How Might Petri Nets Enhance Your Systems Biology Toolkit

Monika Heiner[1], David Gilbert

School of Information Systems, Computing and Mathematics
Brunel University, Uxbridge, Middlesex UB8 3PH, UK,
{monika.heiner|david.gilbert}@brunel.ac.uk
[1] *on sabbatical leave from Brandenburg University of Technology*

Abstract. "How might Petri nets enhance my Systems Biology toolkit?" – this is one of the questions that we get on a regular basis, which motivated us to write an answer in the form of this paper.
We discuss the extent to which the Petri net approach can be used as an umbrella formalism to support the process of BioModel Engineering. This includes the facilitation of an active and productive interaction between biomodellers and bioscientists during the construction and analysis of dynamic models of biological systems. These models play a crucial role in both Systems Biology, where they can be explanatory and predictive, and synthetic biology, where they are effectively design templates. In this paper we give an overview of the tools and techniques which have been shown to be useful so far, and describe some of the current open challenges.

Key words: BioModel Engineering; Systems Biology; synthetic biology biomolecular networks; qualitative, stochastic and continuous Petri nets; model checking.

1 Motivation

Biology is increasingly becoming an informational science. This revolution has been driven by technological advances which have supported the development of studies at many levels of intra- and intercellular activity. These advances have facilitated the analysis of how the components of a biological system interact functionally - namely the field of Systems Biology [39]. In general this analysis is quantitative and over time [1], which can basically be done in a stochastic or continuous fashion.

At the heart of this field lies the construction of models of biological systems, see Figure 1. These models are used for analysis which should ideally be both *explanatory* of biological mechanisms and *predictive* of the behaviour of the system when it is perturbed by, e.g., mutations, chemical interventions or changes in the environment. Furthermore, models can be used to help make genetic engineering easier and more reliable, serving as design templates for novel synthetic biological systems – an emerging discipline known as synthetic biology [19,26]. Central to both Systems and Synthetic Biology is BioModel Engineering (BME) which is the science of designing, constructing and analyzing computational models of biological systems [8].

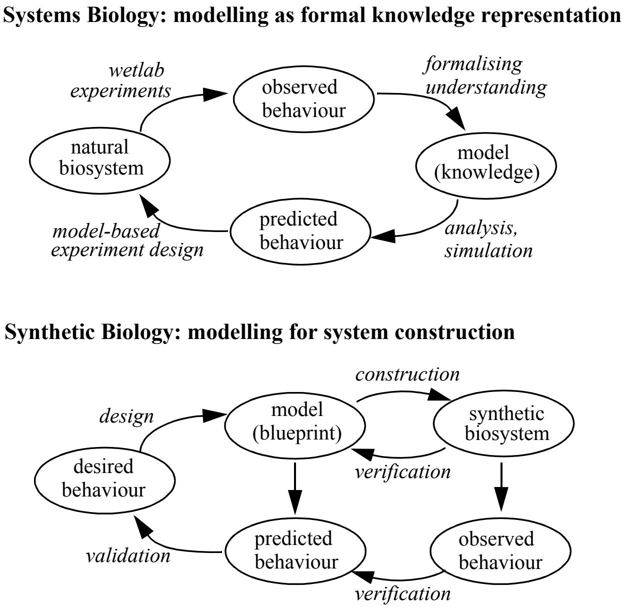

Fig. 1. The role of formal models in Systems and Synthetic Biology.

In this paper we discuss the Petri net approach to BioModel Engineering. We show how Petri nets are highly suited to the support of both Systems and Synthetic Biology. Not only are they immediately attractive to life scientists in terms of their graphical representation, thus facilitating communication with the modeller, but they permit qualitative as well as quantitative (stochastic, continuous, hybrid) descriptions by a family of related models and come with a very powerful analytical toolkit [29].

The structure of this paper is as follows: in Section 2 we discuss representational issues, and in Section 3 we recall our overall conceptual framework, followed by an overview of Petri net contributions in Section 4. We discuss model checking and its use in BioModel Engineering in Section 5, and summarize our toolkit in Section 6. Then we discuss the outlook and challenges for the use of Petri nets in this area in Section 7, and give an overall summary in Section 8.

This paper is deliberately kept informal. We assume a basic understanding of Petri nets; a gentle introduction within the given context of BioModel Engineering can be found in [28, 48], for related formal definitions see [29].

2 Representation style – just a matter of taste?

We consider biochemical *reaction networks* (RN), consisting of a finite set of

- *reactions* (biochemical reactions, complexation/decomplexation, phosphorylation/dephosphorylation, conformational change, transport steps, . . . , any type of biochemical interactions, which are considered to be atomic on the chosen abstraction level), converting or transporting
- *species* (biochemical compounds, proteins, protein conformations, complexes, . . . , any species, which are considered to be atomic on the chosen abstraction level).

Reactions may carry elementary kinetic information, *the reaction rates*, if known. A typical pattern of elementary reaction rates is the *mass/action kinetics*, where the state-dependent rate function v_i of the reaction r_i is assumed to be given by the product of the current values of the species involved and some further constants k_i (usually called reaction parameters), summing up environmental conditions such as temperature, pressure, . . . ; for examples see Figure 2, left and Figure 3, right.

Thus, the notion of a reaction network covers both the qualitative and quantitative modelling paradigms. A reaction network can be specified as:

(1) *list* (or set) of all individual (stoichiometric) reactions, in a reaction-centric or species-centric style,

or a graph describing all individual reactions, which could be either

(2) *hypergraph*, i.e., a graph where an arc can connect any number of nodes,
(3) *bipartite graph*, i.e., a graph with two types of nodes, whereby arcs never connect nodes of the same type; e.g. a Petri net.

Remark: In the Petri net examples we adopt the usual interpretation and represent reactions (the active system components) by transitions, and species (the passive system components) by places.

These three different styles of representations (1)–(3) can be converted into each other without loss of information. There are two well-known representations, which can be uniquely derived from any of (1)–(3), but generally not vice versa:

(4) *incidence matrix* (in Systems and Synthetic Biology better known as stoichiometric matrix),
(5) *system of ordinary differential equations* (ODEs).

Both derived styles of representation lose structural information if there are catalysts involved or species, which are both the subtrate as well as the product of one and the same reaction. It is an easy exercise to imagine two reaction networks generating the same incidence matrix or ODEs, but differing in their discrete behaviour. We start from the introductory example in [61], see Fig. 2, left. The net on the right uses a modifier arc (dotted line), which is a standard

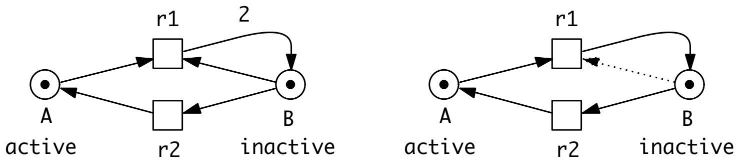

Fig. 2. Two reaction networks generating the same incidence matrix and the same ODEs. With the kinetic rates $v_1 = v(r_1) = k_1 \cdot A \cdot B, v_2 = v(r_2) = k_2 \cdot B$, we get the two equations $dA/dt = v_2 - v_1$, $dB/dt = v_1 - v_2$.

feature of the Systems Biology Markup Language (SBML) [41] to indicate that the rate of r_1 dependents on B. For more examples see [63].

Consequently, the structural interpretation of ODEs models, i.e. converting (5) into any of (1)–(4), calls for special attention. The problematic cases are reactions with complex kinetics. Some tools popular in Systems Biology, e.g., COPASI [40], provide predefined functions representing whole building blocks, i.e. subnets in our reaction networks. See Fig. 3 for an example, which has been taken from the reaction network reported in [9]. The network on the left has the structure as suggested by the schematic representation in [9] and the list of reactions in the model's SBML format created by COPASI. The network on the right shows the correct structure, which is hidden in the kinetics of reactions 23 and 25.

Hence, the (backward) translation from ODEs to structured models suitable for qualitative or stochastic analysis needs to be performed with great care. In [63], we look into the structure inference problem for ODEs models. We provide three biochemically relevant sufficient conditions under which the derived structure is unique and counterexamples showing the necessity of each of the following conditions.

- All reactions use pure mass/action kinetics, and the reaction parameters belong to a finite alphabet of symbols.
- The reaction network does not contain any void reaction.
- The same parameter is never used for two different reactions with the same reactants.

In summary, neither (4) nor (5) are suitable core notations for reaction networks, which may be subject to various complementary qualitative, stochastic, or continuous analysis techniques.

Structure versus list notation. Enumerating all reactions in a list (in fact an unordered set) is perfectly fine, if the notation is to be read by a computer. A structure-oriented graphical notation brings the additional advantage for an human reader of revealing the causality relation (structure, topology) among the reaction set. As such we consider (2) and (3) to be closer to the reality to be

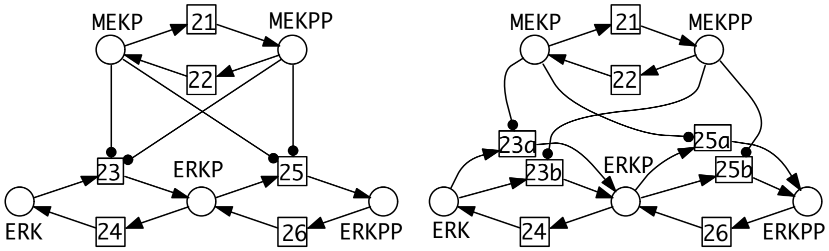

Fig. 3. Freestyle kinetics may hide essential structure. Assuming for reaction r_{23} in the network on the left the complex kinetics $v_{23} = v(r_{23}) = k_{23a} \cdot ERK \cdot MEKP + k_{23b} \cdot ERK \cdot MEKPP$, likewise for r_{25}, yields the network on right with all reactions having mass/action kinetics, e.g., $v_{23a} = v(r_{23a}) = k_{23a} \cdot ERK \cdot MEKP$.

modelled.

Hypergraph versus Petri net. At first glance, both are equally suitable for a graphical representation of the network structure. The difference in the graphical notation style seems to be a matter of taste.

Petri nets enjoy a formal semantics, which could also be formulated in the terminology of hypergraphs. However, this might become a bit over-complicated. Reasoning about the reaction network structure, e.g. to derive behavioural properties, requires splitting a hyperarc into its ingredients. We need to refer to the reaction itself, its reactants – the nodes where the hyperarc starts, and its products – the nodes where the hyperarc goes to, and the individual multiplicities of those parts of the hyperarc in the case of stoichiometric information.

Moreover networks tend to be very large. This motivates the reuse of a well-established engineering principle to manage the design of large-scale systems – hierarchical decomposition. Formal concepts of hierarchical Petri nets with building blocks are have been well elaborated, see, e.g., [20]. They facilitate the modeling of large real-world systems because the comprehension of the whole network builds upon the understanding of all building blocks and the interconnection of their interfaces. Hierarchical structuring changes the style of representation, but does not change the actual net structure of the underlying reaction network.

Even if hierarchical modelling approaches have gained little popularity in practice so far in Systems Biology, they are indispensable for large-scale networks. There is nothing like this for hypergraphs.

This explains why we keep to Petri net terminology when in the next sections we briefly describe some contributions which Petri net theory has to offer for reaction network analysis, and thus to a Systems Biology toolkit.

Disclaimer: Being restricted to 20 pages, this paper can not exhaustively cover the field.

3 The Framework

In the following we recall our overall framework, introduced in [24], that relates the three major ways of modelling and analysing biochemical networks: qualitative, stochastic and continuous, compare Figure 4.

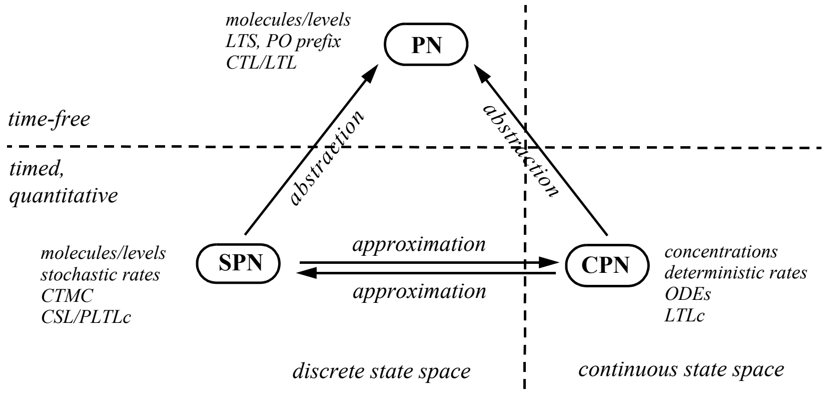

Fig. 4. Conceptual core framework.

Qualitative paradigm. The most abstract representation of a biochemical network is *qualitative* and is minimally described by its topology, usually as a bipartite directed graph with nodes representing biochemical entities and reactions, or in Petri net terminology *places* and *transitions*. Arcs can be annotated with stoichiometric information, whereby the default stoichiometric value of 1 is usually omitted.

The qualitative description can be further enhanced by the abstract representation of discrete quantities of species, achieved in Petri nets by the use of tokens at places. These can represent the number of molecules, or the level of concentration, of a species, and a particular arrangement of tokens over a network is called a *marking*, specifying the system state.

The behaviour of such a net forms a discrete state space, which can either be captured as (1) a Labeled Transition System (LTS) (in the Petri net community better known as reachability or marking graph) to describe the net behaviour by all (totally ordered) interleaving sequences in the style of transition-labelled automata (interleaving semantics), or as (2) a finite prefix of a maximal branching process (PO prefix) to describe the net behaviour by all partially ordered transition sequences (partial order semantics). Animating a Petri net by sequentially firing individual transitions generates a path through the LTS.

Both descriptions of behaviour can be analysed for the purpose of model verification. In the bounded case, this is best done using model checking techniques, where the properties of interest are expressed by, e.g., a branching time temporal logic, one instance of which is Computational Tree Logic (CTL) [15], or a linear-time logic (LTL) [54].

The standard semantics for these qualitative Petri Nets (PN)[1] do not associate a time with transitions or the sojourn of tokens at places, and thus these descriptions are time-free. The qualitative analysis considers however all possible behaviour of the system under any timing. Thus, the qualitative Petri net model itself implicitly contains all possible time-dependent behaviour.

Timed information can be added to the qualitative description in two ways – stochastic and continuous.

Stochastic paradigm. The stochastic Petri net (SPN) description preserves the discrete state, but in addition associates an exponentially distributed firing rate (waiting time) with each reaction. The firing rates are typically state-dependent and specified by rate functions. All reactions, which occur in the PN, can still occur in the SPN, but their likelihood depends on the probability distribution of the associated firing rates. The underlying semantics is a Continuous-Time Markov Chain (CTMC). Stochastic simulation generates a random walk through the CTMC.

Special behavioural properties can be expressed using, e.g., Continuous Stochastic Logic (CSL), a stochastic counterpart of CTL which was originally introduced in [4], and extended in [5], or PLTLc, a probabilistic extension of LTL with constraints [18].

The PN is an abstraction of the SPN, sharing the same state space and transition relation with the stochastic model (if there are no parallel transitions), with the probabilistic information removed. All qualitative properties valid in the PN are also valid in the SPN, and vice versa.

Continuous paradigm. The Continuous Petri Net (CPN) replaces the discrete values of species with continuous values, and hence is not able to describe the behaviour of species at the level of individual molecules, but only the overall behaviour via concentrations. We can regard the discrete description of concentration levels as abstracting over the continuous description of concentrations. Timed information is introduced by the association of a particular deterministic firing rate with each transition, permitting the continuous model to be represented as a set of Ordinary Differential Equations (ODEs). The concentration of a particular species in such a model will have the same value at each point of time for repeated computational experiments. The state space of such models is continuous and linear. It can be analysed by, for example, Linear Temporal Logic with constraints (LTLc) in the manner of [11].

Moving between stochastic and continuous paradigms. One and the same quantitative model can be read either stochastically or continuously, no changes being required (up to some scaling in the rate functions for higher order reactions). In the stochastic case, the rate functions define the state-dependent rates of the individual occurrences of transition firings, while in the continuous case the rate functions define the strength of the state-dependent continuous

[1] As we want to use the term *Petri net* as umbrella term, we refer to the standard Petri net class as *qualitative Petri nets*, if required for the sake of unambiguity.

flow. In [24] we discuss in more detail how the stochastic and continuous models are mutually related by approximation. The CPN replaces the probabilistically distributed reaction firing in the SPN by a particular average firing rate defining the continuous token flow. In turn, the stochastic model can be derived from the continuous model by approximation, reading the tokens as concentration levels, as introduced in [10].

Bridging to the continuous paradigm. It is well-known that time assumptions generally impose constraints on behaviour. The qualitative and stochastic models consider all possible behaviour under any timing, whereas the continuous model is constrained by its inherent determinism to consider a subset. This may be too restrictive when modelling biochemical systems, which by their very nature exhibit variability in their behaviour.

The move from the discrete to the continuous paradigm may come along with counter-intuitive effects; e.g., a trap – a set of places which can never become empty in the discrete case as soon as it has obtained a token – can be emptied in the continuous case [62]. See [2] for another surprising example.

For illustration we briefly discuss the toy example given in Figure 2, left, which has been used in [61] to introduce the idea of Absolute Concentration Robustness (ACR). A place of a CPN (a variable of an ODEs model) is said to have ACR if its concentration is the same in all positive steady states; i.e., it does not depend on the total mass in the system, but only depends on the kinetic constants. The place A is such an ACR place; its steady state concentration is k_2/k_1, while B has the steady state concentration $total - k_2/k_1$, with $total$ being the conserved total mass in the network.

The other place B forms a bad siphon, i.e., a siphon, not containing a trap. It is easy to see that the net does not have a live initial marking in the discrete world. Thus, there is always a dead state reachable, in which the bad siphon, i.e., B, is empty and all tokens reside in A, when the system is considered stochastically. However, there is no sign of the dead state when reading the net as CPN. Moreover, increasing the mass in the closed system will increase the steady value of B, as A has ACR. Thus, the continuous model may predict a simplified behaviour that does not reflect the variety, which is possible in the discrete case, e.g. in the given example, that the siphon will eventually get empty.

When analysing ODEs, one actually considers a family of related models, varying in some structural details, but generating the same ODEs. So far, the bridge to the continuous world is not equally well understood regarding the stochastic world. However, qualitative, stochastic & continuous Petri nets, which share structure should generally share some behavioural properties as well.

Applications. Case studies demonstrating the move between the three paradigms can be found in [29] (signalling cascade), [25] (biosensor gene regulation), and [28] (signal transduction network). In [23], we demonstrate by a case study the relationship between the strongly connected state space of the qualitative Petri net and the steady state behaviour of the corresponding continuous Petri net.

3.1 Beyond the core framework

In addition to the base case techniques, suitably expressive modelling formalisms and very efficient computational techniques are required to support abstraction levels and to give a holistic perspective on inherently multi-scale systems. As a consequence, a variety of different techniques have been proposed during the last decade, and new ones are constantly emerging. At the same time, many well-established and sophisticated modelling techniques from Computer Science, equipped with advanced analytical methods, have been applied to the exciting and challenging research field of Systems and Synthetic Biology.

Figure 5 shows the relationship between the most important modelling techniques contributed by the Petri net community. In the following we briefly characterize the main players, going beyond the core framework.

Extended Petri Nets (XPN) complement the standard ingredients of Petri nets by some dedicated features for the sake of easy modelling, but often sacrificing static analyzability. They exist in numerous variations. An extension particularly useful for Systems and Synthetic Biology are special arcs, e.g., read (test) arcs and inhibitor arcs, which provide adequate abstract modelling means for activating or inhibiting activities; see, e.g., [12, 35].

Functional Petri Nets (FPN) [38] pick up the idea of self-modifying nets [64] and use state-dependent functions to dynamically adjust arc multiplicities, which has been used to simulate metabolic pathways [14].

Time Petri Nets (TPN) equip transitions with a deterministic firing delay, typically characterized by a continuous time interval [49] of minimal and maximal delay. Remarkably, this net class can be fully analysed using a discretized state space only [56]; the particular capabilities for Systems and Synthetic Biology have be demonstrated in [55, 57, 58].

Autonomous Continuous Petri Nets (ACPN) [16] are time-free, but have continuous state space. Their analysis requires the consideration of all possible behaviours for any timing constraint. They are particularly helpful in exploring the relation between discrete and continuous models.

Generalised Stochastic Petri Nets (GSPN) extend SPN by immediate transitions [45], possibly complemented by special arcs. They have been used in, e.g., [42].

Extended Stochastic Petri Nets (XSPN) enrich GSPN with transitions having deterministic firing delays, typically characterized by a constant. They come in two flavours: relatively timed or absolutely timed, which helps in model-based wet-lab experiment design; see [32].

Hybrid Petri Nets (HPN) enrich CPN with discrete places and discrete transitions having deterministic firing delay, see, e.g., [17].

Hybrid Functional Petri Nets (HFPN) are a popular extension of HPN combining them with the self-modifying arcs of FPN. This net class is supported by the licensed tool Cell Illustrator, which has been deployed in many case studies [50].

Generalised Hybrid Petri Nets (GHPN) combine all features of XSPN and CPN [37].

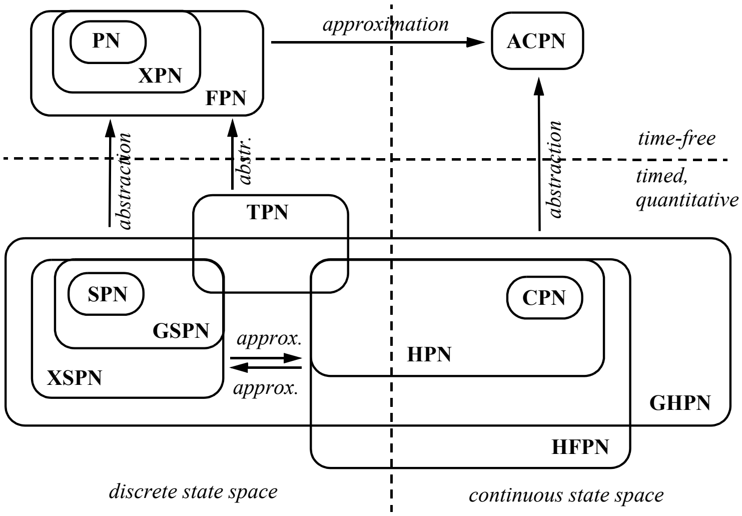

Fig. 5. Extended Framework showing the relation between some of the Petri net classes used in Systems and Synthetic Biology.

In addition, further novel Petri net classes have been developed, which may turn out to be useful in Systems and Synthetic Biology. We briefly discuss two of them.

The *Probability Propagation Nets* [43] provide a Petri net representation for the propagation of probabilities and likelihoods in Bayesian networks. As such they enhance the transparency of propagation processes in the Bayesian world by exploiting structural and dynamic properties of Petri nets. They can be used to express knowledge which has been statistically extracted from gene expression array data (in the wet lab) to obtain executable and analyzable models of the data source (in the dry lab).

The *Error-correcting Petri Nets* [51] allow for the algebraic detection and modulo-p correction of non-reachable Petri net states, without ever constructing the state space. They may gain importance in synthetic biology, e.g., to design biological systems with monitoring capabilities. The observation of states which are supposed to be non-reachable in the designed system will indicate some behavioural deviations from the original design. The method is able to pinpoint the source of the problem to a single place (several places) for a single error (multiple errors) depending on the amount of structural redundancy utlised during model construction.

This variety clearly demonstrates one of the big advantages of using Petri nets as a kind of umbrella formalism – the models may share the network structure, but vary in their quantitative (kinetic) information.

Related material. Formal definitions covering discrete, stochastic, continuous and hybrid variants of the Petri net formalisms, illustrated by many technical examples, can be found in [16]. The net classes closest to the continuous setting of biomolecular networks are discussed in Chapter 7, where the speeds (i.e., the rates) may depend on the C-marking (i.e., the current values of the continuous places). Combining the concept of variable rates with Definition 5.4 (feeding and draining speed of a place) and Definition 5.5 (the balance of the marking of a continuous place) explains how the ODEs model is set up.

However, the modelling complexity on hand calls for formal definitions and related tool support tailored to the specific needs of Systems and Synthetic Biology; e.g., by restricting the domain of the state-dependent rate functions to the pre-places of a transition in order to maintain a close relation between structure and behaviour. Biochemically interpreted Continuous Petri Nets have been introduced in [23], biochemically interpreted Stochastic Petri Nets in [24], and biochemically interpreted Extended Stochastic Petri Nets in [32]. More formal definitions covering the core framework can be found in [29].

4 Overview of Petri net specific contributions

Probably the most popular notions in Petri net theory having their counterparts in Systems Biology are place and transition invariants (P/T-invariants for short). They are defined on a model's representation, which is called incidence matrix in the Petri net community and stoichiometric matrix in Systems Biology.

T-invariants can be seen as the integer counterparts of different representations of the continuous members of a cone, known in Systems Biology as elementary flux modes, extreme pathways, and generic pathways, while P-invariants correspond to conserved moieties [66].

Beyond that, Petri nets offer the following *additional features* to the list of established techniques known for quite a while in Systems Biology, such as nonlinear ODEs analysis, bifurcation theory, chemical reaction network theory [21], stoichiometric and flux balance analysis [52], etc..

4.1 Readability

Petri nets are an intuitive representation of reaction networks, which makes them easily comprehensible. They uniquely define derived notations, such as stoichiometric matrix or ODEs, but not vice versa; for details see Section 2.

Applications. Petri nets allow the unambiguous, but still readable representation of various types of biological processes at different levels of abstraction, with different resolution of detail, in one and the same model, ranging from the conformational change of a single molecule to the macroscopic response of a cell, the development of a tissue, or even the behaviour of a whole organism [48]. A recent survey [6] has shown how Petri nets can be applied to transcriptional, signalling, and metabolic networks, or combinations of them, illustrating this with a rich set of case studies. Most of these focus on the molecular level; however examples at the multi-cellular level include the signal-response behaviour of an organism [47], and developmental processes in multi-cellular pattern formation [7, 13].

4.2 Executability

Petri nets are directly executable with a suitable tool like Snoopy [46, 59]. The visualization of the token flow

- allows the user to better comprehend the net behaviour, and
- facilitates the communication between wet-lab experimentalists and dry-lab (computational) theoreticians.

Applications. The execution allows a time-free animation in the case of qualitative Petri nets (these are the standard Petri nets), and time-dependent animation and simulation in the case of quantitative Petri nets (stochastic, continuous, and hybrid Petri nets).

4.3 Causality

The structure of Petri nets reflects the causality relation among the reactions building the whole network, while clearly distinguishing between alternative and concurrent behaviour; see [31] for a more detailed discussion. This permits reasoning about "which reaction has to happen first, before another reaction can happen afterwards "in order to transport some mass (metabolic networks) or transfer some information (signal transduction network) through the network.

Applications. Reasoning over causality helps, e.g., in evaluating T-invariants (likewise elementary flux modes, extreme pathways, generic pathways), which define subnets in a reaction network. The unfolding of the involved reactions reveals their partial order causality, and may allow for complementary insights into the net behaviour; for examples see [29], [28].

4.4 Efficiency gain

The Petri net structure allows for efficiency gains of some analysis algorithms, which are otherwise not possible. The crucial point is that the occurrence of a reaction only causes changes in its local environment, which consists of its pre-places, the reactants, and its post-places, the products. Vice versa, this requires that all dependencies are reflected in the structure. Thus we restrict the domain of transitions' rate functions to the transitions' pre-places.

Applications. Examples implemented in our toolkit, see Section 6, include:

- *Stochastic simulation* – to compute/update the state-dependent transition rates [34];
- *Static ordering of variables (places)* – to statically determine suitable orders of variables for high compression effects in symbolic representations, e.g., for symbolic model checking. In [60], we demonstrate how the performance of the popular CSL model checker PRISM [53] could take advantage of our structure-based heuristics to derive suitable orders of variables.
- *Static ordering of transitions* – to statically determine suitable orders of transitions for efficient symbolic state space construction, e.g., saturation-based construction, with the aim to avoid intermediate data structures which are much larger than the final ones.

4.5 Static analysis

The standard analysis techniques, which are commonly used in many Computer Science approaches, need to construct the partial or full state space (expressed as LTS or PO prefix). In addition to these dynamic analysis techniques, Petri net theory offers a variety of static analysis techniques, which permit the decisions of general behavioural properties, such as boundedness and liveness, without state space construction.

- *Boundedness* ensures upper bounds for all species, which implies a finite state space. Structural boundedness guarantees this behaviour for any initial marking.
- *Liveness* ensures that all reactions will forever contribute to the system behaviour, which precludes dead states. Structural liveness guarantees that there is a live initial marking.

Static analysis techniques, which we found useful so far in analyzing biochemical reaction networks, comprise:

- *property-preserving reduction rules*, which replace subnetworks by generally smaller ones without changing the general behavioural properties;
- *structural analyses*, which determine, e.g., the net structure classes (such as state machine, synchronisation graph, free choice nets, extended simple nets), Siphon Trap Property (STP), or rank theorem, and apply related theorems;
- *linear programming techniques*, which perform, e.g., structural boundedness check to decide if the state space is finite; P/T-invariant computation and coverage test, which supports also model validation; or state/trap equation check as sufficient criteria for non-reachability.

More details can be found in [29], where we recall basic notions and theorems of Petri net theory. Most of them are implemented in our Petri net analysis tool Charlie [22].

Applications. Case studies demonstrating the power of static analysis techniques can be found in [29] (signalling cascade), [25] (biosensor gene regulation), and [28] (signal transduction network).

The comprehensive textbook [52] is focused on the stoichiometric matrix and related analysis techniques. It is also a good entry point for the growing body of related literature. The use of P/T-invariants for the validation of biochemical networks has been introduced to the Petri net community in [30].

Specific application examples include:

- Angeli et al. introduce in [3] a sufficient criterion for persistency of ODEs models, i.e., of continuous Petri nets, which is based on the structure. It ensures persistency for reaction networks with mass/action kinetics and arbitrary parameters. This structural criterion closely resembles a famous liveness criterion of Petri net theory – the Siphon Trap Property, adjusted to the continuous case: the trap is replaced by a P-invariant.

– In [27], we use T-invariants and the related notion of Abstract Dependent Transition sets (ADT sets) to automatically derive hierarchically structured networks.
– In [36], we apply structural analysis, specifically T-invariants, to reduce a given ODEs model, while preserving the essential steady state behaviour. The structural analysis identifies the core network and its fragile node (robustness analysis).
– In [33], we shed new light on the importance of the Siphon Trap Property for continuization (fluidization), which is determined for a number of biochemical networks from our repository.

We expect further results along this line in the future. Petri nets offer a suitable framework for reasoning about the relationship between the three basic modelling paradigms. For this purpose, our tool Snoopy [59] allows the user to move easily between the three worlds, and thus ensuring the equivalence of the related models.

5 Model checking

Model checking is a powerful analysis approach, well-established in Computer Science and now gaining increasing popularity in Systems and Synthetic Biology, which can be performed in the three modelling paradigms. It permits checking the behaviour of a model for certain properties expressed in temporal logics. These are unambiguous languages, which provide flexible formalisms to check the validity of statements (propositions) in relation to the execution of the model. Temporal logics come in different flavours; for example branching-time and linear-time logics. The latter are more attractive to biologists who are used to reason about time series behaviour.

Analytical model checking is usually restricted to LTS descriptions (interleaving semantics). However, we can take advantage of Petri nets' partial order semantics to perform model checking over behaviour descriptions given as PO prefix, which preserve the difference between alternative and concurrent occurrences of reactions. The power of this technique for Systems and Synthetic Biology has not been systematically explored yet.

Analytical model checking may become computationally expensive and generally requires network boundedness. An efficient alternative approach is *simulative model checking*, where analysis is performed over time series traces generated by model simulation, which works both for bounded and unbounded models. It involves two approximations: a finite number of finite simulation traces is only considered. Simulative model checking can perhaps best be characterised as behaviour checking, being applicable not only to continuous or stochastic models, but also to any time series data. This approach can thus be used to compare models with physical systems in terms of data generated by simulations of a model and laboratory experiments.

In [28,29], we have demonstrated how to perform model checking in the three modelling paradigms in a concerted manner.

5.1 Characterising Biochemical Species' Behaviour

The behaviour of biochemical species can be described for quantitative model checking using four distinct descriptive approaches, with increasing specificity; qualitative, semi-qualitative, semi-quantitative and quantitative [18].

Qualitative formulae use derivatives of biochemical species concentrations or mass, determined by the function $d(.)$, and the temporal operators U, F, G to describe the general trend of the behaviour. Semi-qualitative extend qualitative with relative concentrations using the function $max(.)$, which yields the maximal value of a species observed in the whole run. Semi-quantitative extend semi-qualitative with absolute time values by referring to to the predefined system variable *time*. Finally, quantitative extend semi-quantitative with absolute concentration values.

For example, the transient activation of a biochemical species called $Protein$ can be expressed in these approaches with increasing accuracy. The following queries determine the probabilities of the statements to be true using PLTLc.

Qualitative. Protein rises then falls:

$$\mathbf{P}_{=?}\ [\ d(\text{Protein})\ >\ 0\ U\ (\ G(\ d(\text{Protein})\ <\ 0\ )\ )\ ].$$

Semi-qualitative. Protein rises then falls to less than 50% of its peak concentration:

$$\begin{aligned}\mathbf{P}_{=?}\ [\ (\ d(\text{Protein})\ >\ 0\ )\ U\ (\ G(\ d(\text{Protein})\ <\ 0\ )\ \wedge\\ F(\ [\text{Protein}]\ <\ 0.5 * max[\text{Protein}]\ )\ )\ ].\end{aligned}$$

Semi-quantitative. Protein rises then falls to less than 50% of its peak concentration at 60 minutes:

$$\begin{aligned}\mathbf{P}_{=?}\ [\ (\ d(\text{Protein})\ >\ 0\ )\ U\ (\ G(\ d(\text{Protein})\ <\ 0\ )\ \wedge\\ F(\ \text{time}\ =\ 60\ \wedge\ \text{Protein}\ <\ 0.5 * max(\text{Protein})\ )\ )\ ].\end{aligned}$$

Quantitative. Protein rises then falls to less than $100\mu Mol$ at 60 minutes:

$$\begin{aligned}\mathbf{P}_{=?}\ [\ (\ d(\text{Protein})\ >\ 0\ )\ U\ (\ G(\ d(\text{Protein})\ <\ 0\ )\ \wedge\\ F(\ \text{time}\ =\ 60\ \wedge\ \text{Protein}\ <\ 100\ )\ )\ ].\end{aligned}$$

5.2 Applications for model checking

There are five major areas of applications for model checking in Systems and Synthetic Biology [8].

Model validation. Does the model behave in the way we expect?

Model comparison. How similar are two models, independent of their underlying formalisms?

Model searching. In a database of alternative models, which of them exhibit a particular property? This can be used to select among competing descriptions of a system produced by various research groups. Given a large database of models (either a general collection or variants of the same model), one can use model behaviour checking to perform systematic database queries, such as “find all models that show oscillatory behaviour under some conditions” or “find all descriptions of this signaling pathway that are transiently active after growth factor stimulation”.

Model analysis. In a collection of structure-preserving variants of a model (e.g., different concentrations of species representing in-silico gene knock-outs), which models show a certain behaviour? E.g., how many knock-outs lead to a loss of oscillating behaviour?

Model construction. Which modifications of a model lead to a desired property? Modifications can involve changes in kinetic parameters or initial concentrations, but they can also be more complex, for example changing the topology of the model by removing or even adding new components. How to do this efficiently is still an active area of research.

6 Tools

BioModel Engineering of non-trivial case studies requires adequate tool support. We provide a sophisticated toolkit covering the whole framework; publicly available at `http://www-dssz.informatik.tu-cottbus.de`.

Snoopy – tool for modelling and animation/simulation of hierarchically structured graphs, among them qualitative, stochastic, and continuous Petri nets [59], [46], recently extended by coloured counterparts of these net classes [44] and Generalised Hybrid Petri Nets, supporting dynamic partitioning [37].

Charlie – multi-threaded analysis tool of standard Petri net properties and techniques of Petri net theory including structural boundedness check, P/T-invariants, STP, rank theorem, structural reduction. It also supports explicit CTL and LTL model checking [22], mainly for teaching purposes.

Marcie – is built upon an Interval Decision Diagram (IDD) engine and supports, besides the analysis of standard Petri net properties (boundedness, dead states, liveness, reversibility) symbolic CTL model checking of qualitative Petri nets [35] and multi-threaded symbolic CSL model checking of Generalised Stochastic Petri Nets [60], recently extended by rewards. Both model checkers outperform comparable tools on the market for benchmarks of typical biochemical networks [34].
Exact analyses of bounded models are complemented by approximative model checking, which is also suitable for unbounded models. Two approximative engines support fast adaptive uniformisation and distributed Gillespie simulation.

In addition, Snoopy files are directly read by ADAM – a tool applying computational algebra to analyse the dynamics of discrete models [65], and simulation traces generated with Snoopy are read by MC2(PLTLc) – a Monte-Carlo Model Checker for PLTLc [18]. Additionally, Snoopy provides export to several foreign analysis tools, among them Anastasia [67] for efficient computation of bad siphons and STP decision, and SBML import/export, which opens the door to a bunch of tools popular in Systems and Synthetic Biology.

7 Outlook and challenges

A drawback of current modelling approaches, including Petri nets, are their limitation to relatively small networks. Thus overall, one of the major challenges lies in the application of Petri nets to achieve richer descriptions of biological systems than are currently practicable. This includes modelling compartments, locality and mobility of players at the *intracellular level.* Biological systems can be represented as networks which themselves typically contain regular (network) structures, and/or repeated occurrences of network patterns. This organisation occurs in a hierarchical manner, reflecting the physical and spatial organisation of the organism, from the intracellular to the intercellular level and beyond (tissues, organs etc.). Examples of organisation at the intracellular level include: cytosol, nucleus, ribosome, endoplasmic reticulum, Golgi apparatus, and mitochondria.

Further challenges lie in modelling at the *intercellular level*, for examples societies of single-celled organisms (including mixtures of organisms), or multicellular organisms at the tissue or organ level. There are many examples of communication in multicellular organisms including endocrine, juxtacrine, etc.

Although such network models can be designed using standard Petri nets, so far there is no dedicated support for such structuring; it becomes impractical as the size of the networks to be modelled increases. Besides the purely technical aspect due to the impracticality of handling large flat nets, humans need some hierarchy and organisation in what they are designing in order to be able to conceptualise the modelled object. Thus, models should explicitly reflect the organisation in complex biological systems.

There are two established orthogonal concepts in Petri net modelling to manage large-scale networks - structuring by hierarchies and colours. However, their potential for BioModel Engineering has not been systematically explored so far. In order to cope with the challenge of modelling at such a variety of levels, we need expressive modelling techniques embedded as part of a robust BioModelling Engineering approach, supporting, e.g., model version control in a collaborative environment for model construction.

8 Summary

Petri nets are a natural and established notation for describing reaction networks because both share the bipartite property. We have shown that Petri nets can be used to perform all major modelling and simulation approaches central to Systems Biology. Thus, they may serve as a kind of umbrella formalism integrating qualitative and quantitative (i.e. stochastic, continuous, or hybrid) modelling and analysis techniques.

However, as we have indicated, there are several important open challenges posed by modelling biological systems which need to be addressed.

References

1. Aderem, A.: Systems Biology: Its Practice and Challenges. Cell 121(4), 511–513 (2005)
2. Angeli, D.: Boundedness analysis for open Chemical Reaction Networks with mass-action kinetics. J. Natural Computing doi 10.1007/s11047-009-9163-7 (2009)
3. Angeli, D., De Leenheer, P., Sontag, E.: A Petri net approach to the study of persistence in chemical reaction networks. Mathematical biosciences 210(2), 598–618 (2007)
4. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model checking continuous time Markov chains. ACM Trans. on Computational Logic 1(1) (2000)
5. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model checking algorithms for continuous-time Markov chains. IEEE Trans. on Software Engineering 29(6) (2003)
6. Baldan, P., Cocco, N., Marin, A., Simeoni, M.: Petri Nets for Modelling Metabolic Pathways: a Survey. J. Natural Computing (9), 955–989 (2010)
7. Bonzanni, N., Krepska, E., Feenstra, K., Fokkink, W., Kielmann, T., Bal, H., Heringa, J.: Executing multicellular differentiation: quantitative predictive modelling of c. elegans vulval development. Bioinformatics 25, 2049–2056 (2009)
8. Breitling, R., Donaldson, R., Gilbert, D., Heiner, M.: Biomodel Engineering - From Structure to Behavior. Lect. Notes Bioinformatics 5945, 1–12 (2010)
9. Brightman, F., Fell, D.: Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. FEBS letters 482(3), 169–174 (2000)
10. Calder, M., Vyshemirsky, V., Gilbert, D., Orton, R.: Analysis of signalling pathways using continuous time Markov chains. Trans. on Computat. Syst. Biol. VI, LNBI 4220 pp. 44–67 (2006)
11. Calzone, L., Chabrier-Rivier, N., Fages, F., Soliman, S.: Machine learning biochemical networks from temporal logic properties. Trans. on Computat. Syst. Biol. VI, LNBI 4220 pp. 68–94 (2006)
12. Chaouiya, C.: Petri Net Modelling of Biological Networks. Briefings in Bioinformatics 8(4), 210 – 219 (2007)
13. Chen, L., Masao, N., Kazuko, U., Satoru, M.: Simulation-based model checking approach to cell fate specification during c. elegans vulval development by hybrid functional Petri net with extension. BMC Systems Biology 42 (3 2009)
14. Chen, M., Hariharaputran, S., Hofestädt, R., Kormeier, B., Spangardt, S.: Petri net models for the semi-automatic construction of large scale biological networks. J. Natural Computing doi DOI 10.1007/s11047-009-9151-y (2009)
15. Clarke, E., Grumberg, O., Peled, D.: Model checking. MIT Press 1999, third printing (2001)
16. David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer (2010)
17. Doi, A., Drath, R., Nagaska, M., Matsuno, H., Miyano, S.: Protein Dynamics Observations of Lambda-Phage by Hybrid Petri net. Genome Informatics pp. 217–218 (1999)
18. Donaldson, R., Gilbert, D.: A model checking approach to the parameter estimation of biochemical pathways. In: Proc. 6th International Conference on Computational Methods in Systems Biology (CMSB 2008). pp. 269–287. LNCS 5307, Springer (2008)
19. Endy, D.: Foundations for engineering biology. Nature 438(7067), 449–453 (2005)

20. Fehling, R.: A concept of hierarchical Petri nets with building blocks. In: Advances in Petri nets. pp. 148–168. Springer, LNCS 674 (1993)
21. Feinberg, M.: Mathematical aspects of mass action kinetics, chap. 1, pp. 1–78. Prentice-Hall (1977)
22. Franzke, A.: Charlie 2.0 - a multi-threaded Petri net analyzer. Diploma Thesis, Brandenburg University of Technology at Cottbus, CS Dep. (2009)
23. Gilbert, D., Heiner, M.: From Petri nets to differential equations - an integrative approach for biochemical network analysis. In: Proc. ICATPN 2006. pp. 181–200. LNCS 4024, Springer (2006)
24. Gilbert, D., Heiner, M., Lehrack, S.: A unifying framework for modelling and analysing biochemical pathways using Petri nets. In: Proc. CMSB. pp. 200–216. LNCS/LNBI 4695, Springer (2007)
25. Gilbert, D., Heiner, M., Rosser, S., Fulton, R., Gu, X., Trybiło, M.: A Case Study in Model-driven Synthetic Biology. In: Proc. 2nd IFIP Conference on Biologically Inspired Collaborative Computing (BICC), IFIP WCC 2008, Milano. pp. 163–175 (2008)
26. Heinemann, M., Panke, S.: Synthetic biology - putting engineering into biology. Bioinformatics 22(22), 2790–2799 (2006), http://dx.doi.org/10.1093/bioinformatics/btl469
27. Heiner, M.: Understanding Network Behaviour by Structured Representations of Transition Invariants – A Petri Net Perspective on Systems and Synthetic Biology, pp. 367–389. Natural Computing Series, Springer (2009), http://www.springerlink.com/content/m8t30720r141442m
28. Heiner, M., Donaldson, R., Gilbert, D.: Petri Nets for Systems Biology, in Iyengar, M.S. (ed.), Symbolic Systems Biology: Theory and Methods; Chapter 21. Jones and Bartlett Publishers, Inc. (2010)
29. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets in systems and synthetic biology. In: Schools on Formal Methods (SFM). pp. 215–264. LNCS 5016, Springer (2008)
30. Heiner, M., Koch, I.: Petri Net Based Model Validation in Systems Biology. In: Proc. ICATPN, LNCS 3099. pp. 216–237. Springer (2004)
31. Heiner, M., Koch, I., Will, J.: Model Validation of Biological Pathways Using Petri Nets - Demonstrated for Apoptosis. BioSystems 75, 15–28 (2004)
32. Heiner, M., Lehrack, S., Gilbert, D., Marwan, W.: Extended Stochastic Petri Nets for Model-Based Design of Wetlab Experiments. Transactions on Computational Systems Biology XI pp. 138–163 (2009)
33. Heiner, M., Mahulea, C., Silva, M.: On the Importance of the Deadlock Trap Property for Monotonic Liveness. In: Int. Workshop on Biological Processes & Petri Nets (BioPPN), satellite event of Petri Nets 2010 (2010)
34. Heiner, M., Rohr, C., Schwarick, M., Streif, S.: A Comparative Study of Stochastic Analysis Techniques. In: Proc. CMSB. pp. 96–106. ACM digital library 978-1-4503-0068-1/10/09 (2010)
35. Heiner, M., Schwarick, M., Tovchigrechko, A.: DSSZ-MC - A Tool for Symbolic Analysis of Extended Petri Nets. In: Proc. PETRI NETS 2009. pp. 323–332. LNCS 5606, Springer (2009)
36. Heiner, M., Sriram, K.: Structural analysis to determine the core of hypoxia response network. PLoS ONE 5(1), e8600 (2010), http://dx.doi.org/10.1371/journal.pone.0008600
37. Herajy, M., Heiner, M.: Hybrid Petri nets for modelling of hybrid biochemical interactions. In: Proc. AWPN 2010. CEUR Workshop Proceedings, vol. 643, pp. 66–79. CEUR-WS.org (October 2010)

38. Hofestädt, R., Thelen, S.: Quantitative modeling of biochemical networks. In Silico Biol. 1, 39–53 (1998)
39. Hood, L.: Systems biology: integrating technology, biology, and computation. Mechanisms of Ageing and Development 124(1), 9–16 (2003)
40. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: Copasi – a complex pathway simulator. BioInformatics 22(24), 3067–3074 (2006)
41. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., et al., H.K.: The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. J. Bioinformatics 19, 524–531 (2003)
42. Lamprecht, R., Smith, G., Kemper, P.: Stochastic Petri net models of Ca^{2+} signaling complexes and their analysis. J. Natural Computing doi 10.1007/s11047-009-9143-y (2009)
43. Lautenbach, K., Pinl, A.: A Petri net representation of Bayesian message flows: importance of Bayesian networks for biological applications. J. Natural Computing doi 10.1007/s11047-009-9142-z (2009)
44. Liu, F., Heiner, M.: Computation of enabled transition instances for colored petri nets. In: Proc. AWPN 2010. CEUR Workshop Proceedings, vol. 643, pp. 51–65. CEUR-WS.org (October 2010)
45. Marsan, M.A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. Wiley Series in Parallel Computing, John Wiley and Sons (1995), 2nd Edition
46. Marwan, W., Rohr, C., Heiner, M.: Petri nets in Snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks; Chapter 21. Humana Press, book series "Methods in Molecular Biology" (2011)
47. Marwan, W., Sujathab, A., Starostzik, C.: Reconstructing the regulatory network controlling commitment and sporulation in *Physarum polycephalum* based on hierarchical Petri net modeling and simulation. J. of Theoretical Biology 236(4), 349–365 (2005)
48. Marwan, W., Wagler, A., Weismantel, R.: Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks. J. Natural Computing doi 10.1007/s11047-009-9152-x (2009)
49. Merlin, P.: A Study of the Recoverability of Computing Systems. Irvine: Univ. California, PhD Thesis (1974), available from Ann Arbor: Univ Microfilms, No. 75–11026
50. Nagasaki, M., Saito, A., Doi, A., Matsuno, H., Miyano, S.: Foundations of Systems Biology Using Cell Illustrator and Pathway Databases. Springer, Series: Computational Biology, Vol. 13 (2009)
51. Pagnoni, A.: Error-correcting Petri nets. J. Natural Computing doi 10.1007/s11047-009-9150-z (2009)
52. Palsson, B.: Systems Biology: Properties of Reconstructed Networks. Cambridge University Press (2006)
53. Parker, D., Norman, G., Kwiatkowska, M.: PRISM 3.0.beta1 Users' Guide (2006)
54. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science (FOCS-77). pp. 46–57. IEEE, IEEE Computer Society Press, Providence, Rhode Island (1977)
55. Popova, L., Heiner, M.: Quantitative evaluation of time petri nets and applications to technical and biochemical networks. In: Proc. Int. Workshop on Concurrency, Specification and Programming (CS&P 2007), Lagów. vol. 2, pp. 473–484 (2007)

56. Popova-Zeugmann, L.: On time Petri nets. J. Inform. Process. Cybern. EIK 27(4), 227–244 (1991)
57. Popova-Zeugmann, L.: Quantitative evaluation of time-dependent Petri nets and applications to biochemical networks. J. Natural Computing doi 10.1007/s11047-010-9211-3 (2010)
58. Popova-Zeugmann, L., Heiner, M., Koch, I.: Time Petri Nets for Modelling and Analysis of Biochemical Networks. Fundamenta Informaticae 67 (2005)
59. Rohr, C., Marwan, W., Heiner, M.: Snoopy - a unifying Petri net framework to investigate biomolecular networks. Bioinformatics 26(7), 974–975 (2010)
60. Schwarick, M., Heiner, M.: CSL model checking of biochemical networks with Interval Decision Diagrams. In: Proc. CMSB. pp. 296–312. Springer, LNCS/LNBI 5688 (2009)
61. Shinar, G., Feinberg, M.: Structural Sources of Robustness in Biochemical Reaction Networks. Science 327, 1389–1391 (Mar 2010)
62. Silva, M., Recalde, L.: Petri nets and integrality relaxations: A view of continuous Petri net models. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 32(4), 314–327 (2002)
63. Soliman, S., Heiner, M.: A unique transformation from ordinary differential equations to reaction networks. PlosONE 5(12), e14284 (2010)
64. Valk, R.: Self-modifying nets, a natural extension of Petri nets. LNCS, Automata, Languages and Programming 62, 464–476 (1978)
65. Veliz-Cuba, A., Jarrah, A., Laubenbacher, R.: Polynomial algebra of discrete models in systems biology. Bioinformatics 26(13), 1637–1643 (2010)
66. Wagler, A., Weismantel, R.: The combinatorics of modeling and analyzing biological systems. J. Natural Computing (2009, 101007/s11047-009-9165-5)
67. Wimmel, H.: Anastasia – computing traps, siphons, and more http://service-technology.org/anastasia (2010)