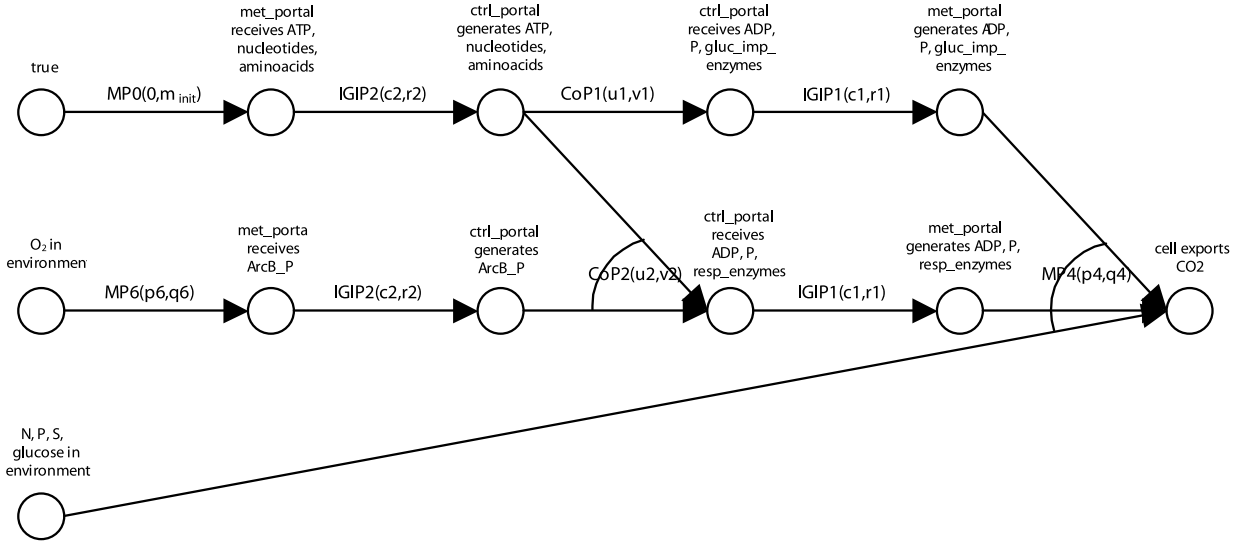

Fig. 6. Flow Diagram for Property CP1(glucose).

to be 0, so they could have been left out as well.

To explain the idea of the interlevel relationships (and in particular, the relations between the time durations involved) in some more detail, Fig. 6 depicts a Flow Diagram for property CP1(glucose). Nodes represent (conjunctions of) state properties. (Combinations of) edges represent dynamic properties. As can be seen in the picture, the cell is able to export CO_2 to the environment if the dynamic properties MP0, MP4(glucose), MP6, CoP1, CoP2, IGIP1 and IGIP2 hold. Figure 6 can be useful for understanding of the dependencies between the parameter values given above. Namely, when two processes occur in a sequence, the time duration of this sequence equals the sum of both individual time durations. For instance, the minimal duration assigned to the sequence of processes described by properties MP0(0, m_{init}) − IGIP2(c2, r2) is 0+c2, whilst the maximal duration is m_{init}+r2. Likewise, the time duration of the combination of two processes occurring in parallel equals the maximum of the individual time durations. In this case the processes have to be synchronised. Thus, if process C needs the simultaneous output of the parallel processes A and B as input, C can only start when both A and B have finished, under the assumption that the output substances of the processes persist long enough in order to co-occur at the same time instance.

Box 11. Interlevel relations for the dynamic properties of the cell as a whole

MP0(m_{init}) & MP4(p4, q4, glucose) & MP6(p6, q6)
& CoP1(u1, v1) & CoP2(u2, v2) & IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP1(d1, w1, glucose)
with d1 = max(0+c2+u1+c1, max(0+c2, p6+c2)+u2+c1)+p4,
w1 = max(m_{init}+r2+v1+r1, max(m_{init}+r2, q6+r2)+v2+r1)+q4.

MP0(m_{init}) & MP1(p1, q1) & MP2(p2, q2)
& MP4(p4, q4, glucose) & MP4(p4, q4, lactose) & MP6(p6, q6)
& CoP1(u1, v1) & CoP2(u2, v2) & CoP4(u4, v4)
& IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP1(d1, w1, lactose)
with d1glucose = max(0+c2+u1+c1, max(0+c2, p6+c2)+u2+c1)+p4,
w1glucose = max(m_{init}+r2+v1+r1, max(m_{init}+r2, q6+r2)+v2+r1)+q4,
d1lactose = max(max(0+c2, p6+c2)+u2+c1, max(0+c2, p1+c2, p2+c2)+u4+c1)+p4,
w1lactose = max(max(m_{init}+r2, q6+r2)+v2+r1, max(m_{init}+r2, q1+r2, q2+r2)+v4+r1)+q4,
d1 = min(d1glucose, d1lactose),
w1 = max(w1glucose, w1lactose).

MP0(m_{init}) & MP1(p1, q1) & MP3(p3, q3)
& MP4(p4, q4, glucose) & MP4(p4, q4, gluconate) & MP6(p6, q6)
& CoP1(u1, v1) & CoP2(u2, v2) & CoP5(u5, v5)
& IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP1(d1, w1, gluconate)
with d1glucose = max(0+c2+u1+c1, max(0+c2, p6+c2)+u2+c1)+p4,
w1glucose = max(m_{init}+r2+v1+r1, max(m_{init}+r2, q6+r2)+v2+r1)+q4,
d1gluconate = max(max(0+c2, p6+c2)+u2+c1, max(0+c2, p1+c2, p3+c2)+u5+c1)+p4,
w1gluconate = max(max(m_{init}+r2, q6+r2)+v2+r1, max(m_{init}+r2, q1+r2, q3+r2)+v5+r1)+q4,
d1 = min(d1glucose, d1gluconate),
w1 = max(w1glucose, w1gluconate).

MP0(m_{init}) & MP5(p5, q5, glucose) & CoP1(u1, v1) & CoP3(u3, v3)
& IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP2(d2, w2, glucose)
with d2 = 0+c2+max(u1+c1, u3+c1)+p5,
w2 = m_{init}+r2+max(v1+r1, v3+r1)+q5.

MP0(m_{init}) & MP1(p1, q1) & MP2(p2, q2)
& MP5(p5, q5, glucose) & MP5(p5, q5, lactose)
& CoP1(u1, v1) & CoP3(u3, v3) & CoP4(u4, v4)
& IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP2(d2, w2, lactose)
with d2glucose = 0+c2+max(u1+c1, u3+c1)+p5,
w2glucose = m_{init}+r2+max(v1+r1, v3+r1)+q5,
d2lactose = max(0+c2+u3+c1, max(0+c2, p1+c2, p2+c2)+u4+c1)+p5,
w2lactose = max(m_{init}+r2+v3+r1, max(m_{init}+r2, q1+r2, q2+r2)+v4+r1)+q5,
d2 = min(d2glucose, d2lactose),
w2 = max(w2glucose, w2lactose).

MP0(m_{init}) & MP1(p1, q1) & MP3(p3, q3)
& MP5(p5, q5, glucose) & MP5(p5, q5, gluconate)
& CoP1(u1, v1) & CoP3(u3, v3) & CoP5(u5, v5)
& IGIP1(c1, r1) & IGIP2(c2, r2) $\Rightarrow$ CP2(d2, w2, gluconate)
with d2glucose = 0+c2+max(u1+c1, u3+c1)+p5,
w2glucose = m_{init}+r2+max(v1+r1, v3+r1)+q5,
d2gluconate = max(0+c2+u3+c1, max(0+c2, p1+c2, p3+c2)+u5+c1)+p5,
w2gluconate = max(m_{init}+r2+v3+r1, max(m_{init}+r2, q1+r2, q3+r2)+v5+r1)+q5,
d2 = min(d2glucose, d2gluconate),
w2 = max(w2glucose, w2gluconate).

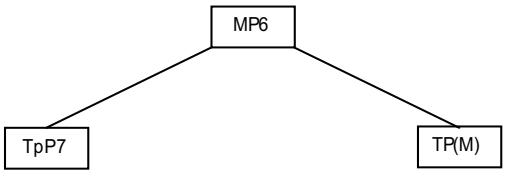

Fig. 7. Property MP6 related to role behaviour property and transfer properties.

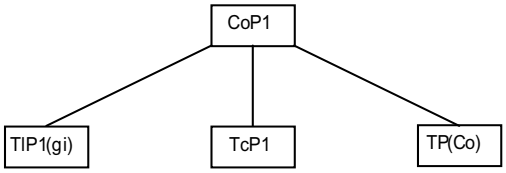

Fig. 8. Property CoP1 related to role behaviour properties and transfer properties.

4.2. Interlevel relations for group properties: Metabolism dynamics

As shown in Fig. 7, Metabolism group property MP6 is related to role behaviour property TpP7, together with the transfer properties of Metabolism, indicated by TP(M). A complete specification of the interlevel relations for all group properties of Metabolism is given in Box 12. Since all transfers are assumed to be instantaneous, the parameters of TP(M) have no influence.

4.3. Interlevel relations for group properties: Control dynamics

Figure 8 shows how Control group property CoP1 is related to role behaviour properties TlP1(glucose_import) and TcP1, together with all transfer properties of Control, indicated by TP(Co). A complete specification of all interlevel relations for the group properties of Control is given in Box 13.

Part III: Experiments and analysis

5. Simulation and verification

Describing the dynamics of complex biological processes in a formal language, as done in this paper, opens up the possibility to perform computer supported analysis, such as simulation and verification. For the Temporal Trace Language TTL introduced in Section 3.1, a number of tools have been developed to perform these kinds of tasks. Below, these tools are described in detail. Section 5.1 describes a simulation environment, Section 5.2 presents a tool to check dynamic properties against traces, and Section 5.3 addresses model checking. For each of these tools, the results of applying it to the *E. coli* case are shown.

5.1. Simulation

A software environment has been created to enable the *simulation* of executable organisation models specified at a high conceptual level [5]. The input of this simulation environment is a set of dynamic properties. Earlier, the language TTL was introduced as an expressive language for the purpose of specification and checking of dynamic properties. For the purpose of simulation, to obtain computational efficiency the format used for dynamic properties is more restricted than the TTL format used to specify various types of dynamic properties: they are in so-called LEADSTO format. This is a real time-valued variant of Executable Temporal Logic [3]. Roughly spoken, in LEADSTO format the following can be expressed [5]:

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

Making use of these LEADSTO properties, the software environment generates simulation traces. A trace is developed by starting at time $t = 0$ and for each time point up to which the trace already has been constructed, checking which antecedents of executable properties hold in the already constructed trace. For these executable properties, add the consequent to the trace, i.e., extend the trace in time in such a manner that the consequent holds.

The relation between the specification and the constructed trace is that the trace is a model (in the logical sense) of the theory defined by the specification, i.e., all executable dynamic LEADSTO properties of the specification hold in the trace (see also [5]).

The software environment described above has been used to simulate the internal dynamics of the organisation of the cell. In order to do this, all lowest level properties have been expressed in LEADSTO format. For this example, these were all intergroup role interaction properties, role behaviour properties and transfer properties. The specific timing parameter values assigned to the role behaviour properties, inspired by [18], are given in Table 1. For the other properties, all time parameters were 0.

In order to initialise the simulation, the truth values of all state properties have been set to *true* from time point

Box 12. Interlevel relations for the dynamic properties of the Metabolism group

TpP1(s1, t1) & TP(M) $\Rightarrow$ MP1(p1, q1)
with p1 = s1, q1 = t1.

TpP2(s2, t2) & TP(M) $\Rightarrow$ MP2(p2, q2)
with p2 = s2, q2 = t2.

TpP3(s3, t3) & TP(M) $\Rightarrow$ MP3(p3, q3)
with p3 = s3, q3 = t3.

TpP4(s4, t4) & TpP7(s7, t7) & TpP8(s8, t8, N)
& TpP8(s8, t8, P) & TpP8(s8, t8, S) & TpP9(s9, t9, CO_2)
& CaP0(c_{init}) & CaP1(i1, j1, glucose6P)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP4(p4, q4, glucose)
with p4 = max(0, s7, 0+s8, 0+s4)+i1+max(s9, m1),
q4 = max(c_{init}, t7, a_{init}+t8, a_{init}+t4)+j1+max(t9, n1).

TpP6(s6, t6) & TpP7(s7, t7) & TpP8(s8, t8, N)
& TpP8(s8, t8, P) & TpP8(s8, t8, S) & TpP9(s9, t9, CO_2)
& CaP0(c_{init}) & CaP1(i1, j1, lactose)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP4(p4, q4, lactose)
with p4 = max(0, s7, 0+s8, 0+s6)+i1+max(s9, m1),
q4 = max(c_{init}, t7, a_{init}+t8, a_{init}+t6)+j1+max(t9, n1).

TpP5(s5, t5) & TpP7(s7, t7) & TpP8(s8, t8, N)
& TpP8(s8, t8, P) & TpP8(s8, t8, S) & TpP9(s9, t9, CO_2)
& CaP0(c_{init}) & CaP1(i1, j1, gluconate6P)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP4(p4, q4, gluconate)
with p4 = max(0, s7, 0+s8, 0+s5)+i1+max(s9, m1),
q4 = max(c_{init}, t7, a_{init}+t8, a_{init}+t5)+j1+max(t9, n1).

TpP4(s4, t4) & TpP8(s8, t8, N) & TpP8(s8, t8, P)
& TpP8(s8, t8, S) & TpP9(s9, t9, acetate) & TpP9(s9, t9, ethanol)
& CaP0(c_{init}) & CaP2(i2, j2, glucose6P)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP5(p5, q5, glucose)
with p4 = max(0, 0+s8, 0+s4)+i2+max(s9, m1),
q4 = max(c_{init}, a_{init}+t8, a_{init}+t4)+j2+max(t9, n1).

TpP6(s6, t6) & TpP8(s8, t8, N) & TpP8(s8, t8, P)
& TpP8(s8, t8, S) & TpP9(s9, t9, acetate) & TpP9(s9, t9, ethanol)
& CaP0(c_{init}) & CaP2(i2, j2, lactose)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP5(p5, q5, lactose)
with p4 = max(0, 0+s8, 0+s6)+i2+max(s9, m1),
q4 = max(c_{init}, a_{init}+t8, a_{init}+t6)+j2+max(t9, n1).

TpP5(s5, t5) & TpP8(s8, t8, N) & TpP8(s8, t8, P)
& TpP8(s8, t8, S) & TpP9(s9, t9, acetate) & TpP9(s9, t9, ethanol)
& CaP0(c_{init}) & CaP2(i2, j2, gluconate6P)
& AP0(a_{init}) & AP1(m1, n1) & TP(M) $\Rightarrow$ MP5(p5, q5, gluconate)
with p4 = max(0, 0+s8, 0+s5)+i2+max(s9, m1),
q4 = max(c_{init}, a_{init}+t8, a_{init}+t5)+j2+max(t9, n1).

TpP7(s7, t7) & TP(M) $\Rightarrow$ MP6(p6, q6)
with p6 = s7, q6 = t7.

0 to 60. Furthermore, for each simulation run particular settings had to be assigned to the environment. An example simulation trace, where lactose and resources are always present, the presence of glucose and O_2 is fluctuating, and gluconate is always absent, can be seen in Fig. 9.

In this trace, time is on the horizontal axis, the properties are on the vertical axis. A dark box on top of the line indicates that the property is true during that time period, and a lighter box below the line indicates

Box 13. Interlevel relations for the dynamic properties of the Control group

TlP1(e1, f1, glucose_import) & TcP1(k1, l1) & TP(Co) ⇒ CoP1(u1, v1)
with u1 = k1+e1, v1 = l1+f1.

TlP2(e2, f2, respiration) & TcP2(k2, l2) & TP(Co) ⇒ CoP2(u2, v2)
with u2 = k2+e2, v2 = l2+f2.

TlP3(e3, f3, fermentation) & TcP3(k3, l2) & TP(Co) ⇒ CoP3(u3, v3)
with u3 = k3+e3, v3 = l3+f3.

TlP4(e4, f4, lactose_import) & TcP4(k4, l4) & TP(Co) ⇒ CoP4(u4, v4)
with u4 = k4+e4, v4 = l4+f4.

TlP5(e5, f5, gluconate_import) & TcP5(k5, l5) & TP(Co) ⇒ CoP5(u5, v5)
with u5 = k5+e5, v5 = l5+f5.

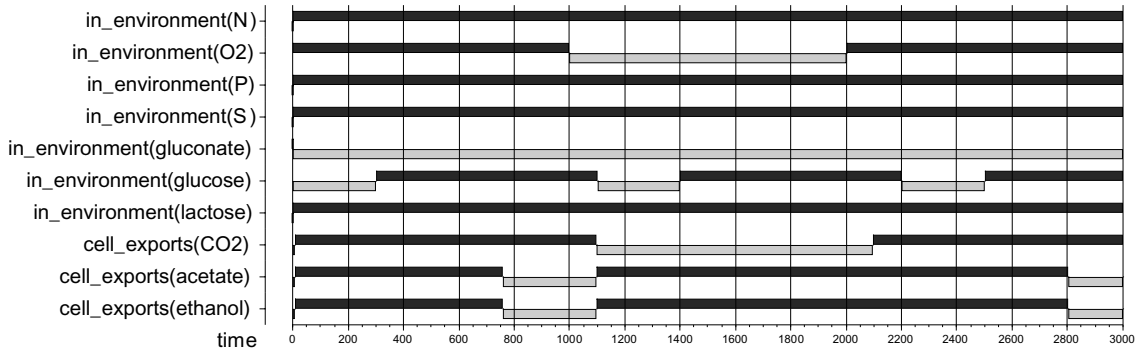

Fig. 9. Simulated overall behaviour.

Table 1
Time parameters for the LEADSTO properties

Property	Minimal delay (e)	Maximal delay (f)	Duration antecedent (g)	Duration consequent (h)
CaP1	4	12	4	80
CaP2	4	12	4	4
AP1	2	6	4	4
TpP1	−4	0	4	4
TpP2	0	0	0.23	0.23
TpP3	0	0	0.23	0.23
TpP4	−4	0	4	80
TpP5	−4	0	4	4
TpP6	−4	0	4	4
TpP7	0	0	4	4
TpP8	0	0	4	4
TpP9	0	0	4	4
TcP1	60	60	1	40
TcP2	60	60	1	40
TcP3	60	60	1	40
TcP4	60	60	1	40
TcP5	60	60	1	40
TlP1	0	0	10	600

that the property is false during that time period. The reaction of the cell within the environment can be seen in the last three lines. Notice that the cell exports acetate, ethanol and CO_2 at the very beginning, because of the initialisation conditions. However, as it adapts to the environment only CO_2 is exported. As the environmental oxygen disappears, the cell's CO_2 emissions stop very soon, and acetate and ethanol are produced instead. After the oxygen re-appears in the environment, the cell adapts by stopping the acetate and ethanol emissions after a while and returning to CO_2 production. Note that the acetate and ethanol emissions are not stopped immediately. This is because the internal substances needed for these emissions (including fermentation enzymes) persist for some time.

An interesting observation is the fact that the fluctuating presence of glucose in the environment does not seem to have any influence on the production of CO_2, acetate and ethanol. According to the highest level properties CP1 and CP2, this is indeed the correct behaviour, since for the behaviour at this level it does not matter whether it is glucose, lactose, or gluconate, as long as one of the nutrients is available. And in this particular case, lactose is always present in the environment. Nevertheless, the fluctuating presence of glucose does influence the behaviour of the cell at a lower level. For instance, consider the next part of the

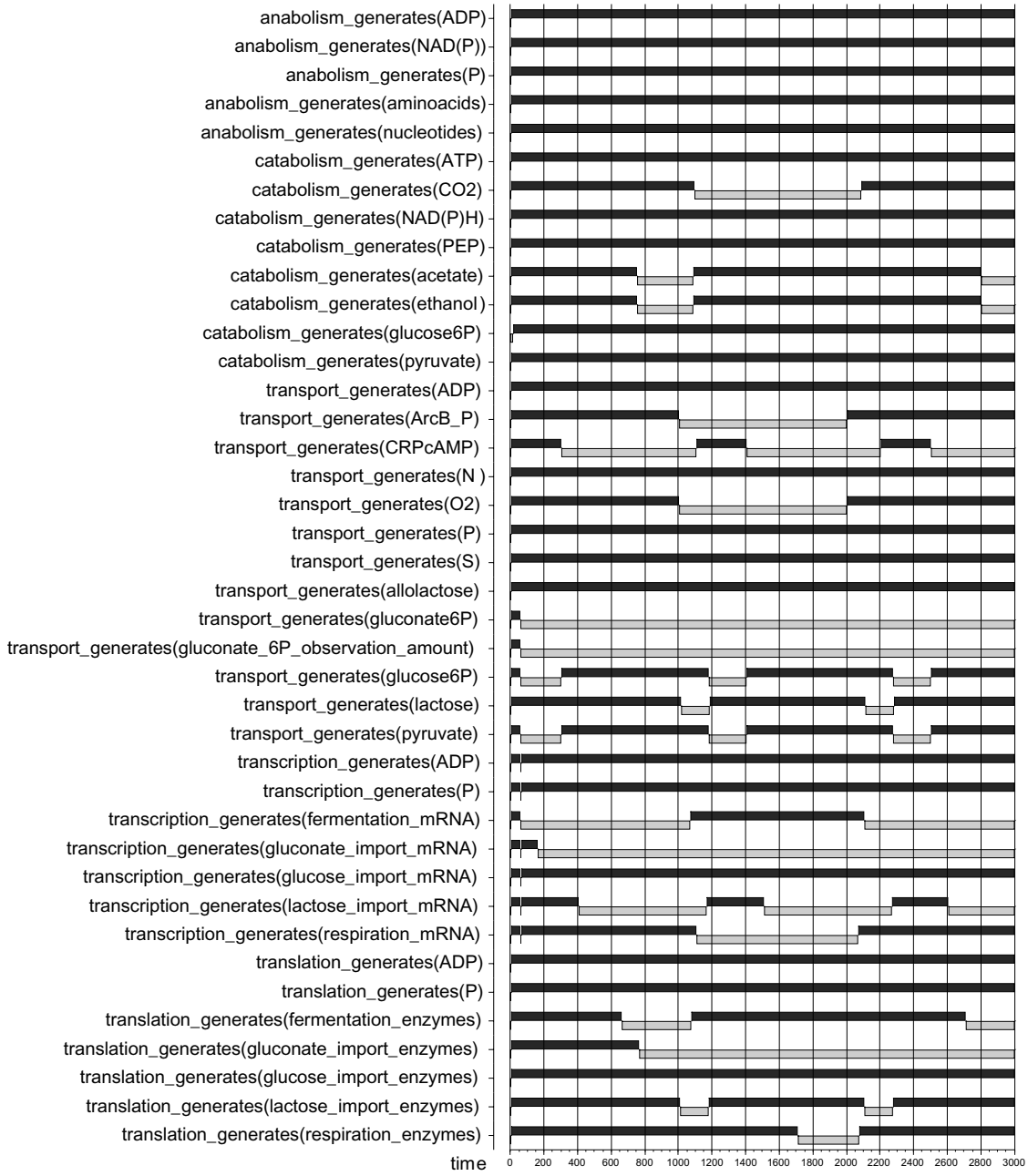

Fig. 10. Simulated internal dynamics.

same trace, depicting the output of the roles Anabolism, Catabolism, Transport, Transcription and Translation, see Fig. 10.

Figure 10 shows that the presence of glucose in the environment influences, for instance, the internal production of the substance CRPcAMP by the Transport role. As a consequence, the presence of (among others) this CRPcAMP leads to the creation of lactose_import_mRNA by the Transcription role, whilst glucose_import_mRNA is created continuously. To go one step further, lactose_import_mRNA and glucose_import_mRNA are used by the Translation role to create, with a certain delay, lactose_import_enzymes and glucose_import_enzymes. It can thus be concluded that from an external perspective there is no visible difference in behaviour of the cell, whether there is only

lactose outside or both lactose and glucose. Nevertheless, from an internal perspective many differences can be seen. The entire trace resulting from this simulation covers 245 state properties, representing not only the output but also the input state properties of the roles shown above. However, since the transfer of substances is instantaneous and without delay in our model, each output state property for one role results in several identical input state properties for the other roles. Likewise, the input and output state properties of the Metabolism Portal and Control Portal group are identical to state properties already shown above. Hence, for reasons of presentation, the rest of the trace is not shown in this paper.

5.2. Checking dynamic properties

The simulation software described above automatically produces log files, containing formal representations of the traces. In addition, a tool has been developed to *automatically check* whether certain high-level properties hold for given (empirical or simulated) traces. This tool is able to read in these formally represented traces together with a set of dynamic properties, and verifies the dynamic properties against the traces. As a result, the tool determines not only whether a property holds for a trace or not, but in case of failure, it also pinpoints which parts of the trace violate the property. For our simulation, checks of this kind have actually been performed for all Global Properties and Group Properties, i.e. all properties of Sections 3.1 and 3.3. They all turned out to hold for the generated traces.

By combining this checker tool with the interlevel relationships between dynamic properties, for example as depicted in Fig. 4, it can be used for *diagnosis of dysfunctioning* within an organisation. For example, suppose for a given trace at some point in time it has been detected (using the checker tool) that the dynamic property CP1(glucose) at the highest aggregation level of the organisation does not hold, i.e., the cell does not produce CO_2 although the substances O_2, glucose, N, P and S are available within the environment. Given the AND-tree structure in Fig. 4, at least one of the children will not hold (if they all would hold for the given trace, also CP1(glucose) would hold for this trace), which means that either MP0, MP6, IGIP2, CoP1, CoP2, IGIP1, or MP4(glucose) will not hold. Suppose by further checking it is found that MP6 does not hold. Then the diagnostic process can be continued by focusing on this property. It follows that either TpP7 or TP(M) does not hold (see Fig. 7). Checking these two properties will pinpoint the cause of the organisation's dysfunctioning. Notice that this diagnostic process is economic in the sense that the whole subtree under e.g. CoP1 is not examined since there is no reason for that, as CoP1 holds.

5.3. Checking interlevel relations

The diagnosis of dysfunctioning mentioned above is only feasible under the assumption that the tree of interlevel relationship is correct (i.e., that each dynamic property at a certain level of aggregation is indeed logically entailed by the dynamic properties at a lower level). To automatically check this logical entailment for given interlevel relations, model checking techniques can be used. For checking interlevel relations between such properties, the model checker SMV is appropriate (http://www.cs.cmu.edu/~modelcheck/smv.html; see also McMillan, 1993). To this end, translations have to be made from properties expressed in TTL to a specific input format for SMV. In particular, in case a number of lower level properties LPi logically entail a higher level property GP, it is required that the LPi have the format of transition rules, whereas the GP should be represented in CTL format (cf. Goldblatt, 1992). For the interlevel relation shown in Fig. 4, these translations have indeed been made, and the interlevel relation has been checked successfully. To give an impression, the result of translating global property CP1 (see also Section 3.1) to CTL looks as follows (assuming that the environment should be stable for at least five time units:

```
AG ((in_environment_glucose & in_environment
_O2 & in_environment_N & in_environment_P &
in_environment_S) →
(AX ((in_environment_glucose & in_environment _O2 &
in_environment_N & in_environment_P & in_ environment_S)
→
(AX ((in_environment_glucose & in_environment _O2 &
in_environment_N & in_environment_P & in_ environment_S)
→
(AX ((in_environment_glucose & in_environment _O2 &
in_environment_N & in_environment_P & in_ environment_S)
→
(AX ((in_environment_glucose & in_environment _O2 &
in_environment_N & in_environment_P & in_ environment_S)
→
AF cell_exports_CO2)))))))))
```

The complete SMV specification for the interlevel relation shown in Fig. 4 can be found in Appendix A.

Part IV: Related work and conclusion

6. Generalisation to other biological domains

In the previous sections, the organisation modelling approach based on AGR has been illustrated by applying it to the case of *E. Coli*. In addition, in the current section it will be shown to what extent the approach can be generalised to other biological domains. To this end, we refer to the work in [6]. In that paper, the organisation of the circulatory system in mammals was addressed using the modelling approach based on AGR. Some details about that paper are given in Section 6.1. Next, in Section 6.2 some commonalities are shown between that case study and the case study on the living cell reported in this article, thereby obtaining a unifying perspective on addressing biocomplexity. Section 6.3 points out some differences between both case studies.

6.1. Another case study: The circulatory system in mammals

The circulatory system takes care of a number of capacities, such as providing nutrients and oxygen to the body and taking waste (e.g., CO_2) out of the body. The main property to focus on in this example is that the system provides oxygen for all parts of the body. The organisation of the circulatory system S is analysed as consisting of the following active components that (by showing their specific behaviours) all play their roles within the overall process: heart, capillaries in lungs and other organs, arteries (pulmonary artery channels, from the heart to the capillaries in the lungs; aorta channels, from heart to the capillaries in the body), veins (pulmonary veins, from the capillaries in the lungs to the heart; inferior and superior vena cava, from the capillaries in the body to the heart).

In [6], the circulatory system is modelled from an organisational perspective, according to the approach based on AGR. Following this approach, at the top level the system can be seen as one component. At a lower aggregation level, properties of *groups* have been identified, as well as properties of *inter-group transfers*. The lowest level comprises properties of *roles* and *transfers* between them. See Fig. 11: at the top level, the circulatory system can be seen as one organisation, which consists of two groups at a lower level, i.e., a *Pulmonary Cycle Group* and a *Systemic Cycle Group*. The main function of the Pulmonary Cycle Group is uptake of oxygen from the environment through the lungs, and the main function of the

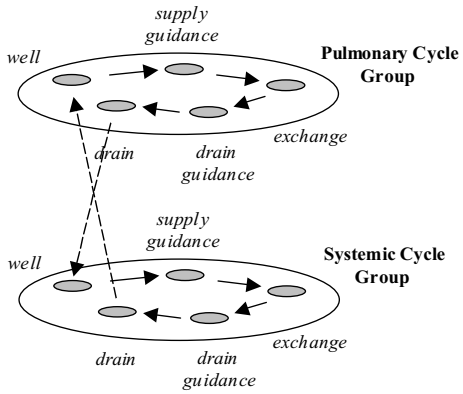

Fig. 11. The circulatory system: groups and interactions.

Systemic Cycle Group is to supply this oxygen to the other organs. At the lowest level, each group consists of a number of roles with transfers between them. Note that both groups are organised according to a similar structure, consisting of the following five roles: *well*, *supply guidance*, *exchange*, *drain guidance*, *drain*.

Moreover, to each role a certain active component (or agent) can be allocated. To be specific, for the Systemic Cycle Group, the allocation of agents to roles is as follows:

heart	– systemic cycle well
aorta channels	– systemic cycle supply guidance
organ capillaries	– systemic cycle exchange
inferior and superior vena cava	– systemic cycle drain guidance
heart	– systemic cycle drain

For the pulmonary cycle group instance the allocation of agents to roles is as follows:

heart	– pulmonary cycle well
pulmonary channels	– pulmonary cycle supply guidance
lung capillaries	– pulmonary cycle exchange
pulmonary veins	– pulmonary cycle drain guidance
heart	– pulmonary cycle drain

Note that in both groups, the heart plays two roles, one of a well, initiating the flow, and one of a drain, where the flow disappears (and will re-appear in the other side).

Like for the case of *E. Coli*, also for the case of the circulatory system a large number of dynamic properties have been specified (both in informal and formal format), and simulation and verification have been performed. Thus, also in this case the AGR-based modelling approach turned out useful to address the complexity of a biological system from an organisation perspective. For more details about the model and further references, see [6].

6.2. Commonalities between both Case Studies

The two biological case studies, the circulatory system and the living cell, have some aspects in common and differ in some other aspects. A main common aspect is that in both cases Nature shows a certain form of organisation. Although the areas are quite distinct, the organisation modelling approach illustrates this common aspect by using generic concepts such as roles and groups to model both example processes. This is a main contribution of this paper: to show that the notion of organisation as observed in Nature, can be addressed and formalised from a generic perspective. Thus a unifying perspective is obtained on the way how Nature copes with (and develops) complexity by exploiting (increasing degrees of) organisation. Organisation modelling techniques as put forward in this paper provide means to describe, compare and distinguish the different forms and principles of organisation possible and/or occurring in Nature, thus providing a structuring of the variety of biodiversity and biocomplexity from the perspective of underlying organisational principles.

Although the techniques proposed in this paper can be applied to different biological domains, this does not mean that no domain knowledge is required to model a specific case study. On the contrary, a rather detailed description of the process under consideration should be known beforehand. Therefore, when modelling a specific case study, ideally there is intensive interaction between the modeller and a domain expert with a background in the concerned specialisation in biology. Consequently, the main aim of the modelling approach is not to support some initial analysis of a certain biological process, but rather to deepen the understanding of an already (partially) known process, in particular by imposing organisational structure upon it. Thus, the main contributions of the approach are comparable to the usual benefits of formal modelling and simulation techniques: it makes explicit many details that are initially lacking or are only described partially and/or informally, thereby enabling the modeller to refine (and possibly improve) the initial theory. In fact, in both of the case studies described in this paper, the process of formalisation has led to many of such refinements. For example, in an initial phase of the cell case study [18], the temporal dependencies of a number of processes involved had to be specified explicitly (see also Table 1).

6.3. Differences between both Case Studies

The two examples addressed – circulatory system and intracellular processes – also illustrate differences. In the circulatory system, modelling the organisation structure is in some sense 'hard-wired' in physical reality. Arteries and veins are physically connected to heart, lungs and other organs, and each of the organs has a specific location, which is non-overlapping with locations of other organs. The functioning of this organisation is forced by this physical configuration. For example, if an artery is cut off or a vein is decoupled from the heart, then the entailed dysfunctioning of the organisation usually is lethal for the organism. In contrast to this 'hard-wired' case, the living cell example shows a kind of opposite situation. Here, all processes are assumed to occupy the same spatial area. No fixed physical separations and connections between the various processes exist (as would be the case in an installation in a chemical factory), except that all substances are kept together within the cell by the membrane (the soup metaphor). Escape is only possible in some cases, which are often controlled by the cell. Within the cell, free mobility is assumed for all substances involved. In this case the functioning of the organisation emerges from the possibilities for the ways in which the various chemical processes can interact with each other.

One might expect that an organisation modelling approach would only apply in the hard-wired circulatory system case. However, as is shown in this paper, also in the free mobility living cell case, the organisation modelling approach can be useful. Hence, not only a physically forced structure can be used and further analysed as an organisation structure, but also an organisational structure that emerges out of a number possibilities for interaction between processes can be successfully analysed.

7. Other related work

Analysis and simulation of biological (and in particular, cellular) processes is a huge research area in which many groups are working. Some of these groups focus on realistic simulation of cellular processes using object-oriented software (e.g. [36]), whereas others address agent-based modelling of cellular processes (e.g. [7,16]). Although the current paper is also based on the idea of applying the agent paradigm to biological modelling, as a novel contribution to the area it explores how recently developed organisation modelling approaches such as AGR [11,20] can be used to analyse and simulate the dynamics of complex biological processes.

As shown by the case studies, the idea of analysing the organisational structure of biological processes turned out useful to obtain more detailed descriptions of these processes. Nevertheless, it is not claimed that AGR is the only possible approach that can be used for this purpose. In computational organisation theory and artificial intelligence, other approaches have been developed that are able to capture both structural and dynamic aspects of organisations. However, usually they describe organisation models, using only two or three levels of abstraction (i.e., the level of an individual role, the level of a group composed of roles, and the overall organisation level, e.g., GAIA [39], MOISE [17], MOCA [1], and OperA [8]), whereas in complex biological domains, it may be desirable to describe multiple abstraction levels. For example, in the cell case study, the process Catabolism may be further decomposed into the sub-processes Glycolysis, Pyruvate Catabolism and Glycogen Catabolism, according to [18]. A feature that distinguishes our approach from other organisation modelling techniques is the fact that it allows the modeller to specify the dynamics of the organisation in a detailed way, using the TTL language.

The methodology presented in this paper is supported by a number of software tools. For example, an editor to specify dynamic properties according to the TTL format [4], a tool for simulation of executable LEADSTO models [5,22], a model checker that verifies interlevel relations between dynamic properties at different aggregation levels [28], and a verification tool that checks whether dynamic properties hold in a given trace [4]. Obviously, for each of these tools various alternatives exist in the literature. For example, other approaches for simulation are the Dynamical Systems Theory [33], Executable Temporal Logic [3], PLC automata [9], and qualitative reasoning (e.g. [14]). For verification of properties, alternative approaches are modal-logic-based temporal languages such as LTL and CTL [15], and calculi like the situation calculus [35] and the event calculus [24]. A main advantage of our modelling tools over most of these approaches is that they enable the modeller to specify dynamics in terms of both qualitative and quantitative aspects, thereby combining the benefits of logic-oriented modelling approaches with the benefits of mathematical approaches. Moreover, our tools are able to deal with real-valued time parameters. As explained in Section 5.1, rules in LEADSTO format include four time parameters e, f, g, h, that indicate, respectively, the minimal delay, the maximal delay, the duration of the antecedent, and the duration of the consequent. Unlike in many other modelling approaches, these parameters may be real numbers, which results in more realistic simulations of biological processes. For this case study addressed, several discussions were held with experts in the domain, in order to define specific time parameters for the LEADSTO rules. As a consequence, the resulting simulation traces closely match the corresponding real world processes. Furthermore, a specific advantage of TTL is its high expressiveness. Since TTL is an extension of predicate logic, all dynamic properties that can be expressed in first-order predicate logic can also be expressed in TTL. In addition, TTL allows explicit references to different temporally ordered sequences of states (traces) in dynamic properties, which makes it possible to express properties that compare different traces with each other, such as the statement "exercise improves skill". This goes beyond the expressiveness of standard temporal languages [15] or calculi [35], in which only a single path of situations can be explicitly encoded in the formulae. For a more extensive comparison of our modelling tools (in particular, LEADSTO and TTL) with other simulation and verification approaches that exist in the literature (including issues related to expressiveness and complexity), see [4,5].

Finally, a specific software environment that is worth mentioning is Simpathica/XSSYS [30]. Although this environment is not aimed at modelling biological systems from an organisational perspective, it uses a verification tool that is similar to our TTL checker tool: in Simpathica/XSSYS, also dynamic properties about biological processes can be specified in a temporal logic (in this case CTL), and checked against a (limited) number of (empirical or simulated) traces. Moreover, Simpathica/XSSYS features a "sentence generation tool", which takes a number of traces as input and generates some new properties that are satisfied by these traces.

8. Conclusion and future work

This article explores how recently developed organisation modelling approaches such as AGR [11,20] can be used to analyse and simulate the dynamics of complex biological processes. This modelling perspective identifies organisational structure occurring in complex decentralised processes and handles complexity of the analysis of the dynamics by structuring these dynamics according to an organisational structure. More specifically, a methodology has been proposed that involves the following ingredients:

- Specify state properties and dynamic properties of the overall process
- Identify the agents and their roles within the overall process
- Specify state properties and dynamic properties for the behaviour of these roles
- Identify groups of roles
- Specify dynamic properties for groups
- Specify dynamic intergroup role interaction properties and transfer properties between roles
- Identify interlevel relations between dynamic properties at different levels of aggregation: relating role, group and organisation dynamics
- Relate state properties to physical or chemical state properties
- Relate dynamic properties to physical or chemical dynamic properties
- Specify executable dynamic properties
- Simulate dynamics based on executable dynamic properties
- Check given traces of dynamics against dynamic properties
- Verify the interlevel relations between dynamic properties at different levels of aggregation

For most of these items, software tools have been developed. Using these tools, the methodology has been illustrated for two case studies: the functioning of intracellular processes and the functioning of the circulatory system (although the details of the latter example have been left out). These biological systems can be modelled as consisting of a number of active components or agents that are connected and grouped together in such a manner that everything functions well. For both case studies, dynamic properties at different levels of aggregation of the organisation model have been identified, and relationships between these dynamic properties at different aggregation levels were made explicit. Based on the executable properties, simulation has been performed and (higher-level) properties have been checked for the produced simulation traces. Thus it was verified that the simulation traces satisfied some expected global properties. Moreover, the interlevel relationships between properties at different aggregation levels have been verified automatically using model checking techniques. These case studies show that organisation modelling techniques can play a useful role in biological application areas.

As mentioned earlier, the approach presented in this paper does not explicitly model the environment as part of the organisation. Instead, interaction of roles with the environment is modelled within the dynamic properties of the roles. However, in some cases it may be desirable to consider the environment as a special component of the organisation model. Therefore, since a number of years, several approaches recognise the importance of explicit modelling of interactions between agents and the environment (e.g. [8]). Recently, also for the AGR approach an extension was proposed (called AGRE), in which (social and physical) environments can be included [13]. Future work will focus on modelling the current examples in more detail using AGRE.

Another potential direction for future work is to explore the possibilities for combining our methodology with the Gene Ontology (GO) project [2]. This project aims at the development of ontologies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. By using terms from GO as the state ontology for the specification of our biological models, the terminology used in our work could be made more consistent with the standard terminology used in the literature. For example, for the *E. Coli* case, this could lead to a modification of the concepts used in Fig. 1. Moreover, comparing the structure of GO with that of our biological models could play a role in improving these models, for example, by pointing at missing elements.

Acknowledgements

The authors have learned a lot of this area from discussions and cooperation with Jaap Heringa, Jacky Snoep, Hans Westerhoff, and Wouter Wijngaards. Moreover, this article has been improved thanks to the comments of anonymous referees.

References

[1] M. Amiguet, J.P. Mueller, J.A. Baez-Barranco and A. Nagy, The MOCA Platform, in: *Proceedings of the Third International Workshop on Multi-Agent Based Simulation, MABS'02*, 2002, 70–88.

[2] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig et al., Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nature Genet* **25** (2000), 25–29.

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

[4] T. Bosse, C.M. Jonker, L. van der Meij, A. Sharpanskykh and J. Treur, Specification and Verification of Dynamics in Cognitive Agent Models, in: *Proceedings of the Sixth International Conference on Intelligent Agent Technology, IAT'06*, T. Nishida et al., eds, IEEE Computer Society Press, 2006, pp. 247–254. Report version URL: http://www.cs.vu.nl/~tbosse/TTL.

[5] T. Bosse, C.M. Jonker, L. van der Meij and J. Treur, LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn, in: *Proceedings of the Third German Conference on Multi-Agent System Technologies, MATES'05*, T. Eymann et al., eds, Lecture Notes in AI, vol. 3550. Springer Verlag, 2005, pp. 165–178. Extended version to appear in Journal of Artificial Intelligence Tools.

[6] T. Bosse, C.M. Jonker and J. Treur, Organisation Modelling for the Dynamics of Complex Biological Processes, in: *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications, RASTA'02*, G. Lindemann, D. Moldt, M. Paolucci and B. Yu, eds, Lecture Notes in AI, vol. 2934. Springer Verlag, 2004, pp. 92–112.

[7] F. Corradini, E. Merelli and M. Vita, A Multi-agent System for Modelling Carbohydrate Oxidation in Cell, in: *Proceedings of the International Conference on Computational Science and its Applications, ICCSA'05*, O. Gervasi et al., eds, Lecture Notes in Computer Science, vol. 3481. Springer Verlag, 2005, pp. 1264–1273.

[8] M. Dastani, J. Hulstijn, F. Dignum and J.-J. Meyer, Issues in Multiagent System Development. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS'04*. ACM Press, 2004, 922–929.

[9] H. Dierks, PLC-automata: A new class of implementable real-time automata, in: *Transformation-Based Reactive Systems Development* (*ARTS'97*), M. Bertran and T. Rus, eds, Lecture Notes in Computer Science, vol. 1231, Springer-Verlag, 1997, pp. 111–125.

[10] J. Ferber, *Multiagent Systems*, Addison Wesley, 1999.

[11] J. Ferber and O. Gutknecht, A meta-model for the analysis and design of organisations in multi-agent systems. In: *Proceedings of the Third International Conference on Multi-Agent Systems* (ICMAS'98), IEEE Computer Society Press, 1998, 128–135.

[12] J. Ferber and O. Gutknecht, Operational Semantics of a role-based agent architecture, in: *Intelligent Agents VI*, N.R. Jennings and Y. Lesperance, eds, Lecture Notes in AI, vol. 1757, Springer Verlag, 2000, pp. 205–217.

[13] J. Ferber, F. Michel and J.A. Baez-Barranco, AGRE: Integrating Environments with Organizations, in: *Proceedings of the First International Workshop on Environments for Multi-Agent Systems, E4MAS'04*, D. Weyns, H. Van Dyke Parunak and F. Michel, eds, Lecture Notes in Computer Science, vol. 3374. Springer Verlag, 2005, pp. 48–56.

[14] K.D. Forbus, Qualitative process theory, *Artificial Intelligence* **24** (1984), 85–168.

[15] R. Goldblatt, *Logics of Time and Computation*, 2nd edition, CSLI Lecture Notes 7, 1992.

[16] P. Gonzalez, M. Cardanas, A.D. Camacho, O. Rosas and J.L. Otero, Cellulat: an agent-based intracellular signalling model, *BioSystem* (2003), 171–185.

[17] J.F. Hubner, J.S. Sichman and O. Boissier, Moise+: towards a structural, functional and deontic model for MAS organization. In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'02*, ACM Press, 2002, 501–502.

[18] C.M. Jonker, J.L. Snoep, J. Treur, H.V. Westerhoff and W.C.A. Wijngaards, The Living Cell as an Organisation: A Compositional Organisation Model of Intracellular Dynamics, in: *Agent-Based Modelling of Dynamics: Biological and Organisational Applications*, W.C.A. Wijngaards, Ph.D. Thesis. Vrije Universiteit Amsterdam, Department of Artificial Intelligence, 2002, pp. 191–254.

[19] C.M. Jonker and J. Treur, Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. *International Journal of Cooperative Information Systems* **11** (2002), 51–92. Earlier, shorter version in: *Proceedings of the International Workshop on Compositionality, COMPOS'97*, W.P. de Roever, H. Langmaack and A. Pnueli, eds, Lecture Notes in Computer Science, vol. 1536, Springer Verlag, 1998, pp. 350–380.

[20] C.M. Jonker and J. Treur, Relating Structure and Dynamics in an Organisation Model, in: *Multi-Agent-Based Simulation II, Proc. of the Third International Workshop on Multi-Agent Based Simulation, MABS'02*, J.S. Sichman, F. Bousquet and P. Davidson, eds, Lecture Notes in AI, vol. 2581, Springer Verlag, 2003, pp. 50–69.

[21] C.M. Jonker and J. Treur, Agent-Oriented Modeling of the Dynamics of Biological Organisms, *Journal of Applied Intelligence*. To appear, 2007.

[22] C.M. Jonker, J. Treur and W.C.A. Wijngaards, Temporal Languages for Simulation and Analysis of the Dynamics Within an Organisation, in: *Proceedings of the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems, CEEMAS'01*, B. Dunin-Keplicz and E. Nawarecki, eds, 2001. Lecture Notes in AI, vol. 2296, Springer Verlag, 2002, pp. 151–160. Extended version to appear in Journal of Applied Intelligence.

[23] H. Kitano, Systems biology: a brief overview, *Science* **295** (2002), 1662–1664.

[24] R. Kowalski and M.A. Sergot, A logic-based calculus of events, *New Generation Computing* **4** (1986), 67–95.

[25] R. Kreitner and A. Kunicki, *Organisational Behavior*, McGraw-Hill, 2001.

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

[27] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Safety*, Springer Verlag, 1995.

[28] K.L. McMillan, Symbolic Model Checking: An Approach to the State Explosion Problem. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1992. Published by Kluwer Academic Publishers, 1993.

[29] H. Mintzberg, *The Structuring of Organisations*, Prentice Hall, Englewood Cliffs, NJ, 1979.

[30] B. Mishra, M. Antoniotti, S. Paxia and N. Ugel, Simpathica: a computational systems biology tool within the valis bioinformatics environment. In: *Computational Systems Biology*. Elsevier, Amsterdam, the Netherlands, 2005.

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

[32] F.C. Neidhardt, R. Curtiss III, J.L. Ingraham, E.C.C. Lin, K. Brooks Low, B. Magasanik, W.S. Reznikoff, M. Riley, M. Schaechter and H.E. Umbarger, eds, *Escherichia coli* and *Salmonella typhimurium*. ASM Press, Washington, DC, 1996.

[33] R.F. Port and T. van Gelder, eds, *Mind as Motion: Explorations in the Dynamics of Cognition*, MIT Press, Cambridge, Mass, 1995.

[34] M. Prietula, L. Gasser and K. Carley, *Simulating Organizations*, MIT Press, 1997.

[35] R. Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, Cambridge MA: MIT Press, 2001.
[36] K. Takahashi, *E-Cell: A Multi-Algorithm, Multi-Timescale Simulation Software Environment*, 2004. URL: http://ecell.sourceforge.net/.
[37] K. Webb and T. White, Cell modeling using agent-based formalisms, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'04*, N.R. Jennings, C. Sierra, L. Sonenberg and M. Tambe, eds, ACM Press, 2004, pp. 1190–1196.
[38] G. Weiss, ed., *Multiagent Systems*, MIT Press, 1999.
[39] F. Zambonelli, N.R. Jennings and M. Wooldridge, Developing multiagent systems: the Gaia Methodology, *ACM Trans on Software Engineering and Methodology* **12** (2003), 317–370.

Appendix A

SMV specification for the interlevel relation MP0 & MP4(g) & MP6 & CoP1 & CoP2 & IGIP1 & IGIP2 ⇒ CP1(g).

```
MODULE main
 VAR
in_environment_glucose: boolean;
in_environment_O2: boolean;
in_environment_N: boolean;
in_environment_P: boolean;
in_environment_S: boolean;
cell_exports_CO2: boolean;
metabolism_portal_generates_ADP: boolean;
metabolism_portal_generates_P: boolean;
metabolism_portal_generates_respiration_enzymes:
boolean;
metabolism_portal_generates_glucose_import_enzymes:
boolean;
metabolism_portal_receives_ATP: boolean;
metabolism_portal_receives_nucleotides: boolean;
metabolism_portal_receives_aminoacids: boolean;
metabolism_portal_receives_ArcB_P: boolean;
control_portal_generates_ATP: boolean;
control_portal_generates_nucleotides: boolean;
control_portal_generates_aminoacids: boolean;
control_portal_generates_ArcB_P: boolean;
control_portal_receives_ADP: boolean;
control_portal_receives_P: boolean;
control_portal_receives_respiration_enzymes: boolean;
control_portal_receives_glucose_import_enzymes: boolean;

ASSIGN
init(in_environment_glucose):=0;
init(in_environment_O2):=0;
init(in_environment_N):=0;
init(in_environment_P):=0;
init(in_environment_S):=0;
init(cell_exports_CO2):=0;
init(metabolism_portal_generates_ADP):=0;
init(metabolism_portal_generates_P):=0;
init(metabolism_portal_generates_respiration_enzymes):=0;
init(metabolism_portal_generates_glucose_import_enzymes):
=0;
init(metabolism_portal_receives_ATP):=1;
init(metabolism_portal_receives_nucleotides):=1;
init(metabolism_portal_receives_aminoacids):=1;
init(metabolism_portal_receives_ArcB_P):=0;
init(control_portal_generates_ATP):=0;
init(control_portal_generates_nucleotides):=0;
init(control_portal_generates_aminoacids):=0;
init(control_portal_generates_ArcB_P):=0;
init(control_portal_receives_ADP):=0;
init(control_portal_receives_P):=0;
init(control_portal_receives_respiration_enzymes):=0;
init(control_portal_receives_glucose_import_enzymes):=0;

next(metabolism_portal_receives_ATP):= case
 in_environment_glucose & in_environment_O2 & in_
environment_N &
 in_environment_P & in_environment_S & metabolism_portal
_generates_ADP &
 metabolism_portal_generates_P & metabolism_portal
_generates_respiration_enzymes &
 metabolism_portal_generates_glucose_import_enzymes: 1;
 metabolism_portal_receives_ATP: 1;
 1: 0;
esac;
next(metabolism_portal_receives_nucleotides):= case
 in_environment_glucose & in_environment_O2 & in_
environment_N &
 in_environment_P & in_environment_S & metabolism_portal
_generates_ADP &
 metabolism_portal_generates_P & metabolism_portal
_generates_respiration_enzymes &
 metabolism_portal_generates_glucose_import_enzymes: 1;
 metabolism_portal_receives_nucleotides: 1;
 1: 0;
esac;
next(metabolism_portal_receives_aminoacids):= case
 in_environment_glucose & in_environment_O2 & in_
environment_N &
 in_environment_P & in_environment_S & metabolism_portal
_generates_ADP &
 metabolism_portal_generates_P & metabolism_portal
_generates_respiration_enzymes &
 metabolism_portal_generates_glucose_import_enzymes: 1;
 metabolism_portal_receives_aminoacids: 1;
 1: 0;
esac;
next(cell_exports_CO2):= case
 in_environment_glucose & in_environment_O2 & in_
environment_N &
 in_environment_P & in_environment_S & metabolism_portal
_generates_ADP &
 metabolism_portal_generates_P & metabolism_portal
_generates_respiration_enzymes &
 metabolism_portal_generates_glucose_import_enzymes: 1;
 1: 0;
esac;

next(metabolism_portal_receives_ArcB_P):= case
 in_environment_O2: 1;
 1: 0;
esac;

next(control_portal_receives_ADP):= case
 control_portal_generates_ATP & control_portal_generates
_nucleotides &
 control_portal_generates_aminoacids: 1;
 1: 0;
```

```
    esac;
    next(control_portal_receives_P):= case
     control_portal_generates_ATP & control_portal_generates
_nucleotides &
     control_portal_generates_aminoacids: 1;
     1: 0;
    esac;
    next(control_portal_receives_glucose_import_enzymes):=
case
     control_portal_generates_ATP & control_portal_generates
_nucleotides &
     control_portal_generates_aminoacids: 1;
     1: 0;
    esac;

    next(control_portal_receives_respiration_enzymes):= case
     control_portal_generates_ATP & control_portal_generates
_nucleotides &
     control_portal_generates_aminoacids & control_portal
_generates_ArcB_P: 1;
     1: 0;
    esac;

    next(metabolism_portal_generates_ADP):= case
     control_portal_receives_ADP: 1;
     1: 0;
    esac;
    next(metabolism_portal_generates_P):= case
     control_portal_receives_P: 1;
     1: 0;
    esac;
    next(metabolism_portal_generates_respiration_enzymes):=
case
     control_portal_receives_respiration_enzymes: 1;
     1: 0;
    esac;
    next(metabolism_portal_generates_glucose_import
_enzymes):= case
     control_portal_receives_glucose_import_enzymes: 1;
     1: 0;
    esac;

    next(control_portal_generates_ATP):= case
     metabolism_portal_receives_ATP: 1;
     1: 0;
    esac;
    next(control_portal_generates_nucleotides):= case
     metabolism_portal_receives_nucleotides: 1;
     1: 0;
    esac;
    next(control_portal_generates_aminoacids):= case
     metabolism_portal_receives_aminoacids: 1;
     1: 0;
    esac;
    next(control_portal_generates_ArcB_P):= case
     metabolism_portal_receives_ArcB_P: 1;
     1: 0;
    esac;

    SPEC

    AG ((in_environment_glucose & in_environment_O2 & in_
environment_N & in_environment_P & in_environment_S) ->
      (AX ((in_environment_glucose & in_environment_O2 & in_
environment_N & in_environment_P & in_environment_S) ->
      (AX ((in_environment_glucose & in_environment_O2 & in_
environment_N & in_environment_P & in_environment_S) ->
      (AX ((in_environment_glucose & in_environment_O2 & in_
environment_N & in_environment_P & in_environment_S) ->
      (AX ((in_environment_glucose & in_environment_O2 & in_
environment_N & in_environment_P & in_environment_S) ->
      AF cell_exports_CO2)))))))))
```

Authors' Bios

Tibor Bosse is an Assistant Professor at the Department of Artificial Intelligence at the Vrije Universiteit Amsterdam. He obtained his Ph.D. in 2005 at the same university, graduating on a project that addressed the analysis of the dynamics of cognitive processes. Currently, his research activities are focussed on modelling dynamics of agent systems in various disciplines, in particular Biology, Cognitive Science, and Criminology.

Catholijn Jonker received her Ph.D. degree in Computer Science in 1994 from Utrecht University. From 1995 to 2004 she had a permanent position as Assistant and Associate Professor in the Department of Artificial Intelligence at the Vrije Universiteit Amsterdam. After that for two years she held a position as full professor in AI and Cognitive Science at the Radboud University Nijmegen. Since 2006 she has a position as full profesor in Human Computer Interaction and Artificial Intelligence at the Delft University of Technology. Her research has focussed on the design and analysis of agent systems and their applications to information agents and Electronic Commerce. Currently a general theme of her reseach interests is dynamics of behaviour of multiple agents (human and software) in a dynamic environment.

Jan Treur received his Ph.D. in Mathematics and Logic in 1976 from Utrecht University. Since 1986 he works in Artificial Intelligence, from 1990 as a full professor and head of the Department of Artificial Intelligence at the Vrije Universiteit Amsterdam. In the 1990s he headed a research programme on component-based design of knowledge-based and agent systems. In the last seven years the research programme focussed on modelling dynamics of agent systems related to other disciplines such as Biology, Cognitive Science, Criminology, Organisation Theory, and Philosophy of Mind.