

Scalable Reaction Network Modeling with Automatic Validation of Consistency in Event-B

Usman Sanwal

Abo Akademi University

Thai Son Hoang

University of Southampton

Luigia Petre

Abo Akademi University

Ion Petre (✉ ion.petre@utu.fi)

University of Turku

Research Article

Keywords: Scalable Reaction Network Modeling , Automatic Validation, biological model, error-prone process

Posted Date: August 26th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-809761/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Scalable reaction network modeling with automatic validation of consistency in Event-B

Usman Sanwal¹, Thai Son Hoang², Luigia Petre¹, and Ion Petre^{3,4,*}

¹Abo Akademi University, Computer Science, Turku, 20500, Finland

²School of Electronics and Computer Science, University of Southampton, UK

³University of Turku, Department of Mathematics and Statistics, Turku, 20014, Finland

⁴National Institute for Research and Development in Biological Sciences, Bucharest, 060031, Romania

*Correspondence to: ion.petre@utu.fi

ABSTRACT

Constructing a large biological model is a difficult, error-prone process. Small errors in writing a part of the model cascade to the system level and their sources are difficult to trace back. In this paper we extend a recent approach based on Event-B, a state-based formal method with refinement as its central ingredient, allowing us to validate for model consistency step-by-step in an automated way. We demonstrate this approach on a model of the heat shock response and its scalability on a model of the ErbB signaling pathway, a key evolutionary pathway with a significant role in development and in many types of cancer. All consistency properties of the model were proved automatically with computer support.

Introduction

Biological processes are instances of large, complex, concurrent systems. It is remarkably difficult to capture all the necessary details of such a system in a single all-encompassing modeling step¹: details that are critical in some parts of the model are neutral in other parts of it; modules build upon each other in a structured way; there can be several levels of detail. An effective solution that has been proposed for this problem²⁻⁷ is to use model refinement: gradually adding details to a model while preserving its consistency. This splits the modeling processes into two stages: build first a simplified, abstract version of the model (and verify/ensure its consistency) and then add details to it step-by-step in a way that ensures that the model remains consistent. At each refinement step, one can focus only on the new elements that are introduced and on their consistency with the previous model. This approach allows to also separate the reasoning about the system under development into smaller steps. Model refinement has been introduced in biomodeling in several different frameworks, such as ODE-based modeling⁸, Boolean networks⁷, process algebras^{3,9}, rule-based modeling^{5,10}, and Petri nets¹¹⁻¹³. The key challenge in deploying this method in practice is verifying the consistency of the initial/basic model and ensuring that the model remains consistent in each step of the refinement. This is very difficult as the model size increases to hundreds/thousands of variables influencing each other through concurrent processes. A small error in writing a part of the model (say, in a variable name or in the multiplicity of a reactant or of a product) cascades into system-level inaccuracies between the basic and the refined versions of the model, whose sources are very difficult to identify.

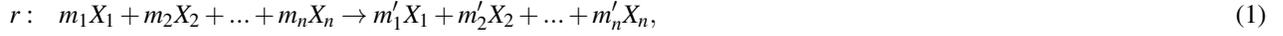
To address the problem of ensuring the model consistency throughout the construction of the model we recently proposed¹⁴ using Event-B for biomodeling. Event-B¹⁵ is a formal method for system modeling, with its original motivations rooted in the specifications of complex software systems. It is based on set theory, with refinement at its core, and with a focus on mathematical proofs that the different refinement levels of a model are consistent. Crucially, proving the consistency is done in Event-B at the level of its basic events, rather than on the system as a whole. This allows the modeler to identify easily the source of model inconsistencies in the events whose proof obligations fail to get discharged. Event-B is accompanied by the software toolset Rodin¹⁶.

In this paper we extend the Event-B-based modeling technique¹⁴ to give the first scalable demonstration of Event-B-based biomodeling. We show that simple mathematical functions can be used to hide some of the details of the refinement, leading to a reduction in the number of events of the model. We show that a model of the heat shock response can be described through 17 events, instead of the 57 events in an earlier model¹⁴. More drastically, we show that a model of the ErbB signaling pathway consisting of 1320 reactions can be written through only 242 events. The models were implemented using Rodin¹⁶ and all proof obligations related to demonstrating the model consistency were automatically discharged in Rodin. All Event-B models discussed in the paper can be downloaded at <https://tinyurl.com/cf7vxs2e>.

Results

From a reaction network to an Event-B model

We model reaction networks as sets of biochemical reactions, where each reaction specifies its reactants, products, and possibly inhibitors and catalyzers. For simplicity, we consider each reversible reaction in our methodology as two irreversible reactions. With these assumptions, a reaction r can be written as a rewriting rule of the form:



where $\mathcal{S} = \{X_1, \dots, X_n\}$ is the set of *reactants* and $m_1, \dots, m_n, m'_1, \dots, m'_n \in \mathbb{N}$ are non-negative integers.

Reaction networks can be modeled straightforwardly in Event-B¹⁴: every reactant is modeled by a variable and every reaction is modeled by an event. Invariants ensure the consistency of the model as well as other biological properties of interest, for instance the mass conservation rule that requires that the number of certain reactants remains constant in the system. The general form of an Event-B model corresponding to reaction (1) is presented in Table 1; more reactions in the network model lead to more events in the Event-B model¹⁴. X_1, X_2, \dots, X_n are the variables of the model, whose initial values are set in the initialization event. For each reaction r of the reaction network, we specify in its guard that it must have enough of each reactant in order for the reaction to be enabled, while the action of the event specifies the changes to happen to each variable.

The data refinement of a model is about adding details to it by replacing generic variables with more specific versions of them. The refined model can include the reactions of the basic model where the refined variables replace the generic ones in all (or some) possible combinations. New reactions between the refined variables can also be added. Obviously, this easily leads to a combinatorial explosion in the size of the refined model. To see this, consider a binary binding reaction $A + B \rightarrow C$. When variable A is refined into two specific versions of it $\{A_0, A_1\}$, B into $\{B_0, B_1, B_2\}$, this leads to a refinement of C into $\{C_{i,j} \mid i = 0, 1, j = 0, 1, 2\}$. The reaction is replaced by six refined versions of it: $A_i + B_j \rightarrow C_{i,j}$, for all $0 \leq i \leq 1, 0 \leq j \leq 2$.

The case of a homogeneous binary reaction (also called a dimerization reaction) is similar: reaction $2A \rightarrow C$ and a refinement of A into $\{A_0, A_1\}$ leads to a refinement of C into $\{C_0, C_1, C_2\}$ and three refined versions of the reaction: $A_i + A_j \rightarrow C_{i+j}$, with $i, j \in \{0, 1\}$.

Size-preserving reaction network refinement in Event-B

We propose in this paper a new approach to reaction network refinement in Event-B that keeps the number of variables and events unchanged. To refine a variable X into l versions of it $\{X_1, X_2, \dots, X_l\}$ we replace the integer variable X with a function rX defined on a domain set partitioned into l singleton sets $\{X_1\}, \{X_2\}, \dots, \{X_l\}$ and taking integer values. Distinguishing between the l versions of variable X can be done by checking the argument of function rX for the partition it belongs to. All of these are easily implementable in Event-B. The complexity of the refinement is transferred in this way to using functions defined on partitioned sets, rather than using integer variables, and in specifying more complex event guards to check the argument of the function. Crucially, this makes writing large refined models possible and it leaves checking the model consistency feasible, at least on the ErbB model we tested on, with more than 1000 reactions in its refined version. We discuss this technique in concrete details in the case of the binary reactions: binding $A + B \rightarrow C$ and dimerization $2A \rightarrow C$.

Consider the binding reaction $A + B \rightarrow C$. Assume that the variable A is refined into two versions, A_0 and A_1 and the variable C is refined into C_0, C_1 . The binding reaction is then refined into two versions of it $A_i + B \rightarrow C_i$, with $i = 0, 1$. Rather than writing two events in the refined Event-B model, one for each of them, we can write a single encompassing event in the following way. We replace variables A and C with the functions $rA : \{\{A_0\}, \{A_1\}\} \rightarrow \mathbb{N}$ and $rC : \{\{C_0\}, \{C_1\}\} \rightarrow \mathbb{N}$, resp., where A_0, A_1, C_0, C_1 are new constants. The refinement of the binding reaction can be described through a single event, shown in Table 2. The key observation is that distinguishing between the two refinements of the reaction is done in a uniform way through the guards of the event. The main guard in this case is **@grd3**, which models that the specific form A_0 of variable A corresponds to the specific form C_0 of variable C and that A_1 corresponds to C_1 . These two cases are covered in **@act1** through parameters a and c , a technique that allows a uniform specification of the two refined versions of the reaction through **@act1** and **@act3**. The gluing invariants specifying the conditions for the consistency of this refinement are $A = rA(A_0) + rA(A_1)$ and $C = rC(C_0) + rC(C_1)$.

A dimerization reaction $2A \rightarrow C$ can be refined in the same way. Assume that the variable A is refined into two versions A_0 and A_1 and the variable C is refined into three versions C_0, C_1 , and C_2 . The dimerization reaction is then refined into three versions of it $A_i + A_j \rightarrow C_{i+j}$, with $0 \leq i \leq j \leq 1$. All three of them can be described through a single event, shown in Table 3. The event is written similarly as for the binding reaction: variable A is replaced with a function $rA : \{\{A_0\}, \{A_1\}\} \rightarrow \mathbb{N}$ and variable C is replaced by function $rC : \{\{C_0\}, \{C_1\}, \{C_2\}\} \rightarrow \mathbb{N}$. The consistency of the refinement is specified through the gluing invariants shown in Table 3.

A compact Event-B model for the heat shock response

The heat shock response is a well conserved cellular mechanism to react to sudden increases in temperature, in an effort to protect the cell against damage to cellular structures and essential functions¹⁷. It also has a key role in oncogenesis and cell death^{18,19}. The reaction network model²⁰ describing the heat shock response is shown in Table 4 and it consists of 10 variables and 17 irreversible reactions. It was described through an Event-B model¹⁴ with each (uni-directional) reaction leading to its own event in the model. We are interested in a refinement of this model where the heat shock factor (hsf) variable is replaced with two variants $rhsf^{(0)}$ and $rhsf^{(1)}$, corresponding to the phosphorylation status of site S230 of the heat shock factor. Consequently, this leads us to also introduce simultaneously the refinement of all the other forms of hsf:

$$\begin{aligned} hsp: hsf &\rightarrow \{hsp: rhsf^{(0)}, hsp: rhsf^{(1)}\}, \\ hsf_2 &\rightarrow \{rhsf_2^{(0)}, rhsf_2^{(1)}, rhsf_2^{(2)}\}, \\ hsf_3 &\rightarrow \{rhsf_3^{(0)}, rhsf_3^{(1)}, rhsf_3^{(2)}, rhsf_3^{(3)}\}, \\ hsf_3: hse &\rightarrow \{rhsf_3^{(0)}: hse, rhsf_3^{(1)}: hse, rhsf_3^{(2)}: hse, rhsf_3^{(3)}: hse\}. \end{aligned}$$

This refinement leads to 22 variables and 57 reactions in the reaction network model. We avoid this increase in the model size using our approach: the refined model can be written in Event-B through 10 variables and 17 events, the same as in the basic model. All proof obligations were discharged automatically in Rodin (some of them required a change of the default theorem prover). The comparison between the size of the basic and the refined variants of the HSR model is in Table 5. The full model can be downloaded at <https://tinyurl.com/cf7vxs2e>.

A compact Event-B model for the ErbB signaling pathway

The ErbB signaling pathway is a very well studied evolutionary pathway, essential in the growth and expansion of organs and of the central nervous system^{21,22}. Its main role is to induce, through the cellular membrane, a signal instigating the cell's growth and differentiation. This pathway is often overly active in various types of cancer and has been used for a long time as a therapeutic target²³.

The reaction network model^{24–26} of the ErbB signaling pathway consists of a basic model consisting of 242 reactions, extended then with the phosphorylation details of the epidermal growth factor receptor (EGFR) and the epidermal growth factor (EGF) to 1320 reactions. The basic model was described in Event-B²⁷ and we extend it here to the full model. The epidermal growth factor receptor is refined into the four receptor members of the ErbB family and the epidermal growth factor is refined into two variants:

$$\begin{aligned} EGFR &\rightarrow \{ErbB1, ErbB2, ErbB3, ErbB4\}; \\ EGF &\rightarrow \{EGF, HRG\}. \end{aligned}$$

This leads to a refinement of EGF-EGFR into the following eight variants:

$$\begin{aligned} EGF-EGFR &\rightarrow \{EGF-ErbB1, EGF-ErbB2, EGF-ErbB3, EGF-ErbB4, \\ &\quad HRG-ErbB1, HRG-ErbB2, HRG-ErbB3, HRG-ErbB4\}. \end{aligned}$$

A similar refinement is also applied for their phosphorylated versions and for their homodimers. They were described in our Event-B model through 242 events, as many as the basic model. All proof obligations were discharged automatically by Rodin (some of them required a change of the default theorem prover), with the comparison between the basic and the refined variants shown in Table 5. The full model can be downloaded at <https://tinyurl.com/cf7vxs2e>.

Discussion

Modeling complex biological processes is a challenging systems engineering task. A solution to addressing its complexity is to use refinement and start modeling from a conceptually simpler (more abstract) model that is consistent. We can then gradually add more details in a correctness-by-construction approach, so that the most detailed (concrete) model is mathematically proved to be consistent. In this approach, the challenge is to prove both the consistency of the basic model and that of the refinement process. The Event-B-based approach we propose in this paper addresses both of these challenges. Through the Rodin toolset, the models get a much needed automatic computer-aided verification of their consistency. The method we proposed, to implement the data refinement through the use of integer-valued functions, allowed us to preserve the size of the models in terms of the number of variables and events, with the increase in complexity reflected in the more complex guards, in the higher number of invariants, and in the switch from integer variables to integer-valued functions. This led to more proof obligations.

Nevertheless, in the proof-of-concept examples we modeled, all the proof obligations were automatically discharged in Rodin, albeit some of them needed a switch from the default theorem prover to another one in the Rodin installation. Even in this case, no input from the user was needed in proving the model consistency.

Additional refinements and replacing refinement relations with more detailed ones can be done in the same way. They only involve replacing some integer variables with integer-valued functions as we discussed in this paper, or replacing the domain set of an integer-valued function with a larger domain set (reflecting the more detailed refinement). This means that the number of variables and events will remain the same (the number of proof obligations will increase). The refinement of the ErbB reaction network, going from 242 reactions to 1320 reactions, left unchanged the number of variables and events, but increased the number of proof obligations by about 50%. The number of variables and events of the model will certainly increase if other operations than refinement are used in the model construction, such as model composition or model union.

Specifying the refinement relation in a consistent manner is essential for our approach. This means that refining a variable A standing for a molecular species should lead to the corresponding refinements of all complexes that it is part of: complexes $A : B$, dimers A_2 , trimers A_3 , etc. Deducing the full consistent refinement relation based on specifying the refinement of a single variable can be done based on⁶ by specifying the relationship between the compound variables and the basic variables.

Scalable modeling remains a challenging problem. The approach we introduced here is a partial solution to the problem in the case of modeling based on data refinement. Event-B has most characteristics required for a mathematical modeling language to facilitate model reuse in systems biology¹: it is modular, human-readable, it is hybrid (it allows for continuous and for discrete mathematics semantics, aspects beyond the focus of this paper), open, and declarative (but not graphical). Significant is also the bridge between the reaction network modeling and Event-B-based modeling, which brings to biomodeling a new framework for computer-aided verification of model consistency. This is relevant to a wide variety of models, beyond the two proof-of-concept examples in this paper.

Methods and models

Event-B

Event-B²⁸ is a state-based formal method, building on the earlier formalisms the B-method²⁹ and the action systems³⁰. The system state in Event-B is described by the values of *variables* and the state changes are modeled using *events*. The variable types and the model properties that must hold during system execution are defined as *invariants*. The initial system state is described with a specific event named *Initialisation*. An event can contain *parameters*, a *guard* and an *action*. The guard is a logical predicate on the variables and parameters, describing the conditions under which the action is enabled to take place. The action describes the updates it yields to the variables when it is executed. If two or more events are enabled at the same time, then one is non-deterministically chosen and executed. If two events do not update each other's variables, then they can be executed in any order and we can consider that they are executed in parallel. The variables and events in an Event-B model are contained in a *machine*, also referred to as the “dynamic part” of the model. An Event-B machine can *see* one or more *contexts*, also known as the “static part” of the model. A context contains definitions of constants, sets, as well as axioms about them. A general structure of an Event-B model, made out of machine M and context C is presented in Table 6.

A key concept in formal modeling with Event-B is that of *refinement*¹⁵, which allows the modeler to start from a simple model of the system and then gradually introduce more details, in the form of new events, variables or context data. Event-B has two types of refinements: *superposition refinement* and *data refinement*. Superposition refinement^{3,4} is the term used when new variables and events are added to the existing model. The validity of model is preserved by making sure that newly added variables and events do not contradict the previous events in any of the preceding models. In *data refinement*², some variables in the abstract machine are replaced by other variables in the refined machine; in this case, a *gluing invariant* is added in the refined machine to define the relation between the abstract variables and the newly introduced, concrete ones. Refinement in Event-B has been used to model numerous protocols and systems, including smart cash card systems³¹, vehicle platoons³², topology discovery in graphs³³, self-recovery in sensor-actor networks³⁴, spacecraft systems³⁵, coordination in peer-to-peer networks³⁶, smart grid recoverability³⁷, proactive routing in wireless networks³⁸, reaction networks¹⁴, etc.

Event-B benefits from the tool support of the Eclipse-based Rodin platform¹⁶. Rodin allows to edit the model, to prove properties of the model, to animate the model and even allows model checking. Proving in Event-B employs several proof engines to automatically prove the different properties of the model. Rodin automatically generates all proof obligations in the form of sequents; these need to be discharged in order for the different properties (e.g., invariance, termination, or refinement) to hold. The internal provers attempt to discharge these proof obligations automatically. The remaining ones can be tackled using the interactive prover, with input from the modeler, for instance by adding extra assumptions or choosing different proof strategies. The model properties are proved at the level of the events, rather than on the systems level. Having a property that cannot be proved for some event can point out to errors in that event. The modeler then has a chance to edit the model to address the issue. Such interleaving between modeling and proving is an important aspect of working with Event-B and the Rodin platform and is quite similar to the compilation of programs¹⁶.

The heat shock response

The heat-shock response is a central, well-conserved, cellular-level regulatory mechanism^{39,40}. Proteins are folded in three-dimensional shapes and the fold determines whether it can achieve its functionality (e.g., bind to a certain site on a DNA molecule or on another protein). Protein folding is a dynamical process, continuously influenced by many factors, such as chemical modifications of the amino-acids forming the protein (e.g., phosphorylation, acetylation, sumoylation) and properties of the environment (e.g., temperature, radiation, heavy metals). Misfolded proteins quickly form large protein bundles that are detrimental to the normal physiology of a cell and eventually lead to cell death. The heat shock response is one of the stress response mechanisms of a cell, aiming to limit the accumulation of misfolded proteins and assisting misfolded proteins to regain their natural fold. The heat shock response synthesizes a group of proteins called heat shock proteins (hsps) that act as molecular chaperones for the misfolded proteins and support their recovery from stress. This is achieved either by repairing the damaged proteins or by degrading them, thus restoring protein homeostasis and promoting cell survival. Without such a mechanism, misfolded proteins will form plaque, which is the hallmark of many neurological diseases.

The basic model we discuss for the eukaryotic heat shock response²⁰ is summarized in Table 4. When the temperature increases, proteins $prot$ begin to misfold, namely transform into mfp (reaction (10)). The heat shock proteins have a high affinity to bind to the misfolded proteins, acting as chaperones and forming $hsp:mfp$ complexes (reaction (11)). Then, the complex $hsp:mfp$ can transform back into the original protein $prot$, freeing the heat shock factor protein hsp too (reaction (12)). The hsp is synthesized as follows. The heat shock factor (hsf) binds in a trimmer form to the hsp 's gene promoter (called the heat shock element hse) (reactions (1)–(3) in Table 4). The formed $hsf_3:hse$ then produces the hsp proteins (reaction (4)). These tend to combine with hsf and stay in inactive state as $hsp:hsf$ complexes (right arrow in reaction (5), as well as reactions (6)–(8)). Once the temperature increases and more hsp are becoming chaperones for mfp , less are available for forming $hsp:hsf$ complexes and the balance changes: the left arrow in the reaction (5) is activated. Finally, hsps can also degrade (reaction (9)).

The phosphorylation level of the heat shock factors has a key contribution to its activation and consequently to the effectiveness of the heat shock response⁴¹. One site in particular, S230, becomes phosphorylated only during heat shock response and drives its activity⁴². Following the model of⁴³, we introduce two variants of hsf : one where S230 is unphosphorylated and the other where S230 is phosphorylated. This cascades into several variants of all variables that hsf is a part of: hsf_2 (phosphorylation level 0, 1, or 2), hsf_3 and $hsf_3:hse$ (phosphorylation level 0, 1, 2, or 3), and $hsp:hsf$ (phosphorylation level 0 or 1). These phosphorylation-based variants replace the generic variables in all reactions, in all possible combinations, leading to an increase in the size of the model⁴³.

The ErbB signaling pathway

We discuss briefly here the key functionality of the ErbB signaling pathway^{24–26} using a highly simplified presentation. The epidermal growth factors (EGF) are a family of proteins that signal to cells to grow and differentiate. They do that by binding to ligand proteins embedded in the cellular membrane – the epidermal growth factor receptors (EGFR). Once bound, the complex dimerizes and then gets phosphorylated. This then activates other (MAPK and ERK) signaling pathways. All of these activations are done step by step through a cascade of reactions, whose effect is the activation of some proteins, that then participate in other reactions activating other proteins, etc.

The model of the ErbB signaling pathway²⁴ that we follow in this paper is a revised version of the two earlier models^{25,26}. It is first presented on a more generic level, along the lines briefly described above. This initial model consists however of 242 irreversible reactions. The full model is then introduced essentially by differentiating between the four members of the EGFR family (ErbB1 (also known as EGFR), ErbB2, ErbB3, ErbB4) and the two members of the EGF family (EGF and HRG). Adding these details leads to many more variables in the model. For example, the complex $EGF : EGFR$ is replaced by 8 different variants of it. The full reaction network model²⁴ has 1320 reactions.

References

1. Schölzel, C., Blesius, V., Ernst, G. & Dominik, A. Characteristics of mathematical modeling languages that facilitate model reuse in systems biology: a software engineering perspective. *npj Syst. Biol. Appl.* **7**, 27, DOI: [10.1038/s41540-021-00182-w](https://doi.org/10.1038/s41540-021-00182-w) (2021).
2. Hoare, C., Jifeng, H. & Sanders, J. Prespecification in data refinement. *Inf. Process. Lett.* **25**, 71–76, DOI: [https://doi.org/10.1016/0020-0190\(87\)90224-9](https://doi.org/10.1016/0020-0190(87)90224-9) (1987).
3. Back, R.-J. & Sere, K. Superposition refinement of reactive systems. *Formal Aspects Comput.* **8**, 324–346 (1996).
4. Katz, S. A superimposition control construct for distributed systems. *ACM Trans. Program. Lang. Syst.* **15**, 337–356, DOI: [10.1145/169701.169682](https://doi.org/10.1145/169701.169682) (1993).
5. Murphy, E., Danos, V., F eret, J., Krivine, J. & Harmer, R. *Rule-Based Modeling and Model Refinement*, chap. 4, 83–114 (John Wiley & Sons, Ltd, 2010). <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470556757.ch4>.

6. Gratie, C. & Petre, I. Complete characterization for the fit-preserving data refinement of mass-action reaction networks. *Theor. Comput. Sci.* **641**, 11 – 24, DOI: <http://dx.doi.org/10.1016/j.tcs.2016.03.027> (2016).
7. Paulevé, L., Kolčák, J., Chatain, T. & Haar, S. Reconciling qualitative, abstract, and scalable modeling of biological networks. *Nat. Commun.* **11**, 4256, DOI: [10.1038/s41467-020-18112-5](https://doi.org/10.1038/s41467-020-18112-5) (2020).
8. Iancu, B., Czeizler, E., Czeizler, E. & Petre, I. Quantitative refinement of reaction models. *Int. J. Unconv. Comput.* **8**, 529–550 (2012).
9. Aceto, L. & Hennessy, M. Towards action-refinement in process algebras. In [1989] *Proceedings. Fourth Annual Symposium on Logic in Computer Science*, 138–145, DOI: [10.1109/LICS.1989.39168](https://doi.org/10.1109/LICS.1989.39168) (1989).
10. Danos, V., Feret, J., Fontana, W., Harmer, R. & Krivine, J. Rule-based modelling and model perturbation. In *Transactions on Computational Systems Biology XI*, 116–137 (Springer, 2009).
11. Suzuki, I. & Murata, T. A method for stepwise refinement and abstraction of petri nets. *J. Comput. Syst. Sci.* **27**, 51–76, DOI: [https://doi.org/10.1016/0022-0000\(83\)90029-6](https://doi.org/10.1016/0022-0000(83)90029-6) (1983).
12. Padberg, J. & Urbášek, M. Rule-based refinement of petri nets: A survey. In Ehrig, H., Reisig, W., Rozenberg, G. & Weber, H. (eds.) *Petri Net Technology for Communication-Based Systems: Advances in Petri Nets*, 161–196, DOI: [10.1007/978-3-540-40022-6_9](https://doi.org/10.1007/978-3-540-40022-6_9) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003).
13. Gratie, D.-E. & Petre, I. Hiding the combinatorial state space explosion of biomodels through colored petri nets. *Annals Univ. Buchar.* **LXI**, 23–41 (2014).
14. Sanwal, U., Petre, L. & Petre, I. Stepwise construction of a metabolic network in Event-B. *Comput. Biol. Medicine* **91**, 1–12 (2017).
15. Abrial, J.-R. *Modeling in Event-B: System and Software Engineering* (Cambridge University Press, New York, NY, USA, 2010), 1st edn.
16. Abrial, J.-R. *et al.* Rodin: an open toolset for modelling and reasoning in Event-B. *STTT* **12**, 447–466 (2010).
17. Richter, K., Haslbeck, M. & Buchner, J. The heat shock response: Life on the verge of death. *Mol. Cell* **40**, 253–266, DOI: <https://doi.org/10.1016/j.molcel.2010.10.006> (2010).
18. Jolly, C. & Morimoto, R. I. Role of the heat shock response and molecular chaperones in oncogenesis and cell death. *JNCI: J. Natl. Cancer Inst.* **92**, 1564–1572, DOI: [10.1093/jnci/92.19.1564](https://doi.org/10.1093/jnci/92.19.1564) (2000).
19. Gomez-Pastor, R., Burchfiel, E. T. & Thiele, D. J. Regulation of heat shock transcription factors and their roles in physiology and disease. *Nat. Rev. Mol. Cell Biol.* **19**, 4–19, DOI: [10.1038/nrm.2017.73](https://doi.org/10.1038/nrm.2017.73) (2018).
20. Petre, I. *et al.* A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Nat. Comput.* **10**, 595–612 (2011).
21. Citri, A. & Yarden, Y. Egf–erbb signalling: towards the systems level. *Nat. Rev. Mol. Cell Biol.* **7**, 505–516, DOI: [10.1038/nrm1962](https://doi.org/10.1038/nrm1962) (2006).
22. Hynes, N. E. & MacDonald, G. Erbb receptors and signaling pathways in cancer. *Curr Opin Cell Biol* **21**, 177–184, DOI: [10.1016/j.ceb.2008.12.010](https://doi.org/10.1016/j.ceb.2008.12.010) (2009).
23. Santarpia, L., Lippman, S. M. & El-Naggar, A. K. Targeting the mapk-ras-raf signaling pathway in cancer therapy. *Expert Opin Ther Targets* **16**, 103–119, DOI: [10.1517/14728222.2011.645805](https://doi.org/10.1517/14728222.2011.645805) (2012).
24. Hornberg, J. J. *et al.* Control of MAPK signalling: from complexity to what really matters. *Oncogene* **24**, 5533–5542 (2005).
25. Kholodenko, B. N., Demin, O. V., Moehren, G. & Hoek, J. B. Quantification of short term signaling by the epidermal growth factor receptor. *J. Biol. Chem.* **274**, 30169–30181 (1999).
26. Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D. & Müller, G. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. *Nat. Biotechnol.* **20**, 370 EP – (2002).
27. Iancu, B., Sanwal, U., Gratie, C. & Petre, I. Refinement-based modeling of the ErbB signaling pathway. *Comput. Biol. Medicine* **106**, 91 – 96, DOI: <https://doi.org/10.1016/j.combiomed.2019.01.016> (2019).
28. Abrial, J.-R. & Hallerstede, S. Refinement, decomposition, and instantiation of discrete models: Application to Event-B. *Fundamenta Informaticae* **Vol. 77, nr 1,2**, 1–28 (2007).
29. Abrial, J.-R. *The B-book: Assigning Programs to Meanings* (Cambridge University Press, New York, NY, USA, 1996).

30. Back, R.-J. & Kurki-Suonio, R. Decentralization of process nets with centralized control. In *Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing*, PODC '83, 131–142, DOI: [10.1145/800221.806716](https://doi.org/10.1145/800221.806716) (ACM, New York, NY, USA, 1983).
31. Butler, M. & Yadav, D. An incremental development of the mondex system in Event-B. *Formal Aspects Comput.* **20**, 61–77, DOI: [10.1007/s00165-007-0061-4](https://doi.org/10.1007/s00165-007-0061-4) (2007).
32. Lanoix, A. Event-B specification of a situated multi-agent system: Study of a platoon of vehicles. In *Theoretical Aspects of Software Engineering*, 297–304, DOI: <http://doi.ieeecomputersociety.org/10.1109/TASE.2008.39> (IEEE Computer Society, Los Alamitos, CA, USA, 2008).
33. Hoang, T. S., Kuruma, H., Basin, D. & Abrial, J.-R. Developing topology discovery in Event-B. In Leuschel, M. & Wehrheim, H. (eds.) *Integrated Formal Methods*, vol. LNCS 5423, 1–19, DOI: [10.1007/978-3-642-00255-7_1](https://doi.org/10.1007/978-3-642-00255-7_1) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009).
34. Kamali, M., Laibinis, L., Petre, L. & Sere, K. Self-recovering sensor-actor networks. In *Proceedings Ninth International Workshop on the Foundations of Coordination Languages and Software Architectures, FOCLASA 2010, Paris, France, 4th September 2010.*, 47–61, DOI: [10.4204/EPTCS.30.4](https://doi.org/10.4204/EPTCS.30.4) (2010).
35. Salehi Fathabadi, A., Rezazadeh, A. & Butler, M. Applying atomicity and model decomposition to a space craft system in Event-B. In Bobaru, M., Havelund, K., Holzmann, G. J. & Joshi, R. (eds.) *NASA Formal Methods*, 328–342, DOI: [10.1007/978-3-642-20398-5_24](https://doi.org/10.1007/978-3-642-20398-5_24) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011).
36. Petre, L., Sandvik, P. & Sere, K. Node coordination in peer-to-peer networks. In *Coordination Models and Languages - 14th International Conference, COORDINATION 2012, Stockholm, Sweden, June 14-15, 2012. Proceedings*, 196–211, DOI: [10.1007/978-3-642-30829-1_14](https://doi.org/10.1007/978-3-642-30829-1_14) (2012).
37. Horsmanheimo, S. *et al.* On proving recoverability of smart electrical grids. In *NASA Formal Methods - 6th International Symposium, NFM 2014, Houston, TX, USA, April 29 - May 1, 2014. Proceedings*, 77–91, DOI: [10.1007/978-3-319-06200-6_6](https://doi.org/10.1007/978-3-319-06200-6_6) (2014).
38. Kamali, M., Höfner, P., Kamali, M. & Petre, L. Formal analysis of proactive, distributed routing. In *Software Engineering and Formal Methods - 13th International Conference, SEFM 2015, York, UK, September 7-11, 2015. Proceedings*, 175–189, DOI: [10.1007/978-3-319-22969-0_13](https://doi.org/10.1007/978-3-319-22969-0_13) (2015).
39. Powers, M. V. & Workman, P. Inhibitors of the heat shock response: Biology and pharmacology. *FEBS Lett.* **581**, 3758–3769, DOI: <http://dx.doi.org/10.1016/j.febslet.2007.05.040> (2007). Cellular Stress.
40. Voellmy, R. Transduction of the stress signal and mechanisms of transcriptional regulation of heat shock/stress protein gene expression in higher eukaryotes. *Critical reviews eukaryotic gene expression* **4**, 357–401 (1994).
41. Kline, M. P. & Morimoto, R. I. Repression of the heat shock factor 1 transcriptional activation domain is modulated by constitutive phosphorylation. *Mol Cell Biol* **17**, 2107–2115, DOI: [10.1128/MCB.17.4.2107](https://doi.org/10.1128/MCB.17.4.2107) (1997).
42. Holmberg, C. I. *et al.* Phosphorylation of serine 230 promotes inducible transcriptional activity of heat shock factor 1. *The EMBO journal* **20**, 3800–3810, DOI: [10.1093/emboj/20.14.3800](https://doi.org/10.1093/emboj/20.14.3800) (2001).
43. Czeizler, E., Rogojin, V. & Petre, I. The phosphorylation of the heat shock factor as a modulator for the heat shock response. *IEEE/ACM Trans Comput. Biol Bioinform* **9**, 1326–1337, DOI: [10.1109/TCBB.2012.66](https://doi.org/10.1109/TCBB.2012.66) (2012).

Acknowledgments

Ion Petre was partially supported by the Romanian Ministry of Education and Research, CCCDI – UEFISCDI, project number PNIII-P2-2.1-PED-2019-2391, within PNCDI III.

Author contributions statement

U.S., L.P. and I.P. designed the study. U.S., T.S.H. and L.P. wrote and analyzed the models. U.S. and I.P. collected and curated the data. All authors analyzed the results. They were all involved in drafting the article, and have reviewed and approved the final version of the manuscript.

Additional information

Competing interests. The author(s) declare no competing interests.

Data availability. All models are available at <https://tinyurl.com/cf7vxs2e>.

Tables

Table 1. The general form of an Event-B model for a reaction network. The event r models reaction (1), implementing the result of that reaction being triggered.

<p>VARIABLES X_1, X_2, \dots, X_n</p> <p>INVARIANTS</p> <p>@inv1 $X_1 \in \mathbb{N}$</p> <p>@inv2 $X_2 \in \mathbb{N}$</p> <p>...</p> <p>@invn $X_n \in \mathbb{N}$</p>	<p>Event r</p> <p>WHERE</p> <p>@grd1 $X_1 \geq m_1$</p> <p>@grd2 $X_2 \geq m_2$</p> <p>...</p> <p>@grdn $X_n \geq m_n$</p> <p>THEN</p> <p>@act1 $X_1 := X_1 + (m'_1 - m_1)$</p> <p>@act2 $X_2 := X_2 + (m'_2 - m_2)$</p> <p>...</p> <p>@actn $X_n := X_n + (m'_n - m_n)$</p> <p>END</p>
<p>INITIALISATION</p> <p>@act1 $X_1 = init_1$</p> <p>@act2 $X_2 = init_2$</p> <p>...</p> <p>@actn $X_n = init_n$</p>	

Table 2. Size-preserving refinement of a binding reaction in Event-B.

<p>Context for the refined binding reaction</p> <p>Constants A_0, A_1, C_0, C_1</p> <p>Sets sA, sC</p> <p>Axioms</p> <p>partition($sA, \{A_0\}, \{A_1\}$)</p> <p>partition($sC, \{C_0\}, \{C_1\}$)</p>	<p>The event for the refined binding reaction</p> <p>GLUING INVARIANTS</p> <p>@inv1 $A = rA(A_0) + rA(A_1)$</p> <p>@inv2 $C = rC(C_0) + rC(C_1)$</p> <p>ANY a, c</p> <p>WHERE</p> <p>@grd1 $rA(a) \geq 1$</p> <p>@grd2 $B \geq 1$</p> <p>@grd3 $(a = A_0 \wedge c = C_0) \vee (a = A_1 \wedge c = C_1)$</p> <p>THEN</p> <p>@act1 $rA(a) := rA(a) - 1$</p> <p>@act2 $B := B - 1$</p> <p>@act3 $rC(c) := rC(c) + 1$</p> <p>END</p>
--	---

Table 3. Size-preserving refinement of a dimerization reaction in Event-B. The symbol \Leftarrow is used in Event-B to specify that the function rA is modified for its arguments A_0 and A_1 .

<p>Context... Constants A_0, A_1, C_0, C_1, C_2 Sets sA, sC Axioms $\text{partition}(sA, \{A_0\}, \{A_1\})$ $\text{partition}(sC, \{C_0\}, \{C_1\}, \{C_2\})$</p>	<p>The event for the refined dimerization reaction</p> <p>GLUING INVARIANTS $\text{@inv1 } A = rA(A_0) + rA(A_1)$ $\text{@inv2 } C = rC(C_0) + rC(C_1) + rC(C_2)$</p> <p>ANY a_0, a_1, c WHERE $\text{@grd1 } a_0 \in \mathbb{N}$ $\text{@grd2 } a_1 \in \mathbb{N}$ $\text{@grd3 } a_0 + a_1 = 2$ $\text{@grd4 } rA(A_0) \geq a_0$ $\text{@grd5 } rA(A_1) \geq a_1$ $\text{@grd6 } a_0 = 2 \Rightarrow c = C_0$ $\text{@grd7 } a_0 = 1 \Rightarrow c = C_1$ $\text{@grd8 } a_0 = 0 \Rightarrow c = C_2$ THEN $\text{@act1 } rA := rA \Leftarrow \{$ $\quad A_0 \mapsto rA(A_0) - a_0,$ $\quad A_1 \mapsto rA(A_1) - a_1\}$ $\text{@act2 } rC(c) := rC(c) + 1$ END</p>
--	--

Table 4. The molecular model for the eukaryotic heat shock response proposed in ²⁰.

(1) $2 \text{ hsf} \rightleftharpoons \text{hsf}_2$	(7) $\text{hsp} + \text{hsf}_3 \rightarrow \text{hsp}:\text{hsf} + 2 \text{ hsf}$
(2) $\text{hsf} + \text{hsf}_2 \rightleftharpoons \text{hsf}_3$	(8) $\text{hsp} + \text{hsf}_3:\text{hse} \rightarrow \text{hsp}:\text{hsf} + 2 \text{ hsf} + \text{hse}$
(3) $\text{hsf}_3 + \text{hse} \rightleftharpoons \text{hsf}_3:\text{hse}$	(9) $\text{hsp} \rightarrow \emptyset$
(4) $\text{hsf}_3:\text{hse} \rightarrow \text{hsf}_3:\text{hse} + \text{hsp}$	(10) $\text{prot} \rightarrow \text{mfp}$
(5) $\text{hsp} + \text{hsf} \rightleftharpoons \text{hsp}:\text{hsf}$	(11) $\text{hsp} + \text{mfp} \rightleftharpoons \text{hsp}:\text{mfp}$
(6) $\text{hsp} + \text{hsf}_2 \rightarrow \text{hsp}:\text{hsf} + \text{hsf}$	(12) $\text{hsp}:\text{mfp} \rightarrow \text{hsp} + \text{prot}$

Table 5. The heat shock response and the ErbB models statistics. The number of variables and events is the same for the basic and the refined model. Some of the proof obligations of the refined model required another internal theorem prover than the default one to get them automatically discharged.

	Heat shock response model		ErbB signaling pathway	
	Basic	Refined	Basic	Refined
Molecular species	10	20	110	394
Molecular reactions	17	57	242	1320
Variables	10	10	110	110
Events	17	17	242	242
Invariants	13	18	110	110
Proof obligations discharged by the default prover	79	73	794	15
Proof obligations discharged by other internal provers	0	62	154	1260

Table 6. General structure of machine M and context C in Event-B

