

Automatic Verification of Biochemical Network Using Model Checking Method

Jinkyung Kim, Younghee Lee and Il Moon*

Department of Chemical Engineering, Yonsei University, 134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea

Abstract This study focuses on automatic searching and verifying methods for the reachability, transition logics and hierarchical structure in all possible paths of biological processes using model checking. The automatic search and verification for alternative paths within complex and large networks in biological process can provide a considerable amount of solutions, which is difficult to handle manually. Model checking is an automatic method for verifying if a circuit or a condition, expressed as a concurrent transition system, satisfies a set of properties expressed in a temporal logic, such as computational tree logic (CTL). This article represents that model checking is feasible in biochemical network verification and it shows certain advantages over simulation for querying and searching of special behavioral properties in biochemical processes.

Keywords automatic verification, path networks, biological process, model checking, computational tree logic

1 INTRODUCTION

With the rapidly growing amount of new experimental data, it has become clear that biologists need systematical methods and computational tools to prove, analyze, compare and simulate these complex networks. To develop computer applications that aid in this study, it is first necessary to be able to represent biological systems and networks. Biological process consists of many levels of detail, ranging from atomic mechanism to general processes, such as cell division, cell adhesion or cell invasion. These levels are connected by their functions, conditions and logics. The hierarchical representation of the biological process is therefore a major challenge for informatics. Based on the representation, this article describes a novel method for path analysis and verification in biological process networks using model checking method which is an automatic method for deciding if a circuit or logic is expressed as a concurrent transition system.

Models of biochemical networks may become quite large, as they include many genes, proteins, and complex interconnections, manual verification of these properties, behaviors and logics is error-prone or even practically infeasible. The quantitative information about the dynamics of biological systems is not sufficient even with the current best technology, thus it is difficult for traditional numerical models and analysis techniques to be applied. In addition, numerical predictions on the biological system are difficult to verify, because available data are mostly qualitative in nature. Simulators are often used to investigate the behavior of biological processes based on the determined model. Unfortunately, examining the output of simulation is usually time-consuming, so exhaustive simulations are rarely feasible. Although simulations are helpful, in practice they cannot be used to ensure the proper behavior of the system.

The use of formal tools for modeling biochemical processes and for reasoning about the behaviors seems to be a required research to contribute a lot in com-

puter science and bioinformatics. Automatic verification technique using model checking is to ensure that the comparison of model predictions with experimental data is efficient and reliable.

Recently, researchers have explored the possibility of applying the model checking methods to biological and biochemical system using both differential equation based framework and logic based frameworks. Antonioti *et al.* [1] used the computational tools Simpathica and XSSYS to perform verification on a set of differential equations representing the behavior of biochemical pathways. Eker *et al.* [2] used the term rewriting tool Maude to perform model checking on biological signaling pathways, where possible reactions are modeled as logic rules. Chabrier and Fages [3] also modeled possible reactions as logic rules and used the computational tools NuSMV and DMC to perform model checking on mammalian cell cycle control. In this article, an approach toward model verification and alternative path search is proposed to focus on the issue of providing automated model checking method for querying and verifying models in biochemical processes.

2 MODEL CHECKING FOR BIOLOGICAL PROCESS MODELING

Model checking is an automatic technique that, given a finite-state model of a system and a logical property, systematically checks whether this property holds for that model. Model checking is performed as an exhaustive state space search that is guaranteed to terminate since the model is finite. The first work of this study is to develop a biological model that can represent the knowledge about hierarchical system structure including properties, networks amongst the properties, system behaviors and transition logics (OR, XOR and AND). Based on the model, a path search with constraints, the control flow properties, transitions and qualitative simulations can be verified. The search

for alternative paths and their analysis are a crucial point in the understanding of biological networks. The verification for transition logics including logical operator (OR, XOR and AND) is also required in the biological networks.

2.1 Model description and simulation

Model checker consists of three parts, model description, simulation and model checking. The model description is a non-deterministic guarded command language with simple data [4, 5]. It serves as a modeling or design language to describe the biological process behaviors (*e.g.* activity, metabolite, compound, catalyst, inhibitor, *etc.*) as networks of automata extended with time and transition logics, reactions, constraints and control properties in biological processes.

The biological process is modeled based on automata theory as a network of places and transitions in hierarchical and parallel structure. The model is further extended with bounded discrete variables that are part of the state. A state of the system is defined by the locations of all automata and the values of the discrete variables. Every state may fire and edge separately or synchronize with another state, which leads to a new state. Transitions are used to change location

and transitions between these locations and define how the biological system behaves. To control when to fire a transition, it is possible to have a guard and synchronization. A guard is a condition on the variables saying when the transition is enabled. The synchronization mechanism is a hand-shaking synchronization [6, 7].

2.2 Path search and verification

The graphical simulation of the developed model is a validation which enables examination of possible dynamic pathways executions of a biological system. The model checker accepts a model description and specifications written as temporal logic formulas, and it determines if the formulas are true or not for that model. The strength of this method is to synthesize a feasible sequence (desirable or undesirable pathways) through a counter example and to verify its correctness using computation tree logic (CTL) simultaneously. The queries available in the CTL are: $E \diamond p$: there exists a path where p eventually hold, $A[] p$: for all paths p always hold, $E[] p$: there exists a path where p always hold, $A \diamond p$: for all paths p will eventually hold, and $p \rightarrow q$: whenever p holds q will eventually hold [7]. These CTL have been applied in path

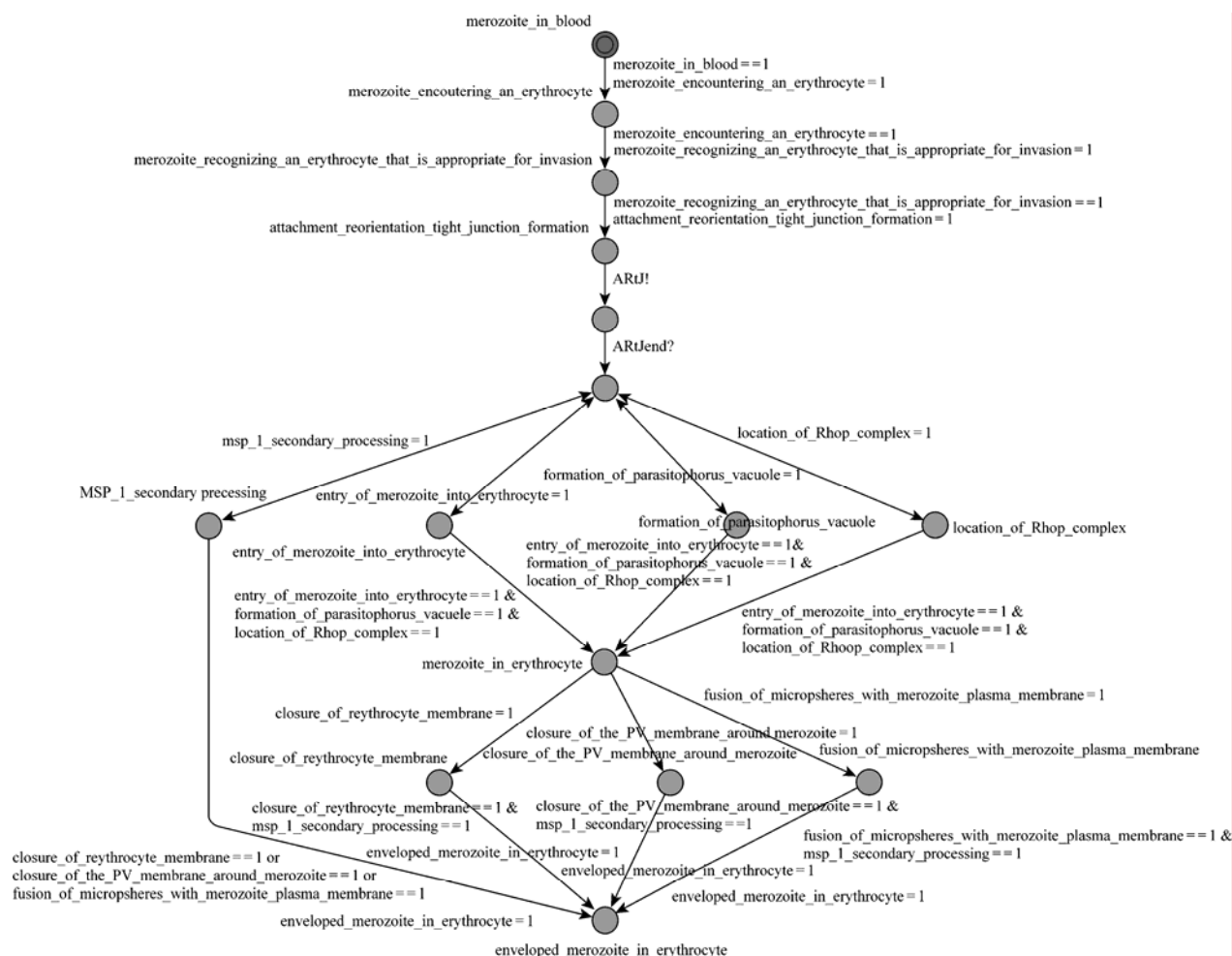


Figure 1 Automata model of the invasion of erythrocyte by *P. falciparum* merozoites

searches in biological process to find alternative paths with constrains, (with or without specific activity, reaction, a shortest pathway, a fastest pathway, some pathways in parallel). A counter example shows the result of these queries. This approach was tested by representing malaria parasites invading host erythrocytes [8] and composed queries to find alternative pathways in the biological system networks. This study represents an automatic reachability verification to answer queries using CTL about the pathways of biological processes.

3 CASE STUDY

3.1 The invasion process model

This process is based on the invasion of human host erythrocytes by the malarial parasite *Plasmodium falciparum* [9]. Fig. 1 shows the automata model of the invasion process. The invasion process has three steps. In the first step, the merozoite recognizes an erythrocyte and attaches to it. In the second step, the merozoite enters the erythrocyte and is positioned inside a vacuole within the cytoplasm of the erythrocyte. In the final step, the membrane of the parasitophorous vacuole and the plasma membrane of the erythrocyte close. The invasion process starts with “merozoite in blood”, followed by the processes “merozoite encountering an erythrocyte” and by “merozoite recognizing an erythrocyte that is appropriate for invasion”. The next stage in the invasion process involves “attachment, reorientation, and tight-junction formation”. The details and hierarchy of the activity “attachment, reorientation,

and tight-junction formation” is shown in Fig. 2. The process in Fig. 2 is a submodule of the expansion of the activities named as ARTJ in Fig. 1. The first stage of the process shown in Fig. 2 involves one of the three activities, because these activities are connected to the upper process by “XOR” transition logic. Three alternative pathways are composed of “alternative pathway for initial attachment involving glycoporphin B”, “Initial attachment to erythrocyte involving glycoporphin A”, and “alternative pathway for initial attachment involving no sialic acid”. One of these activities can be followed by “reorientation of merozoite”. After every stage in Fig. 2 is finished, the process is synchronized to “ARTJend” in Fig. 1. Both “ARTJ” and “ARTJend” have no meaning in the invasion process and are as the synchronization stages. In the following stage in Fig. 1, four activities are done in parallel, because of the four activities connected to the next stages by “AND” transition logic. “Entry of merozoite into erythrocyte”, “formation of parasitophorous vacuole”, and “localization of Rhop complex” are followed by “merozoite in erythrocyte” in parallel. “MSP-1 secondary processing” is shown on the left side of the model. It is known that it correlates with the time of either the last stage or stage before last of the invasion process. This activity is, therefore, specified in parallel to both these stages.

3.2 Verification and alternative path search

Specification 1

$A[] !(((\text{glycophorin_B}=1 \ \& \ \text{glycophorin_A}=1) \ \text{or} \ (\text{glycophorin_B}=1 \ \& \ \text{no_sialic_acid}=1) \ \text{or} \ (\text{glyco-}$

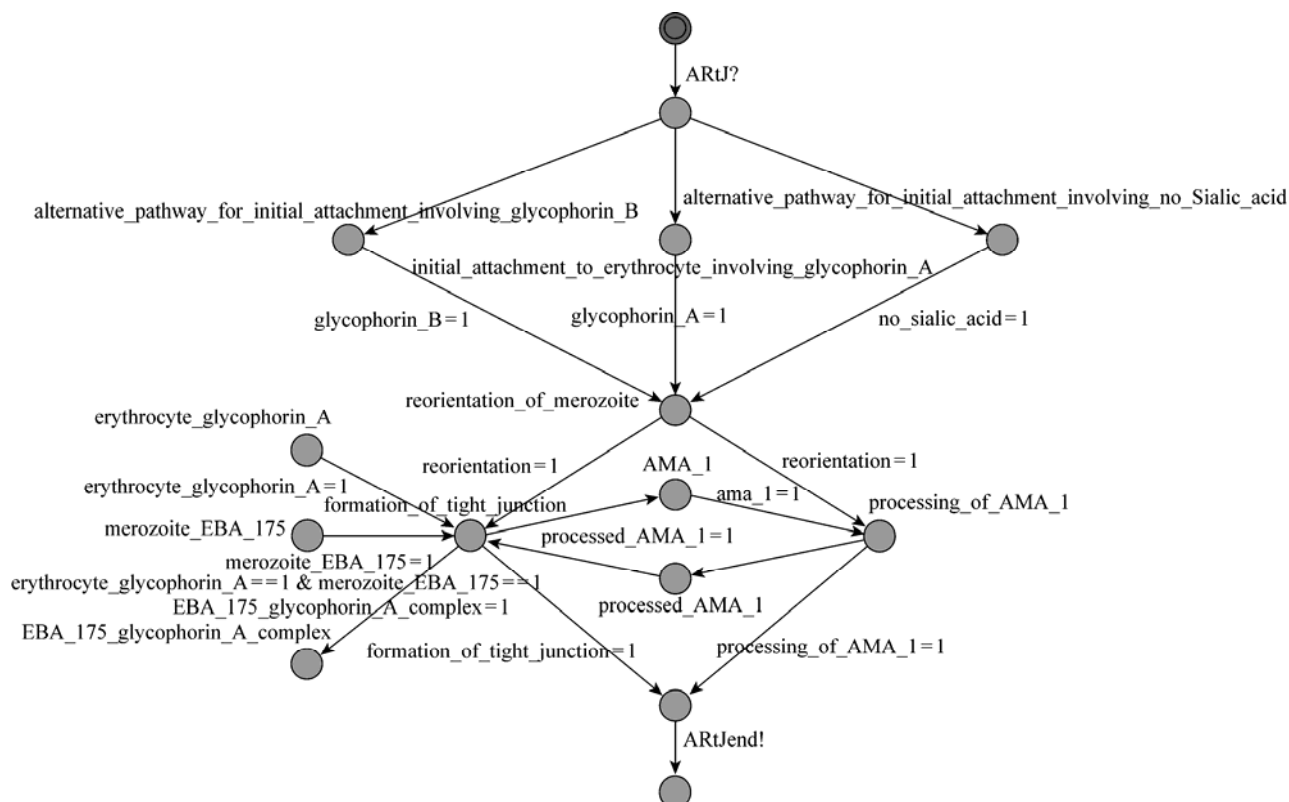


Figure 2 The submodule of the stage “Attachment, reorientation, and tight-junction formation” of Fig. 1

phorin_A==1 & no_sialic_acid==1)) & enveloped_merozoite_in_erythrocyte==1)

Specification 2

A[] !(enveloped_merozoite_in_erythrocyte==1 & (msp_1_secondary_processing==0 or entry_of_merozoite_into_erythrocyte==0 or formation_of_parasitophorus_vacuole==0 or location_of_Rhop_complex==0))

phorus_vacuole==0 or location_of_Rhop_complex==0))

Specification 3

A[] !(glycophorin_B==0 & enveloped_merozoite_in_erythrocyte==1)

Specification 4

UPPAAL version 4.0.4 (rev. 2887), January 2007 -- server.

A[](((glycophorin_B==1 & glycophorin_A==1) or (glycophorin_B==1 & no_sialic_acid==1) or (glycophorin_A==1 & no_sialic_acid==1)) & enveloped_merozoite_in_erythrocyte==1)

Property is satisfied.

A[]!(enveloped_merozoite_in_erythrocyte==1 & (msp_1_secondary_processing==0 or entry_of_merozoite_into_erythrocyte==0 or formation_of_parasitophorus_vacuole==0 or location_of_Rhop_complex==0))

Property is satisfied.

A[]!(glycophorin_B==0 & enveloped_merozoite_in_erythrocyte==1)

Property is not satisfied.

A[]!(glycophorin_B==1 & enveloped_merozoite_in_erythrocyte==1)

Property is not satisfied.

A[]!(enveloped_merozoite_in_erythrocyte==1 & closure_of_erythrocyte_membrane==1)

Property is not satisfied.

Figure 3 The results of Specifications 1, 2, 3, 4 and 5

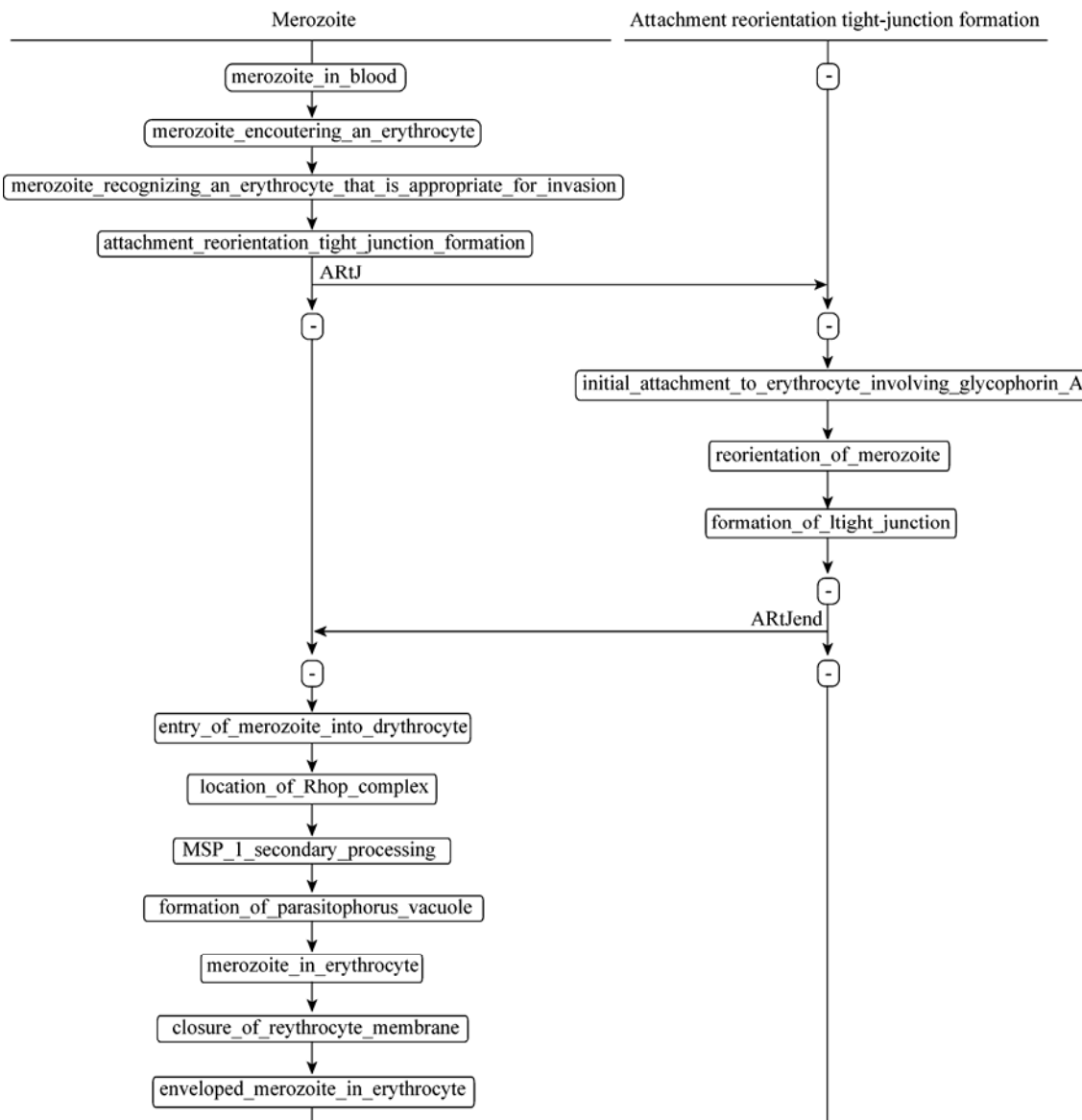


Figure 4 The alternative path for Specification 3

$A[] \text{!(glycophorin_B}=1 \ \& \ \text{enveloped_merozoite_in_erythrocyte}=1)$

Specification 5

$A[] \text{!(enveloped_merozoite_in_erythrocyte}=1 \ \& \ \text{closure_of_erythrocyte_membrane}=1)$

These specifications are used for verification of transition logics and path search queries with constraints (see Fig. 3). Specification 1 is a question for the verification of transition logical operator "XOR" involved in three activities "Alternative pathway for initial attachment involving glycophorin B", "initial attachment to erythrocyte involving glycophorin A", and "Alternative pathway for initial attachment involving no sialic acid". Specification 2 is a question for verifying that these activities "entry of merozoite into erythrocyte", "formation of parasitophorous vacuole", "localization of Rhop complex", and "MSP-1 secondary processing" eventually get to "enveloped merozoite in erythrocyte" in parallel. Specification 3 is a query to find an alternative pathway from "merozoite in blood" to "enveloped merozoite in erythrocyte" excluding the activity "alternative pathway for initial attachment involving glycophorin B". An alternative pathway including the activity "Alternative pathway for initial attachment involving glycophorin B" can be found by Specification 4. Specification 5 is a final query to find a pathway including "closure of erythrocyte membrane".

4 RESULTS

The answers for Specifications 1 and 2 are satisfied. These results illustrate that the developed model of the invasion process is available and can be validated for the transition logic operators. The answers for Specifications 3, 4 and 5 are not satisfied. These questions have logical meaning that there is no pathway to find in the entire process stages. The answer is not satisfied if an alternative pathway is in the process. Model checking method shows a counter example to prove the answer. The counter example can give an alternative pathway to find by tracing the simulated model. Fig. 4 shows a counter example of Specification 3. This case study illustrates that model checking method can find alternative pathways, which give useful information to biochemists, when the system is very complex. Although biological process has hierarchical structures, the pathways can be easily found because they are modeled by sub-modules. It can also verify structural logics, such as AND, OR and XOR amongst the branches of biological networks. The model checking method becomes a useful computer-aided technique to make up for the weak points of the experiments.

5 CONCLUSIONS

This research introduces a new modeling and searching technique to verify the transition logics and to find alternative paths with constraints in biological networks. Discrete models for all possible path networks are developed using automata theory and specifications to find desirable and undesirable paths are formulated using CTL. How symbolic model checking techniques could be applied to the querying and verification of Boolean abstractions of biological networks is also shown. Model checker tool verifies the logical operation and produces the alternative paths of the sensible biological interpretation. The main advantage of this approach is that automatic verification and paths search with interest constraint through all possible paths and behaviors in complex biological networks.

NOMENCLATURE

A	for all computation paths (path quantifier)
E	for some computation paths (path quantifier)
"or"	logical or
[]	globally all along the states (state quantifier)
<>	in the future (state quantifier)
"!"	logical not
"&"	logical and
"="	equality

REFERENCES

- 1 Antoniotti, M., Policriti, A., Ugel, N., Mischra, B., "Model building and model checking for biochemical processes", *Cell Biochem. Biophys.*, **38**, 271–286 (2003).
- 2 Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Talcott, C., "Pathway logic: executable models of biological networks", *Electron. Notes in Theor. Comput. Sci.*, **71**, 144–161 (2002).
- 3 Chabrier, N., Fages, F., "Symbolic model checking of biochemical networks", In: Proceedings of the International Workshop on Computational Methods in Systems Biology, number 2602 in Lecture Notes in Computer Science, Rovereto, Italy, 149–162 (2003).
- 4 Kim, J., Moon, I., "Synthesis of safe operation procedure for multi-purpose bath processes using SMV", *Comput. Chem. Eng.*, **24**, 385–392 (2000).
- 5 Moon, I., Power, G.J., Burch, J.R., Clarke, E.M., "Automatic verification of sequential control systems using temporal logic", *AIChE J.*, **38**, 67–75 (1992).
- 6 Alur, R., Dill, D., "A theory of timed automata", *Theor. Comput. Sci.*, **126**, 183–235 (1994).
- 7 Behrmann, G., David, A., Larsen, K.G., "A tutorial on Uppaal", <http://www.uppaal.com>.
- 8 Peleg, M., Yeh, I., Altman, R.B., "Modeling biological processes using workflow and Petri net models", *Bioinformatics*, **18**, 825–837 (2002).
- 9 Blackman, M.J., "Proteases involved in erythrocyte invasion by the malaria parasite: Function and potential as chemotherapeutic targets", *Curr. Drug Targets*, **1**, 59–83 (2000).