Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

VERIFICATION AND TESTING OF BIOLOGICAL MODELS

Allan Clark

University of Edinburgh
School of Informatics
Edinburgh, UK

Stephen Gilmore

University of Edinburgh
School of Informatics
Edinburgh, UK

Jane Hillston

University of Edinburgh
School of Informatics
Edinburgh, UK

Peter Kemper

College of William and Mary
Department of Computer Science
Williamsburg, VA 23187, USA

ABSTRACT

Simulation modeling in systems biology embarks on discrete event simulation only for cases of small cardinalities of entities and uses continuous simulation otherwise. Modern modeling environments like Bio-PEPA support both types of simulation within a single modeling formalism. Developing models for complex dynamic phenomena is not trivial in practice and requires careful verification and testing. In this paper, we describe relevant steps in the verification and testing of a TNFα-mediated NF-κB signal transduction pathway model and discuss to what extent automated techniques help a practitioner to derive a suitable model.

1 INTRODUCTION

Modeling the dynamics of a biological system is not trivial and much research has gone into the development of stochastic discrete event system models and continuous simulation models, where the latter are mostly described by systems of ordinary differential equations. The challenges in this area are manifold, starting from the generation of experimental data in vivo, the derivation of conceptual models and a theoretical understanding of the dynamics of a system, its constituents and mechanisms and the development of executable models.

In this paper we investigate the role of automated support in the verification and testing of stochastic discrete event simulation models in this context. Designing and implementing a large simulation model is a complex task in any circumstance, and the inclusion of stochastic elements can cause the model to exhibit non-intuitive behaviour, making it difficult to know whether output from the model is "*as expected*". However, in the context of biological models there are several factors which exacerbate the problem.

Biological models play a dual role of documenting current knowledge about the system or mechanism under study and providing the basis for study of dynamic behaviour. Thus the construction of models is intimately related to the local availability, or not, of experimental data relating to the system under study. As a result, new models may be built as refinements of old ones taken from the literature; model structure may be taken at least in part from a collated database such as KEGG or the BioModels database; model parameters may also be derived from data in a database such as BRENDA. Moreover, when experimental data is available automated Bayesian approaches may be used to generate the reaction network. Furthermore the availability of accessible workflows which automate processes such as database search and Bayesian inference is accelerating the trend towards semi-automatic model construction. Even if originally handcrafted, a model may subsequently be modified and extended using evolutionary algorithms and the increased use of version control tools for collaborative model development as favoured in biomodel engineering approaches (Breitling et al. 2010) may also lead to the development of larger and more complex models than previously seen. This all contributes to the situation where the link between the model and the scientist is tenuous since the model user may not be entirely responsible for the model's construction.

For reasons such as these we think that it is timely for model verification and testing to receive increased attention from researchers who are considering the next generation of biological modelling tools. We focus on the Bio-PEPA Eclipse Plug-in tool suite (Duguid et al. 2009), extending it with several features to assist the model user in verifying a model before embarking on *in silico* experimentation. In particular we have written a special-purpose implementation of the Gillespie algorithm which writes its results in the form of a trace which can be read by Traviando (Kemper and Tepper 2009). Traviando is a general purpose trace analyzer and model checker, which has been developed primarily to assist in the debugging of simulation models. It had not previously been applied to biological models.

978-1-4244-9865-9/10/$26.00 ©2010 IEEE

To illustrate our approach we consider the first scenario mentioned above, the case of a model taken from the literature. When a model has been previously published it is tempting to start by seeking to reproduce the corresponding results from the literature. A common experience is that this is less straightforward than expected and typing errors, underspecified formalisms, omission of necessary details and the like make this work tedious and time consuming. In many cases the models are large and complex and the form of model supplied may not have been generated with human readability in mind. This means that uncovering the nature of unexpected behaviour, never mind its source, can be extremely difficult. Moreover, this assumes that the given model is already verified and correct. For our chosen model in this study, a model of the TNFα-mediated NF-κB signal transduction pathway published in (Cho et al. 2003), we elaborate on several steps to obtain a consistent model. Following a famous remark that adopting the right point of view gains you 80 additional IQ points (attributed to Alan Kay), we show several views of this model, reformulated in Bio-PEPA, that make it simple to recognize certain modeling errors and to achieve a model that passes a face validity check.

The rest of the paper is structured as follows. In Section 2, we describe the published pathway model. In Section 3, we briefly recall the main features of the Bio-PEPA modeling formalism and its corresponding modeling framework. In Section 4, we explain how the model of Cho et al. (2003) has been reformulated as a Bio-PEPA model. Section 5 illustrates our concept of views that is used to identify several errors in the published model. This section gives some guidance to a modeler on how to perform consistency checks on Bio-PEPA models and describes several implemented fully automated approaches to increase a modelers productivity. In Section 6, we perform a face validity check and compare simulation results for different variants of the model of Cho et al. (2003). Related work is discussed in Section 7 and we conclude in Section 8.

2 BIOLOGICAL PROBLEM OF INTEREST

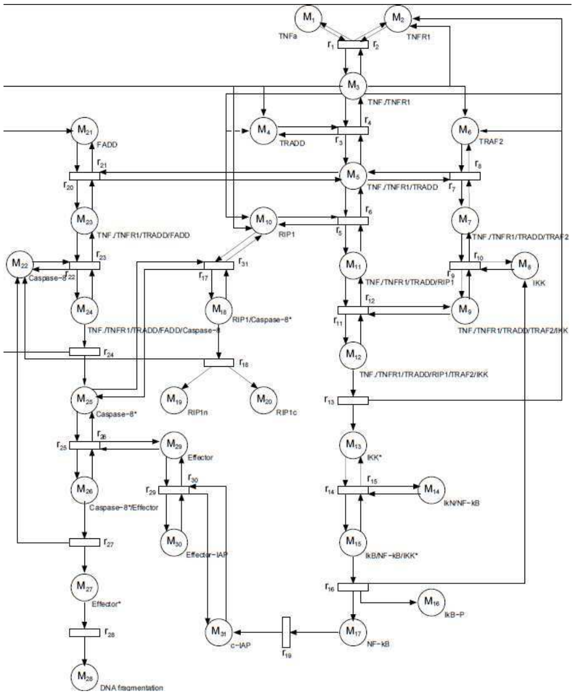

Figure 1: Circuit diagram of the TNFα-mediated NF-κB signal transduction pathway model

Our starting point for this paper is a published study of the TNFα-mediated NF-κB signal transduction pathway (Cho et al. 2003). This pathway plays an important role in immunity and inflammation, and in the control of cell proliferation, cell differentiation, and apoptosis (programmed cell death). The model of the pathway which we use as an illustration here is concise, but not trivial. There are 31 chemical species and 31 reactions in the model. The model is presented in (Cho et al. 2003) as a system of differential equations and also illustrated using a graphical representation (termed a "circuit diagram") in a style reminiscent of a Petri net (reproduced in Figure 1).

We selected the model for a number of reasons. The presentation of the model given in the original paper was clear and complete — a full set of ordinary differential equations was included in the paper. Furthermore the inclusion of the circuit diagram meant that we had an alternative view of the authors' intention for the model in addition to the mathematical representation. Finally one of our authors had some familiarity with the underlying biology through having conducted related modelling studies in the past (Hillston and Duguid 2009, Ciocchetta et al. 2010).

3 BIO-PEPA: MODELING FORMALISM AND TOOL

In this section, we recall the Bio-PEPA modelling formalism and its corresponding modelling framework.

Bio-PEPA is a recently defined formal model description language for biochemical pathways based on a stochastic process algebra (Ciocchetta and Hillston 2009). In Bio-PEPA a *reagent-centric* style of modelling is adopted, and a variety of analysis techniques can be applied to a single model expression. In the reagent-centric style each reagent or species within the pathway is modelled as a distinct component which can undertake reactions, which are represented as actions in the process algebra. Currently supported analysis techniques include stochastic simulation at the molecular level, ordinary differential equations, probabilistic model checking and numerical analysis of a continuous time Markov chain.

Process algebras are a well-established modelling approach for representing concurrent systems facilitating both qualitative and quantitative analysis. Within the last decade they have also been proposed as the basis for several modelling techniques applied to biological problems, particularly intracellular signalling pathways, e.g. (Regev 2001, Calder, Gilmore, and Hillston 2006). A process algebra model captures the behaviour of a system as the actions and interactions between a number of entities, usually termed *processes* or *components*. In stochastic process algebras, such as PEPA (Hillston 2006) or the stochastic π-calculus (Priami 1995), a random variable representing average duration is associated with each action. In the stochastic π-calculus, interactions are strictly binary whereas in PEPA and Bio-PEPA the more general, multiway synchronisation is supported.

The main components of a Bio-PEPA system are the *species components*, describing the behaviour of each species, and the *model component*, describing the interactions between the various species. The species initial amounts are given in the model component. The syntax of the Bio-PEPA components is defined as:

$$S ::= (\alpha,\kappa)\ \mathtt{op}\ S \mid S+S \mid C \quad \text{with } \mathtt{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot \qquad\qquad P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(x)$$

where S is the *species component* and P is the *model component*. In the prefix term $(\alpha,\kappa)\ \mathtt{op}\ S$, κ is the *stoichiometry coefficient* of species S in reaction α, and the *prefix combinator* “op” represents the role of S in the reaction. Specifically, $\downarrow$ indicates a *reactant*, $\uparrow$ a *product*, $\oplus$ an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*. In the following we will use $\alpha\ \mathtt{op}\ S$ as an abbreviation for $(\alpha,1)\ \mathtt{op}\ S$. The operator “$+$” expresses the choice between possible actions, and the constant C is defined by an equation $C \overset{\mathit{def}}{=} S$. The process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes synchronisation between components P and Q, the set $\mathcal{L}$ determines those activities on which the operands are forced to synchronise, with $\underset{*}{\bowtie}$ denoting a synchronisation on all common action types. In the model component $S(x)$, the parameter $x \in \mathit{REAL}$ represents the initial amount of the species. Thus Bio-PEPA models are population models, which keep track of the impact of reactions on the number of molecules of the involved species.

In practical terms, process algebras resemble computer programs. They provide a textual description of the model using a domain-specific language, as opposed to a general-purpose programming language such as Java or C++. Process algebras might appear to be less user-friendly than graphical notations but – being similar to programming languages – they are well supported by editors used for programming, e.g., with cut and paste, find and replace, and a textual description has room for comments and descriptive names. They can be used seamlessly with a version control system such as CVS, as typically used to manage software projects. This means that it is very easy to check differences between model versions, which strongly supports biomodel engineering and makes it very convenient for researchers to collaborate on jamboree-style development of community-curated models.

The process algebraic specifications of the behaviours of species provide the central part of the Bio-PEPA model but some auxiliary information is also required. These include

- Information about the spatial organisation of the system in terms of compartments and their sizes. This allows correct translation to be made between the concentration view of species (as presented in ODEs) and the molecular counts used in stochastic simulation.
- Kinetic rate functions. It is assumed that each reaction is governed by a rate function which may depend on the current population of involved species. Typical examples include mass action kinetics and Michaelis Menten kinetics. These functions are specified separately.
- Parameters. Some kinetic rate functions require separate parameters such as Michaelis Menten constants. These are also defined separately.

This additional information completes a Bio-PEPA model to form a Bio-PEPA system suitable for analysis.

The Bio-PEPA Eclipse Plugin (Duguid et al. 2009) is an integrated development environment for Bio-PEPA which builds on the functionality of the Eclipse platform to support all aspects of Bio-PEPA modelling from management of projects and editing of models, through simulation and experimentation, to visualisation of results. Useful features inherited from Eclipse include built-in support for version control systems. Using the Eclipse Plug-in, Bio-PEPA models can be exported to SBML (Hucka et al. 2008) or to the PRISM modelling language (Kwiatkowska, Norman, and Parker 2009).

The Bio-PEPA Eclipse Plugin includes a suite of simulators, both continuous and discrete. Of the discrete simulators there are both exact simulators (implementing Gillespie's Direct Method (Gillespie 1977) and the Gibson-Bruck Next Reaction Method (Gibson and Bruck 2000)) and approximate simulators (implementing the τ-leap procedure (Gillespie 2001)). The simulation algorithm which we have modified to produce output for Traviando is an exact, discrete stochastic procedure, Gillespie's Direct Method. This has a range of important qualities for the present purpose including: 1) the algorithm simulates the reaction dynamics exactly, with every reaction represented without omission or approximation; 2) the algorithm operates on integer-valued molecular counts with transitions from one state to the next caused by the occurrence of a single reaction without averaging or amalgamation; and 3) the algorithm converges in the limit (as population levels tend to infinity) to the solution of the continuous-deterministic model of the system (Gillespie 2009), securing comparisons with this form of solution for Bio-PEPA models. In summary, the detailed single execution run provides us with a mean to embark on a trace analyzer for debugging purposes like Traviando to analyze the dynamic behavior of a model.

4 THE BIO-PEPA MODEL

For the purposes of this study our intention was to start from a Bio-PEPA model which as closely as possible recreated the model of Cho et al. (2003). Since in that paper the model was presented as a set of ordinary differential equations (ODEs) that was our starting point.

Each ODE is focused on one variable in the model, which in this case will be the concentration of a single species in the pathway, and the reactions which will increase or decrease the concentration. This is actually quite close to the Bio-PEPA view of the system. As explained above, in Bio-PEPA we model in a *reagent-centric* style, meaning that there is a species component for each biochemical species in the pathway and the species definition specifies which reactions increase or decrease that species. Thus the state of the system is similarly focused on the populations of the individual species in both the ODE model and the Bio-PEPA model. Therefore it was reasonably straightforward to generate a Bio-PEPA model from the published ODEs. However there is a subtle difference.

In the ODE model each reaction is represented anonymously by the mathematical expression detailing its impact on the variable. In contrast in the Bio-PEPA model each reaction is named and the quantitative impact of the reaction is specified separately in its declaration of the kinetic rate function. This separation of concerns allows the logical impact of the reaction to be viewed independently of the quantitative impact based on variable values. Moreover in the Bio-PEPA model a single reaction can have only one expression of its dynamics. Consequently in our development of the Bio-PEPA model from the ODE model we distinguished apparent reactions which impacted on the same variables, but with differing rates. In order to derive a species component from an ODE the first step was to associate a Bio-PEPA species component name with each variable in the ODEs ($M_1, M_2, \ldots, M_{31}$ in the model of Cho *et al.*). If a term in the following ODE was positive the corresponding component is taken to be a product of the reaction resulting in a $\uparrow$ prefix in the Bio-PEPA definition. Conversely each negative term is taken to be a reaction where the species is a reactant resulting in a $\downarrow$ term in the Bio-PEPA definition. A selection of the ODEs from the original model and the corresponding Bio-PEPA kinetic rate functions and species definitions are shown in Figure 2. Note that the ODEs for M_{30} and M_{31} contain different terms for reaction r_{29}. This forced us to specify two separate reactions r_{29} and $r_{29}alt$ in the matching Bio-PEPA definition in Figure 2. This issue will come up later again.

5 MODEL CONSISTENCY

A biochemical model can generally be viewed in one of two ways which we label the *reagent-centric* view and the *reaction-centric* view. The reagent-centric view defines the model in terms of the reagents; each reagent is defined with its relationship to each of the reactions. The reaction-centric view defines the model as a list of reactions specifying the effect each reaction has on the reagents of the system. Any one biochemical system can be defined in terms of either view, in our software the modeller must define the model in the reagent-centric view and the chemical equation view is generated automatically. We begin this section with a more detailed description of each view and then show how the combination assists the modeller in detecting modelling errors.

5.1 Reagent-centric point of view

This view is used in Bio-PEPA to specify a model in the format described in Section 3. There are four main sections of a Bio-PEPA model; the first defines all numerical constants used within the model. The second defines the rates of all the reactions in the model. These rates are usually dependent on the concentrations of the reactants of the reaction associated with the given rate. There is a set of predefined common functions such as mass action or Michaelis-Menten kinetics to assist in the definition of the reaction rates. The third section of the model defines all reagents (species) as a set of equations. Each equation defines one reagent and declares the relationship between the reagent and each reaction

$$
\begin{array}{rcl|rcl}
\frac{dM_1}{dt} & = & -k_1 \cdot M_1 \cdot M_2 + k_2 \cdot M_3 & M1 & \overset{def}{=} & r_1 {\downarrow} M1 + r_2 {\uparrow} M1 \\
\frac{dM_2}{dt} & = & -k_1 \cdot M_1 \cdot M_2 + k_2 \cdot M_{24} + k_{13} \cdot M_{12} & M2 & \overset{def}{=} & r_1 {\downarrow} M2 + r_2 {\uparrow} M2 + r_{24} {\uparrow} M2 + r_{13} {\uparrow} M2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{dM_{30}}{dt} & = & k_{29} \cdot M_{29} \cdot M_{31} - k_{30} \cdot M_{30} & M30 & \overset{def}{=} & r_{29} {\uparrow} M30 + r_{30} {\downarrow} M30 \\
\frac{dM_{31}}{dt} & = & -k_{29} \cdot M_{29} \cdot M_{30} + k_{30} \cdot M_{30} + k_{19} \cdot M_{17} & M31 & \overset{def}{=} & r_{29}alt {\downarrow} M31 + r_{30} {\uparrow} M31 + r_{19} {\uparrow} M31 \\
\hline
\end{array}
$$

$$
\begin{array}{llll}
r_1 = [k_1 \cdot M1 \cdot M2] & r_2 = [k_2 \cdot M3] & r_{13} = [k_{13} \cdot M12] & r_{19} = [k_{19} \cdot M17] \\
r_{24} = [k_{24} \cdot M24] & r_{29} = [k_{29} \cdot M29 \cdot M31] & r_{29}alt = [k_{29} \cdot M29 \cdot M30] & r_{30} = [k_{30} \cdot M30]
\end{array}
$$

Figure 2: Fragment of the published ODE model of the TNFα-mediated NF-κB signal transduction pathway (upper left), the corresponding fragment of the Bio-PEPA model (upper right) and its functional rate definitions (bottom).

in which it is involved. This is done using the operators defined in the preceding section to specify that the reagent is a product, reactant, activator, inhibitor or general modifier of the given reaction. The equation for a given reagent need make no mention of any reactions in which it is not involved. The fourth and final section of the Bio-PEPA model provides a system equation which composes all the reagents into a single system and specifies the reagents' initial quantities.

This view incorporates style recommendations known from software engineering, e.g. numerical constants are defined only once and a corresponding name is used throughout a model. It makes it straightforward to check which reactions are affecting a particular reagent. Special cases of reagents that act as sources or sinks are easily identified. This view is supported with the Bio-PEPA model editor, a text editor that also incorporates a rule set to provide immediate visual feedback for syntax violations and errors such as the use of undefined constants, reactions or reagents. Further details on this view can be found in (Duguid et al. 2009).

When we encoded the TNFα-mediated NF-κB signal transduction pathway model, the static analysis checks incorporated in the Bio-PEPA Eclipse Plug-in detects that the model has at least two areas of concern. We see from the species definitions in Figure 2 that reaction *r29alt* consumes species M_{31}, but from the kinetic laws we see that the rate of *r29alt* does not depend on the concentration of M_{31}, but on the concentration of M_{29} and M_{30} instead. This is highlighted by the software as a warning, meaning that the software considers the model to be suspicious but the modeller may override this concern and the model may still be compiled and analysed. We decided to fix this error by replacing *r29alt* with *r29* in the defining equation of M_{31} such that *r29alt* becomes obsolete and can be removed. The software also recognizes a similar error in the equation of M_5 where reaction *r20* is given a rate of $k_{20}M_5M_{20}$ instead of $k_{20}M_5M_{21}$ and we fixed this error accordingly.

5.2 Reaction-centric view

This view focuses on a representation of the specification of reactions in a model in the familiar format of chemical reactions. Since biologists are trained to read reaction definitions, they can easily spot errors in reagents involved in a reaction. Common errors that can be easily detected in this view are missing reagents, misspelled or incorrect reagents or stoichiometry values. Special cases like reactions that only produce or only consume reagents are easily recognized.

This view is not edited by the user at all, it is generated automatically from the above defined reagent-centric view. This means that the two views are always of the same model. In addition to listing all the reactions in the specified model, the software also highlights any sources or sinks in the model. Both reagents and reactions may be sources or sinks. For a reagent to be a source it must be a reactant in at least one reaction and must not be a product of any reaction. Conversely for a reagent to be a sink it must be the product of some reaction without being a reactant of any reaction. Both source and sink reagents are straightforward to detect from the reagent-centric view but non-trivial from the chemical equation view. The following reagent definition is that of a source reagent because it is consumed but never produced: $S \overset{def}{=} r_1 {\downarrow} S + r_2 {\downarrow} S$.

For a reaction to be a source reaction, it must have at least one product and no reactants. Analogously for a reaction to be a sink reaction it must have at least one reactant and no products. In contrast to source and sink reagents, source and sink reactions are straightforward to detect from the chemical equation view but non-trivial to see from the reagent-centric view. Both kinds are automatically computed and displayed to the modeller. Note that reagent source

$$\begin{array}{lrcl} r_1: & & \rightarrow & A \\ r_2: & A+B & \rightarrow & Z \\ r_3: & Z & \rightarrow & C+B \end{array}$$

$$\begin{array}{lcl lcl} A & \overset{def}{=} & r_1{\uparrow}A + r_2{\downarrow}A & B & \overset{def}{=} & r_2{\downarrow}B + r_3{\uparrow}B \\ Z & \overset{def}{=} & r_2{\uparrow}Z + r_3{\downarrow}Z & C & \overset{def}{=} & r_3{\uparrow}C \end{array}$$

Figure 3: A model which is not fully covered by an invariant as the total mass in the system is not conserved but which still has a local invariant: $B+Z$. The reaction r_1 is a source reaction and the reagent C is a sink reagent.

and sinks do not produce or consume mass whereas as reaction source and sinks do. The following is an example sink reaction because it consumes mass without producing any: $r_1 \overset{def}{=} P+S \longrightarrow$.

5.3 Conservation of matter view

For most models in systems biology some conservation rules apply. For instance, the total number of molecules of certain kinds remain constant throughout a simulation albeit these molecules may become part of a variety of more complex chemical compounds. This property can be formalized as a state invariant, a weighted sum of reagents whose total value depends on the initial quantities of reagents but otherwise remains constant throughout a simulation. Since a chemical master equation can be represented by a Petri net, we can readily apply invariant analysis techniques known from the theory of Petri nets. We can therefore compute a minimal generating set of invariants with a version of the Fourier-Motzkin method described in (Martinez and Silva 1982) for a simple class of Petri nets.

We implemented the method and provide a view on the computed invariants as well as a list of those reagents that are not covered by any invariant at all. The theoretical worst case complexity of the method is exponential although it is rarely seen in practice. Nevertheless we implemented the method with an upper bound on the internal matrix data structure to prevent this case from bringing down the overall framework and we also included a greedy heuristic that aims at minimizing internal data structures in an attempt to prevent the worst case from happening. We postprocess the computed invariants to reduce the generated set to a linearly independent set with a preference for invariants with minimal support. The rationale is that invariants that cover only a few reagents give more insight to a modeller and are easier to verify than large invariants that include a large number of reagents.

Note that a model does not need to fully conserve the mass for it to have reagent invariants. There may still be source and/or sink reactions which produce or consume mass such that the overall mass in the entire system is not invariant. A simple case of this is an enzyme enabled reaction such as that shown in Figure 3.

However, although it is possible to have localised invariants in the presence of a system which does not conserve mass, if all of the reagents in a model are covered by at least one invariant then the whole system must conserve mass. For this reason our software notifies the user of any reagents which are not covered by any invariant, the presence of which means that the whole model does not conserve mass. This notification allows the user to to decide whether this is behaviour which they expect from their model.

Our Bio-PEPA model yields the following invariants. The total amount of FADD is invariant, i.e. $M_{21}+M_{23}+M_{24}$ is constant. The total amount of Capase-8 is invariant, i.e., $M_{18}+M_{22}+M_{24}+M_{25}+M_{26}$ is constant. The Effector may be temporarily bound with IAP or permanently consumed as DNA fragmentation, i.e., $M_{26}+M_{27}+M_{28}+M_{29}+M_{30}$ is constant. The total amount of IKK is invariant, i.e., $M_8+M_9+M_{12}+M_{13}+M_{15}$ is constant. IkB/NF-kB (M_{14}) contributes to the production of NF-kB, i.e., $M_{14}+M_{15}+M_{16}$ is constant. In addition, $M_{14}+M_{15}+M_{17}+M_{30}+M_{31}$ is constant, which shows that IkN/NF-kB (M_{14}) contributes to the production of c-IAP (M_{31}) which is used in the regulation module to bind the Effector (M_{29}) and thus disable r_{25} to avoid DNA fragmentation.

There are six other invariants that contribute to cover all species which are not presented here in the interests of space. However, although all the reagents in the model are covered by some invariant, certain expected invariants such as the amount of TRAF2 (M_6) being preserved, are not derived.

In order to determine if an error in the model has caused the unexpected absence of the preservation of TRAF2 (M_6) we calculated the overall net effect of certain sequences of reactions in the pathway model. These allow us to deduce that, TNFα (M_1) can result in a DNA fragmentation (M_{28}) via the sequence of reactions $r1, r3, r20, r22, r24, r25, r27, r28$. However, this sequence also consumes Effector (M_{29}). Since Effector (M_{29}) cannot be produced by the model this implies that a sufficient initial amount of Effector is necessary to enable this sequence of reactions. We also observe that TRAF2 (M_6) and RIP1 (M_{10}) are increased as a side effect of r_{24}, which enables r_5 and r_7. Note that those two reactions compete with r_{20} for TNF./TNFR1/TRADD (M_5), such that this reaction sequence has a tendency to reduce its likelihood of occurring multiple times.

We also observe that the generation of RIP1n (M_{19}) and RIP1c (M_{20}) in the regulatory module is achieved via the sequence $r_1, r_3, r_{20}, r_{22}, r_{24}, r_{17}, r_{18}$. That sequence has the effect of producing one RIP1n and one RIP1c from one TNFα (M_1). However, it also generates one TRAF2 (M_6) as a side effect due to r_{24}.

Finally, the generation of IkB-P (M_{16}) is performed via the sequence $r_1, r_3, r_5, r_{11}, r_{13}, r_{14}, r_{16}, r_7, r_9$. This sequence consumes two times TNFα to generate IkB-P once due to r_5 and r_7. It also consumes IkB/NF-kB (M_{14}) compounds, which are not produced within the model and draw from a sufficient initial amount to enable reaction r_{14}. Furthermore, this sequence produces NF-kB (M_{17}) which will ultimately increase c-IAP (M_{31}) with r_{19} which is then captured in a reversible reaction of r_{29}, r_{30} but acts like a sink. The consumption of two TNFR1 (M_2) and two TRADD (M_4) due to r_{11} is only partially given back by r_{13}, when the compound breaks up and produces IKK* (M_{13}).

In light of these observations, one possible change to the model is to remove the creation of TRAF2 (M_6) and RIP1 (M_{10}) by r_{24} because this violates conservation of matter. It would also be necessary to increase the stoichiometry constants for r_{13} with respect to the production of TNFR1 (M_2) and TRADD (M_4) from one to two in order to preserve conservation of matter because two of each are previously consumed to generate TNF./TNFR1/TRADD/RIP1/TRAF2/IKK (M_{12}) that is input to r_{13}. (Other, more complex changes to the model might be better than these, but a domain expert should evaluate other possibilities once the modelling tools have highlighted the possibility of error in the model.)

If the above changes are applied then the model would respect invariants that are more intuitive. The total amount of TRAF2 is invariant, i.e., $M_6 + M_7 + M_9 + M_{12}$ is constant. The total amount of RIP1 is invariant and since r_{18} produces two species at once, we see two corresponding invariants, i.e., $M_{10} + M_{11} + M_{12} + M_{18} + M_{19}$ and $M_{10} + M_{11} + M_{12} + M_{18} + M_{20}$ is constant. The total amount of TRADD is invariant, i.e., $M_4 + M_5 + M_7 + M_9 + M_{11} + 2 \cdot M_{12} + M_{23} + M_{24}$ is constant. The weight of 2 on M_{12} for this invariant, indicates that r_{11} creates one M_{12} for inputs from M_9 and M_{11}. The total amount of TNFR1 is invariant, i.e., $M_2 + M_3 + M_5 + M_7 + M_9 + M_{11} + 2 \cdot M_{12} + M_{23} + M_{24}$ is constant. The final invariant covers TNFα (M_1) and most of the pathway model because it tracks what may result from the transformation of TNFα. It says that $M_1 + M_3 + M_5 + M_7 + M_9 + M_{11} + 2 \cdot M_{12} + 2 \cdot M_{13} + 2 \cdot M_{15} + 2 \cdot M_{16} + M_{18} + M_{19} + M_{23} + M_{24} + M_{25} + M_{26} + M_{27} + M_{28}$ is constant, which describes that TNFα (M_1) may result in IkB-P (M_{16}), RIP1n M_{19}, or DNA fragmentation M_{28}.

5.4 Cyclic behavior and reversible reactions

In a biological model, certain reactions may be reversible as well. In addition there may be longer sequences of reactions which are cyclic in the sense that performing exactly those reactions returns the model to the state it was in before the cycle. All these scenarios formally relate to an invariant in the sense that performing a set of reactions in particular quantities will yield the starting configuration again. As with the conservation of matter, this notion of invariant can be transferred from Petri net theory (Grafahrend-Belau et al. 2008). In Petri net theory a sequence of reactions with no effect on the state of the model is known as a t-invariant. A generating set can be computed with the Fourier-Motzkin method. The result is a weighted sum of reactions that if performed accordingly would result in the same starting state, i.e., it would describe a cycle in a simulation run of a stochastic discrete event simulation of the given model.

As with the reagent invariants, reaction invariants were implemented for Bio-PEPA. With the reagent invariants we present to the user the list of reagents not involved in any invariant as this means that the whole system does not conserve mass. We do the same for reaction invariants, reporting a list of reactions which are not involved in any reaction invariant.

The reaction invariant analysis reveals pairs of reversible reactions, which is obvious and straightforward to confirm yet helpful to increase our confidence in the encoding. The invariant analysis also shows that reactions r_{13}, r_{16}, r_{18}, r_{19}, r_{24}, r_{27}, and r_{28} are not contained within cyclic behavior. The effect of these reactions therefore cannot be reversed. The pathway is intended to generate certain output such as the DNA fragmentation (M_{28}) for some TNFα (M_1) as input. So a modeller can create a test harness of artificial reactions which directly feed back any output into its corresponding input. In other words the test reactions are the reverse of the intended overall effect of the pathway. If the original pathway has been modelled correctly then the addition of the test reactions will result in the model being covered by invariants. The total effect of the pathway and its test reactions should be to return the system to the state which it was in before traversing the pathway and the test reactions. Formulating this test harness requires a modeller to clarify their expectations on the overall effects of a sequence of reactions, an activity which may also reveal modelling or specification errors.

To our example model we added three test reactions, namely $r_{32} : M_{28} \rightarrow M_1 + M_{29}$, $r_{33} : M_{19} + M_{20} \rightarrow M_1 + M_{10}$, and $r_{34} : M_{16} + M_{31} \rightarrow 2 \cdot M_1 + M_{14}$, which describe the inverse effects of the intended overall effects of the pathway. With the addition of the reactions we re-ran the invariant analysis and obtained the expected result, i.e. the modified model is covered with invariants which describe the intended effect of the pathway and the test reactions as zero. Note that the test reactions r_{32}, r_{33}, and r_{34} are disabled for subsequent analysis. The Bio-PEPA software includes functionality which allows a modeller to knock-out (disable) specific reactions in simulation experiments.

5.5 A summary of steps towards a consistent model

The different views supported by Bio-PEPA can be used to proceed through the following list of steps to derive a model that is consistent at least from a qualitative point of view.

1) Encode a model in the reagent-centric view and resolve reported issues until a model reaches a state where all constants, rates, species, reactions and the overall system are specified and dependencies of reaction rates are checked.
2) Confirm chemical reactions in the reaction-centric view with respect to participating reactants and products as well as stoichiometry constants. Focus also on source/sink reactions and reactants. Return to the reagent-centric view and perform changes as necessary.
3) Compile a list of invariants that reflect conservation of mass of chemical compounds and check this list with reagent invariants calculated in the conservation of matter view. It is possible that invariants do not exactly match with expectations, but species that are expected to be part of some invariant should indeed be covered by the calculated ones. Also calculate the net total effect of key reaction sequences, most likely the ones that start at a source reaction/reagent and end at a sink reaction/reagent, to see if these sequences produce unexpected side effects on certain species. If this is the case, return to the reagent-centric view and perform changes as necessary.
4) Compile a list of reversible reactions and check those with the computed reaction invariants. For pathway models, calculate the overall effect of sequence of reactions that starts with some input reagent/reaction and goes to an expected corresponding output reagent/reaction. Add artificial reactions and reagents such that these perform the inverse of this effect and the overall model performs in a closed-loop. The resulting model should have corresponding reaction invariants. These artificial reactions confirm the correct understanding of the overall effect of a sequence of reactions.

The Bio-PEPA suite supports these steps, calculation of invariants included. For steps 3 and 4, Traviando provides information for invariants and for the effects of reaction sequences between source and sink reaction/reagents. To obtain these results, we produce a simulation trace where all reactions perform at least once with the Bio-PEPA suite and feed this trace into Traviando's report generator. After performing these steps, a model is sufficiently well-understood to investigate its dynamic properties and to consider quantitative aspects of its behavior.

6 FACE VALIDITY OF A DES SIMULATION

For a face validity check, we compare simulation results with known results either from the literature or with those expected by an expert in the field. Since Cho et al. (2003) focus on a sensitivity analysis, their published results are not sufficiently detailed to compare results for particular model configurations. We therefore limit our considerations to presentations of the simulated behavior for a version V1 of the model where errors in rate definitions have been fixed as discussed in Section 5.1 and a final version V2 that is consistent with a conservation of matter and incorporates all changes discussed in Section 5.

Figure 4 shows simulation results averaged over 1000 runs of Gillespie's algorithm up to time $t = 30$ with an initial configuration and rates as reported by Cho et al. (2003). We measure concentrations of species that serve as input, TNFα (M_1), and output, NF-κB (M_{17}), RIP1n (M_{19}), and DNA fragmentation (M_{28}). Both versions show equivalent results as can be seen in the graphs to the upper left and right of Figure 4 but for DNA fragmentation, which converges to a higher value in V2. There are of course concentrations of various species significantly affected by the changes we made. The lower left and right graphs in Figure 4 show simulation results for TNFR1 (M_2), TRADD (M_4), TRAF2 (M_6), and RIP1 (M_{10}). As expected from our changes, the amount of TNFR1 and TRADD increases due to the conservation of matter that is observed now. Concentrations of TRAF2 and RIP1 converge to lower values since they are not produced by r_{24} in V2 as it is the case in V1. This explains the higher values for DNA fragmentation in V2, because lower values of RIP1 reduce the inhibiting effect that r_{17} has on DNA fragmentation as it binds Capase-8* (M_{25}).

At this point, a serious model validation needs to take wet lab measurement data into account and requires a more substantial discussion on signal transduction pathways which is beyond the scope of this paper.

7 RELATED WORK

Post-hoc model-checkers complementary to this work are BioCham (Fages and Rizk 2007), BioNessie (Liu and Gilbert 2010) and MC^2 (Heiner, Gilbert, and Donaldson 2008), in that they can be applied to a continuous interpretation of the model (and in some cases to discrete-state simulations also). In contrast we are working exclusively with a discrete interpretation here. Because Bio-PEPA has both a continuous and a discrete-state interpretation it is possible that post-hoc model checkers such as these could be productively used alongside Traviando to check Bio-PEPA models.

A point of difference between the work reported here and the related work is that in approaches based on model-checking the user is required to invent a proposition to check against the model for every use of the model-checker. This proposition must then be correctly expressed in the available logic of the model-checker, such as PLTLc (Heiner, Gilbert, and Donaldson 2008). Although Traviando supports Linear Time Logic model-checking it is also possible to perform trace analysis and compute results such as invariants without encoding any property as a logical formula. This accessibility lowers the barrier to use by modellers who are familiar with computational tools but are not confident with sophisticated logics.

Through the link to Traviando it is now possible to visualise simulation traces which have been generated by the implementation of the Direct Method in the Bio-PEPA Eclipse Plugin. This approach to visualisation contrasts with

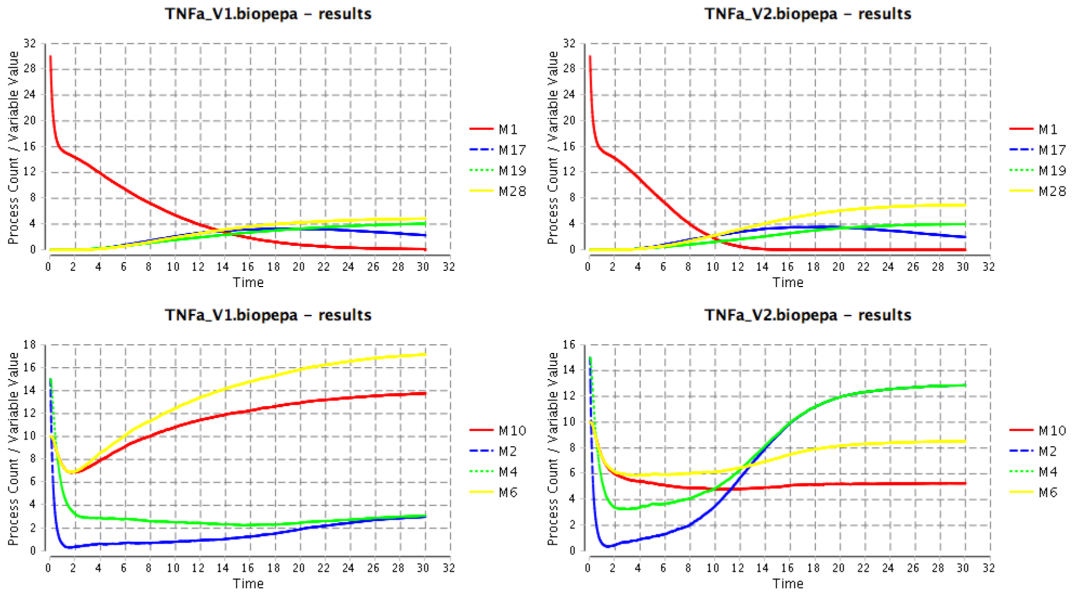

Figure 4: Simulation results for a Gillespie simulation of different model versions

animation-based approaches such as (Phillips 2010), where the progress of a chemical simulation is represented using 3D pictures, showing increase in the quantity of each chemical species using an increase in the volume assigned to a sphere associated with the species. One concern with animation-based approaches is whether they will scale with an increased number of chemical species. The example shown in (Phillips 2010) has three species. In contrast we have used Traviando successfully with the example from (Ciocchetta et al. 2008) which contains 108 species.

8 CONCLUSION

Making use of an existing body of knowledge in biological modeling is less trivial than one may expect. We illustrate several steps in the process of recovering a biological simulation model from the literature. These are steps necessary to familiarize oneself with a given model and to increase confidence in the internal consistency of a model. We chose an existing non-trivial pathway model and the Bio-PEPA Eclipse Plug-in tool suite to work with. The Bio-PEPA suite provides a variety of recently added automated techniques that support a modeler in the verification and testing of a biological model. These techniques provide different views to a model as well as the computation of reaction and reagent invariants which are helpful to identify certain types of errors ahead of any simulation. We found two types of inconsistency in the model from (Cho et al. 2003). One kind of inconsistency was between the ODEs and diagram showing their intention for the model. These may be regarded as programming/implementation errors. The others were inconsistencies apparent in both the ODEs and the diagram. These may be regarded as design errors. Our analysis techniques were able to uncover both kinds of error.

The Bio-PEPA suite can be downloaded from <www.biopepa.org/>. Traviando is available on request.

ACKNOWLEDGMENTS

Allan Clark, Stephen Gilmore and Jane Hillston are supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/E031439/1 “Stochastic Process Algebra for Biochemical Signalling Pathway Analysis”. Jane Hillston is supported by the EPSRC Advanced Research Fellowship and research grant EP/C543696/1 “Process Algebra Approaches to Collective Dynamics”. Stephen Gilmore and Jane Hillston are supported by the Centre for Systems Biology at Edinburgh. The Centre for Systems Biology at Edinburgh is a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1. Peter Kemper was supported by a Distinguished Visiting Fellow award from the Scottish Informatics and Computer Science Alliance.

REFERENCES

Breitling, R., R. A. Donaldson, D. R. Gilbert, and M. Heiner. 2010. Biomodel Engineering – From Structure to Behaviour. *Transactions on Computational Systems Biology XII* Springer-Verlag, LNBI 5945:1–12.

Calder, M., S. Gilmore, and J. Hillston. 2006. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Transactions on Computational Systems Biology*.

Cho, K., S.-Y. Shin, W. Kolch, and O. Wolkenhauer. 2003, December. Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: A case study for the TNFα-mediated NF-κB signal transduction pathway. *Simulation* 79 (12): 726–739.

Ciocchetta, F., A. Degasperi, J. K. Heath, and J. Hillston. 2010. Modelling and analysis of the NF-κB pathway in Bio-PEPA. In *Transactions on Computational Systems Biology XII*, Volume 5945, 229–262: Springer.

Ciocchetta, F., and J. Hillston. 2009, August. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science* Concurrent Systems Biology: To Nadia Busi (1968–2007), 410 (33–34): 3065–3084.

Ciocchetta, F., J. Hillston, M. Kos, and D. Tollervey. 2008. Modelling co-transcriptional cleavage in the synthesis of yeast pre-rRNA. *Theoretical Computer Science* 408 (1): 41–54.

Duguid, A., S. Gilmore, M. Guerriero, J. Hillston, and L. Loewe. 2009. Design and development of the software tools for Bio-PEPA. In *Proceedings of the Winter Simulation Conference*, 956–967. Austin, Texas: IEEE Press.

Fages, F., and A. Rizk. 2007. On the analysis of numerical data time series in temporal logic. In *Computational Methods in Systems Biology '07*, Volume 4695 of *Lecture Notes in Computer Science*, 48–63: Springer.

Gibson, M. A., and J. Bruck. 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A* 104:1876–1889.

Gillespie, D. T. 1977. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* 81 (25): 2340–2361.

Gillespie, D. T. 2001. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.* 115 (4): 1716–1733.

Gillespie, D. T. 2009. Deterministic limit of stochastic chemical kinetics. *J. Phys. Chem. B* 113:1640–1644.

Grafahrend-Belau, E., F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch. 2008. Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics* 9 (90).

Heiner, M., D. Gilbert, and R. Donaldson. 2008. Petri nets in systems and synthetic biology. In *SFM'08:Bio*, 215–264: Springer. LNCS 5016.

Hillston, J. 2006. *A compositional approach to performance modelling*. Cambridge University Press.

Hillston, J., and A. Duguid. 2009. Deriving differential equations from process algebra models in reagent-centric style. In *Algorithmic Bioprocesses*, 487–504: Springer.

Hucka, M., S. Hoops, S. Keating, N. Le Novère, S. Sahle, and D. Wilkinson. 2008. Systems Biology Markup Language (SBML) Level 2: Structures and facilities for model definitions. *Available from Nature Precedings*. ⟨http://dx.doi.org/10.1038/npre.2008.2715.1⟩.

Kemper, P., and C. Tepper. 2009. Automated trace analysis of discrete-event system models. *IEEE Transactions on Software Engineering* 35 (2): 195–208.

Kwiatkowska, M., G. Norman, and D. Parker. 2009. PRISM: Probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review* 36 (4): 40–45.

Liu, X., and D. Gilbert. 2010. BioNessie: A biochemical networks simulation environment. *Journal of Cell Research*. To appear.

Martinez, J., and M. Silva. 1982. A simple and fast algorithm to obtain all invariants of a generalized Petri net. In *Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets*, 301–310. London, UK: Springer-Verlag.

Phillips, A. 2010. *Symbolic systems biology: Theory and methods*, Chapter A Visual Process Calculus for Biology. Jones and Bartlett. In Press.

Priami, C. 1995. Stochastic π-calculus. *The Computer Journal* 38 (6): 578–589.

Regev, A. 2001. Representation and simulation of molecular pathways in the stochastic π-calculus. In *Proceedings of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks*.

AUTHOR BIOGRAPHIES

ALLAN CLARK is a post doctorate researcher at the University of Edinburgh. His research interests include algorithms and implementations for performance modelling via process algebras as well as solutions to Markov chains and techniques for stochastic and deterministic modelling of very large scale systems. Ongoing work into the query specification for performance models written in process algebras has led to the development of the eXtended Stochastic Probes specification language.

STEPHEN GILMORE is a Reader in Computer Science and a member of the Laboratory for the Foundations of Computer Science at The University of Edinburgh. His research interests are centred on the development of quantitative and computational tools for modelling discrete dynamical systems. These have included modelling tools for predicting performance and quality-of-service measures for software systems, and simulators and analysers for biological modelling applications. His web page can be found via `<homepages.inf.ed.ac.uk/stg>` and his email address is

<stg@inf.ed.ac.uk>.

JANE HILLSTON FRSE holds a personal chair in quantitative modelling at The University of Edinburgh. She is inventor of the PEPA modelling language, which has been applied to the performance analysis of systems ranging from protocols in wireless networks to circadian rhythms in biological cells. Her research contribution has been recognised by both a Distinguished Dissertation and the BCS/Microsoft Roger Needham award. In 2005 she was awarded a five-year Advanced Research Fellowship from the EPSRC to support her research on quantified methods and process algebras. Her web page can be found via <homepages.inf.ed.ac.uk/jeh> and her email address is <jeh@inf.ed.ac.uk>.

PETER KEMPER is an Associate Professor in the Department of Computer Science at the College of William and Mary (previously TU Dortmund and TU Dresden, Germany). His research interests include modeling techniques and tools for performance, performability and dependability analysis of systems. His group develops the Traviando trace analyzer. His web page can be found via <www.cs.wm.edu/∼kemper> and his email address is <kemper@cs.wm.edu>.