

Combining Refinement and Signal-Temporal Logic for Biological Systems

Usman Sanwal¹(✉) and Umair Siddique²

¹ Computational Biomodeling Laboratory, Department of Computer Science, Åbo Akademi University, Turku Centre for Computer Science, 20500 Turku, Finland
msanwal@abo.fi

² Department of Computing and Software, McMaster University, Hamilton, Canada

Abstract. System-level modeling and analysis of biological phenomena have become an important research topic amongst different fields including mathematics, computer science, electrical and system engineering. This is a consequence of the recent development in these fields which can be utilized to understand the dynamics of complex biological organisms such as cancer, malaria and diabetes, etc. However, the concept of model refinement (i.e., the transformation of an abstract models into a detailed model) is largely unexplored in biology. In this paper, we describe our ongoing project which aims at combining the concept of model refinement and temporal logic for the analysis of a wide class of biological systems.

Keywords: Biomodeling · Refinement · Event-B · Signal-Temporal Logic

1 Introduction

In the last two decades, significant progress has been made towards the system-level modeling of biological systems and diseases such as cell signaling pathways, metabolic networks, diabetes and cancer. The concept of *abstraction* provides a convenient method to model complex systems at various levels of detail. The main idea is to build a simple model of the system and progressively add details which ultimately allow us to consider a suitable model for a specific study. Recent research in this direction includes differential equations based modeling [9], Petri nets [8], guarded command languages [7] and rule-based modeling [4]. The main focus of this research project is twofold:

- modeling of biological systems in Event-B [2], a formal method for stepwise development and refinement of complex models. To the best of our knowledge, this is the first time that Event-B is used for describing biological systems. One of the main strengths of refinement in Event-B is the formal relation amongst different abstract models, which is established through gluing invariants. However, establishing such a relation is difficult in other approaches, e.g., models based on ordinary differential equations, continuous time Markov chains and Petri nets.

- transforming Event-B models into a suitable form (e.g., a systems of differential equations) for quantitative analysis using the Signal-Temporal Logic (STL), a temporal logic capable of reasoning about continuous functions. Consequently, we can quantitatively evaluate interesting biological behaviors such as *oscillations*, *stabilization* and *interdependence* of different biological entities in the model. For example, we can describe the following property in STL: “if the concentration of entity x goes above some threshold θ_1 then within $t \in [T_1, T_2]$ time the concentration of entity y drops below θ_2 ”.

Organization: Sects. 2 and 3 provide an overview of Event-B and STL, respectively. In Sect. 4, we describe our proposed framework and current status of the project along with the description of ongoing and future tasks. Finally, Sect. 5 concludes the paper.

2 Event-B

Event-B [2] is a formal method widely used for modeling and reasoning about complex computing systems. The core concept and design of Event-B are inspired by B-Method [1]. In Event-B, we can progressively add details in the underlying model (i.e., introduction of new system variables) through refinement. Event-B model is composed of two components: Machine and Context. The machine is the dynamic part of the Event-B model and consists of variables, invariants and events whereas context is the static part of the Event-B model and it includes axioms, constants and theorems.

Example 1: Metabolic networks describe the physiological and biochemical properties of biological cells. Every species is modelled by a variable of type \mathbb{N} (natural numbers) denoting the current number of that species. The state of the system is modelled by the species interacting with each other and the reactions become events in Event-B. Assume we have two reactions:



An Event-B model for these reactions consists of 3 variables, corresponding to the three species:

VARIABLES x, y, z

The types of these variables are specified as invariants $x \in \mathbb{N}$, $y \in \mathbb{N}$ and $z \in \mathbb{N}$ and their initial values are described by an Initialisation event. For modelling these two reactions, we define two events. To model the event guard, we need the precondition necessary for the event to happen. In our case, this is ensured by having enough reactants to consume. Thus, the guard of the first event will be $x \geq 2$ (we can conserve two x quantities) and the guard of the second event

will be $x \geq 1$ and $y \geq 1$. The actions of the event model the updates in the species involved in the reaction, including both the reactants and the products. Thus, as in the first reaction we consume $2x$ and produce y , this is modelled in the actions: $x := x - 2$ and $y := y + 1$. In the second reaction, we produce z and consume for this x and y , hence the corresponding actions are $x := x - 1$, $y := y - 1$ and $z := z + 1$. *Proof obligations* of these events are discharged automatically. In Event-B, these two events have the following form:

FirstEvent
WHERE
@grd1 $x \geq 2$
THEN
@act1 $x := x - 2$
@act2 $y := y + 1$
END

SecondEvent
WHERE
@grd1 $x \geq 1$
@grd1 $y \geq 1$
THEN
@act1 $x := x - 1$
@act2 $y := y - 1$
@act2 $z := z + 1$
END

3 Signal-Temporal Logic (STL)

Temporal logic provides a mechanism to specify evolution of a system behaviour over time. Traditionally, temporal logic has been used to specify the properties of software and hardware systems over discrete state-space. However, *Metric Interval Temporal Logic* (MITL) [3] uses the notion of dense-time and allows to model continuous evolution of system variables. Our work is based on *Signal Temporal Logic* (STL) [10] which strengthens MITL by incorporating predicates on the real-valued variables.

STL Syntax: A formula φ in STL is constructed from atomic predicates which describe the instantaneous behaviour of systems variables, combined using Boolean and temporal operators. Formally, a formula is constructed using the following grammar:

$$\varphi ::= \mu \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \mathbf{U}_{[a,b]} \varphi \mid \diamond_{[a,b]} \varphi \mid \square_{[a,b]} \varphi$$

where μ , \mathbf{U} , \diamond and \square represent, a predicate, ‘until’, ‘eventually’ and ‘always’ operators, respectively. Formally, μ describes a generic constraint applied to a trajectory ζ defined as a function of time instant τ . For example, such a constraint (corresponding to example in the Introduction) is $\mu : (x(\tau) - \theta_1 > 0)$. Temporal formula $\diamond_{[0,5]} \mu$ describes that the constraint μ has to be true at least once within 5 time units, whereas $\square_{[0,5]} \mu$ specifies that the constraint μ must be true all time during the 5 time units. Whereas temporal formula $\phi_1 \mathbf{U}_{[0,5]} \phi_2$ is satisfied if ϕ_1 holds continuously until some time within 5 time units when ϕ_2 becomes true.

STL Semantics: The semantics of STL has been defined for both qualitative (Boolean) [10] and quantitative [6] evaluation. The Boolean semantics of STL decides whether the trajectory ζ satisfies the formula φ at time τ by structural induction on the STL formula. The quantitative semantics of STL is defined in

terms of a function \mathcal{T} which takes the trajectory ζ , time τ and formula φ and returns a real-valued number $\mathcal{T}(\varphi, \zeta, \tau)$ quantifying the degree of satisfaction of φ by ζ at time τ .

Breach [5] is a MATLAB/C++ toolbox which provides an efficient framework to formally model hybrid dynamical systems and analyze their properties specified in STL. Some important features of Breach include simulation and plotting of continuous signals, qualitative and quantitative monitoring of STL properties and parameter synthesis of STL specifications.

Example 2: The biological reaction described in Eq. (2) can be transformed into a differential equations based model and consequently specified in Breach as follows:

$$\begin{cases} \frac{dx(t)}{dt} = -\alpha x(t)y(t) \\ \frac{dy(t)}{dt} = -\alpha x(t)y(t) \\ \frac{dz(t)}{dt} = \alpha x(t)y(t) \\ x(0) = p_1, y(0) = p_1 \text{ and } z(0) = p_3 \end{cases} \quad (3) \quad (\text{initial conditions})$$

where α represents the rate of reaction, and variables $x(t)$, $y(t)$ and $z(t)$ represent the concentration of X , Y and Z at time t , respectively. We can now formalize several interesting biological behaviours in STL. For example:

- $z(t)$ will become more than 10 within 5 s: $\Diamond_{[0,5]}(z(t) \geq 10)$
- $z(t)$ remains low until $x(t)$ stabilizes within 10 s:
 $z(t) < 1 \mathbf{U}_{[0,10]}(\Box(\|\frac{dx(t)}{dt}\| < 0.0000001))$

Notice that \Box without a (subscripted) time interval represents $\Box_{[0,\infty]}$.

4 Framework for Biomodeling and Analysis

The proposed framework, given in Fig. 1, outlines the main idea behind our biomodeling and analyzing a wide range of biological systems. Our modeling starts from the available literature of a biological network (e.g., metabolic network, cancer or malaria, etc.). As a first step, we build an abstract model \mathcal{M}^1 in Event-B. At this point, we have two targets:

- transform the abstract model (\mathcal{M}^1) into an equivalent differential equations based model \mathcal{D}^1 ;
- build a refined model \mathcal{M}^2 depending upon the availability of additional information.

In biology, it is common to find underlying networks with various levels of details. For example, it can be a consequence of different studies by different research groups or discovery of new biological entities through experimental observations (e.g., imaging, laser scanning, etc.). Following these steps, we can

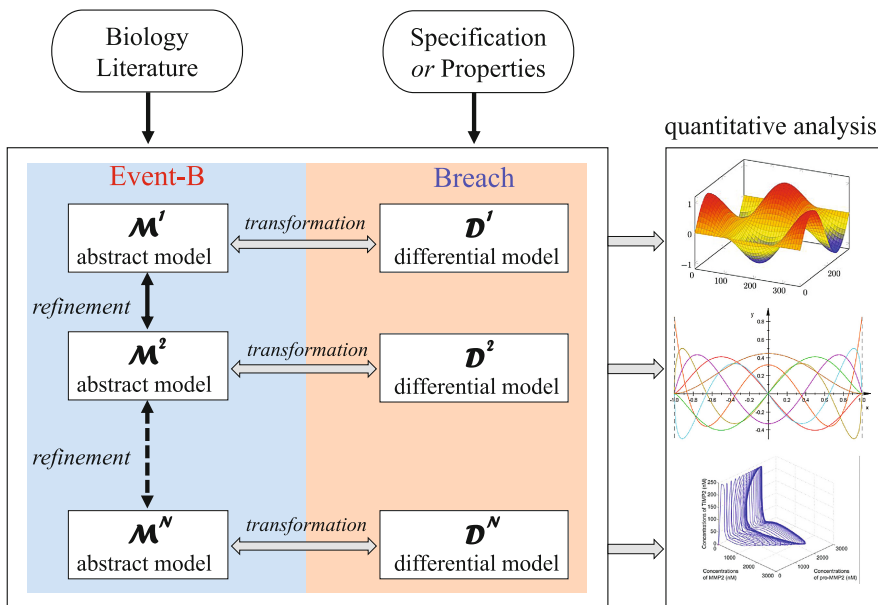


Fig. 1. Proposed framework for biomodeling and refinement

build N refined models ($\mathcal{M}^1 - \mathcal{M}^N$) through successive refinements and N differential models ($\mathcal{D}^1 - \mathcal{D}^N$) through transformation for each refined model (\mathcal{M}^i to \mathcal{D}^i). During the course of this development, we can perform quantitative analysis for each model (\mathcal{D}^i) for a given biological property specified as an STL formula. Indeed, we perform quantitative analysis in Breach toolbox and can visually inspect (through various plots) interesting behaviours. Note that our framework provides two novel advantages as compared to other approaches: Firstly, each refined model is formally linked and serves for bookkeeping purposes in a formal language. Secondly, we can perform quantitative analysis in various steps which can greatly reduce computational cost. In fact, analysis of some properties might take huge computational resources or might not terminate for some cases. However, it might be sufficient to analyse certain properties at a higher-level of abstraction thus reducing the size of the model.

As a proof-of concept, the molecular model for heat shock response introduced in [11] was formally modeled in Event-B in [12]. The model was first introduced in a simplified presentation, after which details were added to it through 5 consecutive refinements in Event-B. As a result of this refinement, model expands from 10 species and 17 irreversible reactions to 20 species and 55 irreversible reactions. Our ongoing and future work includes the following tasks:

- Formalize the rules for transforming the model \mathcal{M}^i to \mathcal{D}^i . Indeed transformation rules are based on the Law of Mass Action which provides ODEs for the concentrations of the species involved in the bio-(chemical) reactions. We plan

to formalize this transformation in HOL Light¹ proof assistant and formally prove its soundness.

- Develop a template for frequently used STL properties (e.g., oscillations, stabilization, steady state, etc.). Potentially, we would like to generate them from plain text, so that our framework can be used by biologists without prior knowledge of Event-B and STL. Another important aspect is to explore the parameter synthesis capabilities of Breach toolbox.
- Build an extensive library of biological networks including cancers, diabetes, heart diseases and brain disorders.
- Build a graphical user interface to seamlessly combine Event-B and Breach toolbox through verified transformation rules.

5 Conclusion

We describe in this paper our ongoing project involving quantitative model refinement as an approach to stepwise construction of a biological network in Event-B and quantitative analysis of Signal-Temporal Logic specifications in Breach toolbox. This is useful from several points of view: First, to the best of our knowledge, there is no work available related to implementation of biological models in Event-B, except the proof of concept model of the heat shock response in [12]. Second, formally refined models can serve as a library of reusable biological models. Third, parametric Signal-Temporal Logic specifications for each refined model can be used for parameter synthesis. Such parameters can provide some useful indications for future drug designs for infectious diseases.

References

1. Abrial, J.-R.: *The B-book: Assigning Programs to Meanings*. Cambridge University Press, New York, NY, USA (1996)
2. Abrial, J.-R.: *Modeling in Event-B: System and Software Engineering*, 1st edn. Cambridge University Press, New York, NY, USA (2010)
3. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. *J. ACM* **43**(1), 116–146 (1996)
4. Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J.: Rule-based modelling and model perturbation. In: Priami, C., Back, R.-J., Petre, I. (eds.) *Transactions on Computational Systems Biology XI*. LNCS, vol. 5750, pp. 116–137. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-04186-0_6](https://doi.org/10.1007/978-3-642-04186-0_6)
5. Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: Touili, T., Cook, B., Jackson, P. (eds.) *CAV 2010*. LNCS, vol. 6174, pp. 167–170. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-14295-6_17](https://doi.org/10.1007/978-3-642-14295-6_17)
6. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) *FORMATS 2010*. LNCS, vol. 6246, pp. 92–106. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-15297-9_9](https://doi.org/10.1007/978-3-642-15297-9_9)

¹ <http://www.cl.cam.ac.uk/~jrh13/hol-light/>.

7. Gratie, D.E., Iancu, B., Azimi, S., Petre, I.: Quantitative model refinement in four different frameworks, with applications to the heat shock response. In: Petre, L., Sekerinski, E. (eds.) *From Action Systems to Distributed Systems*, pp. 201–214. Taylor & Francis, Boca Raton (2016)
8. Gratie, D.E., and Petre, I.: Hiding the combinatorial state space explosion of bio-models through colored petri nets. *Annals of University of Bucharest, LXI*, pp. 23–41 (2014)
9. Iancu, B., Czeizler, E., Czeizler, E., Petre, I.: Quantitative refinement of reaction models. *Int. J. Unconventional Comput.* **8**(5–6), 529–550 (2012)
10. Maler, O., Nickovic, D., Pnueli, A.: Checking temporal properties of discrete, timed and continuous behaviors. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds.) *Pillars of Computer Science. LNCS*, vol. 4800, pp. 475–505. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-78127-1_26](https://doi.org/10.1007/978-3-540-78127-1_26)
11. Petre, I., Mizera, A., Hyder, C.L., Meinander, A., Mikhailov, A., Morimoto, R.I., Sistonen, L., Eriksson, J.E., Back, R.-J.: A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Nat. Comput.* **10**(1), 595–612 (2011)
12. Sanwal, U., Petre, L., and Petre, I.: Stepwise construction of a metabolic network in Event-B: the heat shock response. *TUCS Technical reports*, (1160) (2016)