Hybrid Petri Nets for Modelling the Eukaryotic Cell Cycle

Mostafa Herajy[1], Martin Schwarick[2], and Monika Heiner[2]

[1] Port Said University, Faculty of Science,
Department of Mathematics and Computer Science,
42521 - Port Said, Egypt
[2] Brandenburg University of Technology at Cottbus,
Computer Science Institute,
Data Structures and Software Dependability,
Postbox 10 13 44, 03044 Cottbus, Germany
http://www-dssz.informatik.tu-cottbus.de/

Abstract. System level understanding of the repetitive cycle of cell growth and division is crucial for disclosing many unknown principles of biological organisms. The deterministic or stochastic approach – when deployed separately – are not sufficient to study such cell regulation due to the complexity of the reaction network and the existence of reactions at different time scales. Thus, an integration of both approaches is advisable to study such biochemical networks. In this paper we show how Generalised Hybrid Petri Nets can be used to intuitively represent and simulate the eukaryotic cell cycle. Our model captures intrinsic as well as extrinsic noise and deploys stochastic as well as deterministic reactions. Additionally, marking-dependent arc weights are biologically motivated and introduced to Snoopy – a tool for animating and simulating Petri nets in various paradigms.

Keywords: Generalised hybrid Petri nets, hybrid modelling, eukaryotic cell cycle, Snoopy, marking-dependent arc weight.

1 Introduction

The reproduction of eukaryotic cells is controlled by a complex regulatory network of reactions known as cell cycle [19,20,24]. During a cell cycle, cells grow, replicate and divide into two daughter cells [13,21]. This regulation cycle consists of four phases: S phase (synthesis) and M phase (mitosis) separated by two gap phases: G1 and G2 [24]. During the S phase, the cell replicates all of its components, while it divides each component more or less evenly between the two daughter cells at the end of the M phase [13]. After the S phase, there is a gap (G2) where the cell ensures that the duplication of DNA has been completed and prepares itself for mitosis. Newborn cells are not immediately replicated, instead they are located at the G1 gap. The processes of synthesis and mitosis alternate during the reproduction process; see Figure 1 for a graphical illustration of the cell cycle regulation process. Please note that the phases G1, S, and

M. Koutny et al. (Eds.): ToPNoC VIII, LNCS 8100, pp. 123–141, 2013.

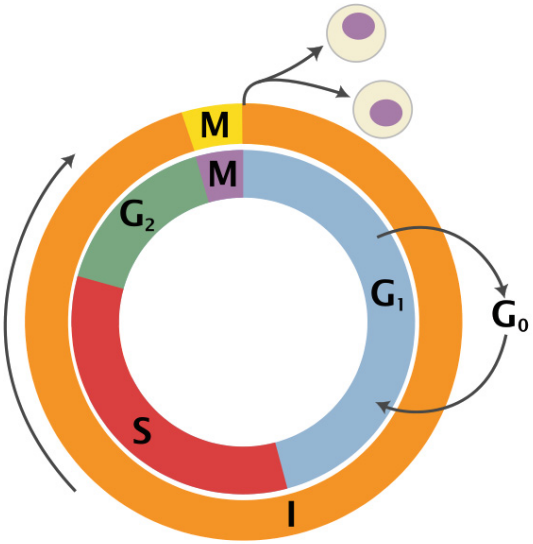


Fig. 1. Graphical illustration of the cell cycle [27]. The cell cycle consists of four distinct phases: G_1, synthesis (S), G_2, and mitosis (M), respectively. The first three phases are known as interphase (referred to by the outer ring). Cells that have stopped dividing enter the G_0 phase.

G2 are commonly subsumed as interphase as indicated by the outer cycle in that figure. Understanding such control cycles is crucial for revealing defects in cell growth that underlies many human diseases (e.g., cancer) [25].

In the eukaryotic cell cycle, the alternation between the S and the M phase as well as the balance of growth and division is governed by the activity of a family of cyclin-dependent protein kinases (CDK) [24]. Therefore, many computational models have been constructed to study the control system of CDK (e.g., in [1,13,19,20,24]). Some of these models are based on the deterministic approach which represents changes of species concentrations as continuous variables which evolve deterministically and continuously with respect to time (in the following called *continuous simulation*). However, an important requirement of the cell cycle model is to capture the variability of the cellular volume to reproduce the "in vivo" experiment results. Unfortunately, the deterministic approach cannot capture such cellular volume variability [20]. Motivated by this argument, a number of stochastic models have been created and simulated using either a stochastic simulation algorithm (e.g., [13]) or by introducing noise to the model through Langevin equation [22]. However, the stochastic approach is computationally expensive, particularly when the model under study contains reactions with high rates and/or species with large numbers of molecules.

The eukaryotic cell cycle model does indeed exhibit high rates of some reactions, while some other reactions have low rates, which are responsible for the

intrinsic noise due to molecular fluctuations [13]. Similarly, the model contains some species with a large number of molecules, while some other species have a few number of molecules. The existence of reactions at different time scales (fast and slow) suggests a simulation using a hybrid approach.

Generalised Hybrid Petri Nets ($\mathcal{GHPN}_{bio}$) have been introduced in [10], [11] and [12], to represent and simulate stiff biochemical networks where fast reactions are represented and simulated continuously, while slow reactions are carried out stochastically. $\mathcal{GHPN}_{bio}$ provide rich modelling and simulation functionalities by combining all features of Continuous Petri Nets [3] and Extended Stochastic Petri Nets [16], including three types of deterministic transitions. Moreover, the partitioning of reaction networks (i.e., the assignment of each reaction to either the stochastic or the continuous paradigm) can either be done off-line (statically, i.e., before the simulation starts) or on-line (dynamically, i.e., while the simulation is in progress). The implementation of $\mathcal{GHPN}_{bio}$ is available as part of Snoopy [7] - a tool to design and animate or simulate hierarchical graphs, among them qualitative, stochastic, continuous and hybrid Petri nets. Indeed, the cell cycle model turns out to be an ideal case study where the majority of the $\mathcal{GHPN}_{bio}$ features can be demonstrated. Moreover, it makes a strong case for the introduction of marking-dependent arc weights.

Another hybrid net class which provides functionalities related to $\mathcal{GHPN}_{bio}$ is known as Hybrid Functional Petri nets (HFPN) [18]. However, HFPN have been developed to focus on hybrid (discrete/continuous) model construction where stochastic transitions are not required. Moreover, modelling features like logical nodes, hierarchy, and modifier arcs, which are imperative when considering larger models, are not supported [7].

In this paper we present another argument to motivate hybrid simulation of the cell cycle control system. The cell cycle model contains some reactions which would be better represented as continuous processes, specifically the growth of the cellular volume needs to be treated continuously, while other reactions of low rates have to be considered as stochastic processes. For instance, Mura and Csikasz-Nagy constructed in [19] a stochastic version of the model in [1] using stochastic Petri nets. However, they could not intuitively represent the cell growth process which evolves continuously and exponentially with respect to time using stochastic Petri net primitives only. Indeed, cell growth is a typical example where continuous transitions are an appropriate means.

This paper is organised as follows: we start off by pinpointing some related work. After that, a brief introduction of Generalised Hybrid Petri Nets is presented. To conveniently model the cell cycle regulation behaviour, we extend the formal definition of $\mathcal{GHPN}_{bio}$, as they have been introduced in [10], to include marking-dependent arc weights. Next, we discuss a hybrid Petri net model of the eukaryotic cell cycle and discuss in detail some of its key modelling components. In Section 5 we show the simulation results produced by Snoopy's hybrid simulation engine and compare them to the continuous and stochastic ones. Finally, we sum up with conclusions and outlook.

2 Related Work

Mura and Csikasz-Nagy converted the deterministic model of Chen et al. [1] into a stochastic Petri net [19] to study the effect of noise on cell cycle progression. However, some components could not intuitively be modelled using stochastic Petri net primitives only (e.g., cell growth). Moreover, their model is based on phenomenological rate laws (e.g., Michaelis-Menten) which do not work well with stochastic simulation algorithms [13]. Sabouri-Ghomi et al. [20], and Kar et al. [13] asserted that applying Gillespie's stochastic simulation algorithm [4,5] directly to phenomenological rate laws might produce incorrect results. Therefore, they unpacked the deterministic model of Tyson-Novak [24] (who use non-elementary reaction kinetics, e.g., Michaelis Menten and Hill functions) to express it completely in terms of elementary mass-action kinetics. The Tyson-Novak model is based on a bistable switch between the complex CycB-Cdk1 (denoted by variable X) and the complex Cdh1-APC (denoted by the variable Y). CycB-Cdk1 phosphorylates Cdh1-APC and free Cdh1-APC catalyses the degradation of CycB-Cdk1. Figure 2 presents a continuous Petri net representation of the Tyson-Novak model. To model a complete cell cycle, Kar et al. [13] unpacked the effect of Cdc20 and Cdc14 which are lumped in the variable Z in the Tyson-Novak model. High activity of CycB-Cdk1 promotes the synthesis of Cdc20 which activates Cdc14. Finally the dephosphorylated Cdc14 activates Cdh1-APC. The Kar et al. model accounts for both intrinsic and extrinsic noise. Intrinsic noise is due to the fluctuation of species with low numbers of molecules, while extrinsic noise is due to the unequal division of the cell between the two daughter cells [13].

In [2] and [17], two detailed HFPN models are constructed for the Fission yeast and Xenopus cell cycles, respectively. However, intrinsic noise, which is necessary for reproducing the variability of the cellular volume, is not captured because HFPN do not support the (full) interplay between stochastic and continuous regimes. Thus, these models are built using the hybrid (discrete/continuous) paradigm.

In [21], a hybrid model, which combines ordinary differential equations (ODEs) and discrete boolean networks, has been constructed to integrate quantitative as well as qualitative parts in one model. The latter approach requires less knowledge of realistic kinetic rate constants. Liu et al. [15] simulate the stochastic model of [13] using the Haseltine and Rawlings approach [6]. However, such models cannot be represented structurally or graphically which makes their maintenance and extension more diffecult.

In this paper a hybrid Petri net model of the eukaryotic cell cycle is presented as a sophisticated example for the kind of hybrid models that can be constructed using $\mathcal{GHPN}_{bio}$. The model is hybrid in the sense that it combines continuous, stochastic and immediate transitions to represent deterministic, stochastic and control behaviour. Our main goal is to show how such a class of models is intuitively represented and executed using hybrid Petri net primitives. Besides, Petri nets analysis tools can be applied to the constructed models as well [8]. Using Snoopy's simulator, cell cycle models incorporating continuous net components can be simulated using either the continuous or hybrid engine.

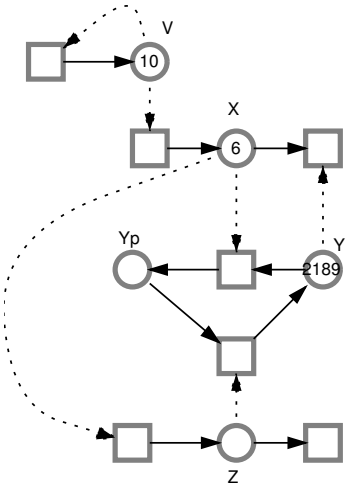


Fig. 2. A continuous Petri net representation of the Tyson-Novak model [24]: X (*CycB-Cdk1*) phosphorylates Y (*Cdh1-APC*) and free Y catalyses the degradation of X. Z denotes the effects of *Cdc20* and *Cdc14*. High activity of X promotes the synthesis of *Cdc20* which activates *Cdc14*. The dephosphorylated *Cdc14* activates Y. This behaviour results in a bistable switch that is responsible for the transitions between *G1* and *S-G2-M* states.

3 Generalised Hybrid Petri Nets

To model stiff biochemical networks, $\mathcal{GHPN}_{bio}$ [10] combine both stochastic and continuous elements in one and the same model. Indeed, continuous and stochastic Petri nets complement each other. Fluctuation and discreteness can conveniently be modelled and simulated in the stochastic paradigm and at the same time, the computational expensive parts can be simulated deterministically via ODE solvers. Modelling and efficient simulation of stiff biochemical networks (i.e., networks that contain reactions at more than one time scale) are helpful functionalities that $\mathcal{GHPN}_{bio}$ provide for systems biology.

Generally speaking, biochemical systems can involve reactions from more than one type of biological networks, for instance gene regulation, metabolic pathways or signal transduction pathways. Incorporating reactions which belong to distinct (biological) network types, tends to result into stiff systems. This follows from the fact that, e.g., species in gene regulation networks may contain few numbers of molecules, while species in metabolic networks often contain large numbers of molecules [14].

In the rest of this section, we will give a brief introduction of $\mathcal{GHPN}_{bio}$ in terms of the graphical representation of its elements as well as the firing rule and connectivity between the continuous and stochastic net parts. The formal semantics is given in [10].

3.1 Elements

The $\mathcal{GHPN}_{bio}$ elements are classified into three categories: places, transitions, and arcs.

$\mathcal{GHPN}_{bio}$ offer two types of places: discrete and continuous. A discrete place (single line circle) holds a non-negative integer number which represents, e.g., the number of molecules of a given species (tokens in Petri net notions). A continuous place (shaded line circle) holds a non-negative real number which represents, e.g., the concentration of a given species.

Furthermore, $\mathcal{GHPN}_{bio}$ offer five transition types: stochastic, immediate, deterministically delayed, scheduled, and continuous transitions [8]. Stochastic transitions, which are drawn in Snoopy as a square, fire with an exponentially distributed random delay. The user can specify a set of firing rate functions, which determine the random firing delay. The transitions' pre-places can be used to define the firing rate functions of stochastic transitions. Immediate transitions (black bar) fire with zero delay, and have always highest priority to fire. They may carry weights which specify the relative firing frequency in the case of conflicts between immediate transitions. Deterministically delayed transitions (black square) fire after a specified constant time delay. Scheduled transitions (grey square) fire at user-specified absolute time points. Continuous transitions (shaded line square) fire continuously in the same way like in continuous Petri nets. Their semantics is governed by ODEs which define the continuous change in the transitions' pre- and post-places. More details about the biochemical interpretation of deterministically delayed, scheduled, and immediate transitions can be found in [9] and [16]. To simplify the presentation, we occasionally refer to stochastic, immediate, deterministically delayed or scheduled transitions as discrete transitions.

The connection between those two types of nodes (places and transitions) takes place using a set of different arcs (edges). $\mathcal{GHPN}_{bio}$ offer six types of arcs: standard, inhibitor, read, equal, reset and modifier arcs. Standard arcs connect transitions with places or vice versa. They can be discrete, i.e., carry non-negative integer-valued weights (stoichiometry in the biochemical context), or continuous, i.e., carry non-negative real-valued weights. In addition to their influence on the enabling of transitions, they also affect the place marking when a transition fires by removing (adding) tokens from (to) the transition's pre-places (post-places).

Extended arcs like inhibitor, read, equal, reset, and modifier arcs can only be used to connect places with transitions, but not vice versa. A transition connected with an inhibitor arc is enabled (with respect to this pre-place) if the marking of the pre-place is less than the arc weight. Contrary, a transition connected with a read arc is enabled if the marking of the pre-place is greater than or equal to the arc weight. Similarly, a transition connected using an equal arc is enabled if the marking of the pre-place is equal to the arc weight.

The other two remaining arcs do not affect the enabling of transitions. A reset arc is used to reset a place marking to zero when the corresponding transition fires. Modifier arcs permit to include any place in the transitions' rate functions and simultaneously preserve the net structure restriction.

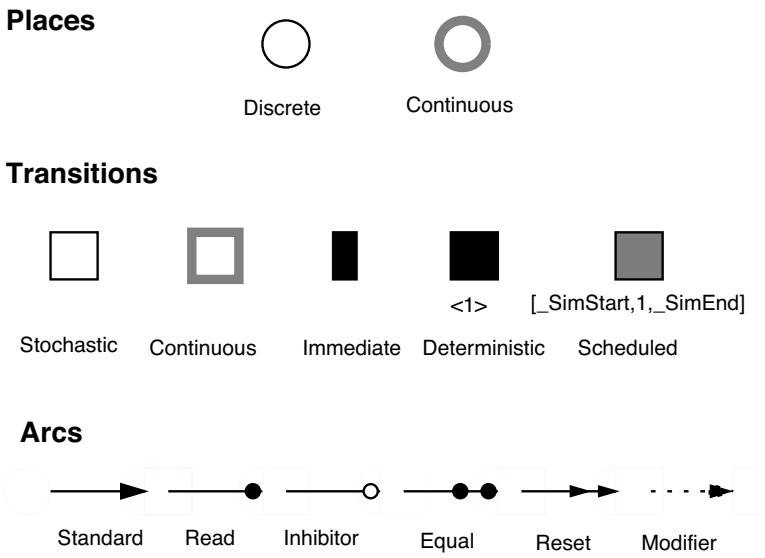


Fig. 3. Graphical representation of the $\mathcal{GHPN}_{bio}$ elements. Places are classified as discrete and continuous, transitions as stochastic, continuous, immediate, deterministically delayed, and scheduled, and arcs as standard, inhibitor, read, equal, reset, and modifier.

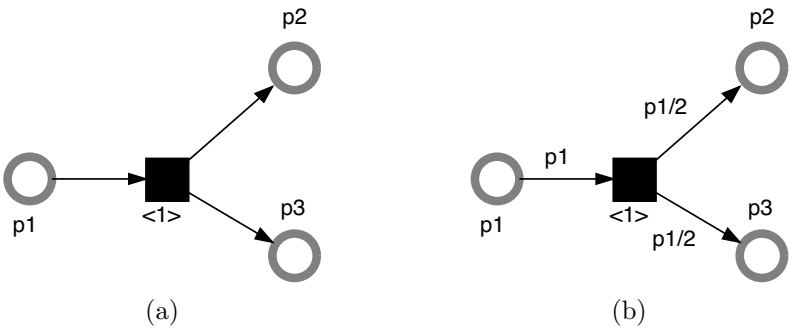


Fig. 4. Marking-dependent arc weights illustrated by a simple biological example. (a) cell division cannot be modelled, (b) cell division can intuitively be modelled. The numbers between angle brackets are the delays of the deterministically delayed transitions. Later we will assume that cell division does not consume time.

The connection rules and their underlying formal semantics are discussed in more details below. Figure 3 provides a graphical illustration of all elements. Although this graphical notation is the default one, it can easily be customised using Snoopy, the Petri nets editing tool. To support special modelling requirements of some biological models (e.g., the cell cycle model), we extended $\mathcal{GHPN}_{bio}$ to permit pre-places of a transition as arc weight, similar to the idea

of self-modifying nets which has been originally introduced in [26], or even a function which is defined in terms of a transition's pre-places [18].

Consider the following simple biological example. When a cell divides its mass between two daughter cells, each daughter obtains approximately half of the mass. This example cannot easily be modelled using discrete Petri nets. Moreover, there is no way to model it if the mass is represented by a continuous place as shown in Figure 4a. In Figure 4b, using marking-dependent arc weights; the ingoing arc of the transition t has a weight equal to the marking of the place p_1, while each of the two outgoing arcs has a weight equal to half of the marking of that place.

Motivated by the case study discussed in this paper, marking-dependent arc weights have been introduced for the majority of arc types supported by Snoopy (standard, read, inhibitor, and equal arc). For more details see Section 4.2.

3.2 Connection Rules

An important question arises when considering the combination of discrete and continuous elements: how are these two different parts connected with each other? Figure 5 provides a graphical illustration of how the connection between different elements of $\mathcal{GHPN}_{bio}$ takes place.

First, we will consider the connection between continuous transitions and the other elements of $\mathcal{GHPN}_{bio}$. Continuous transitions can be connected with continuous places in both directions using continuous arcs (i.e., arcs with real-valued weight). This means that continuous places can be pre- or post-places of continuous transitions. These connections typically represent deterministic biological interactions.

Continuous transitions can also be connected with discrete places, but only by one of the extended arcs (inhibitor, read, equal, and modifier). This type of connection allows a link between discrete and continuous parts of a biochemical model.

Discrete places are not allowed to be connected with continuous transitions using standard arcs, because the firing of continuous transitions is governed by ODEs which require real values in the pre- and post-places. Hence, this cannot take place in the discrete world.

Second, discrete transitions can be connected with discrete or continuous places in both directions using standard arcs. However, the arc weight needs to be considered. The connection between discrete transitions and discrete places takes place using arcs with non-negative integer numbers, while the connection between continuous places and discrete transitions is weighted by non-negative real numbers. The general rule to determine the weight type of arcs is to follow the type of the connected place.

3.3 Formal Definition

In this section, the syntax of $\mathcal{GHPN}_{bio}$ is formally defined to include the making-dependent arc weight. The formal semantics including the enabling and firing rules as well as the conflict resolution are given in [10].

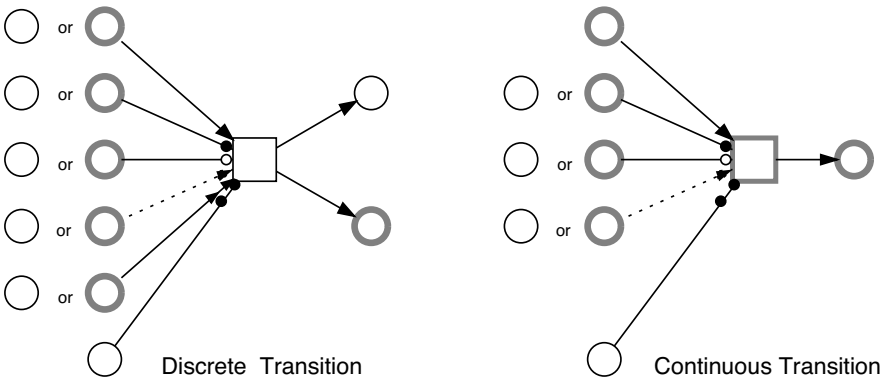


Fig. 5. Possible connections between $\mathcal{GHPN}_{bio}$ elements. The restrictions are: discrete places cannot be connected with continuous transitions using standard arcs, continuous places cannot be tested with equal arcs, and continuous transitions cannot use reset arcs.

Definition 1 (Generalised Hybrid Petri Nets). *A Generalised Hybrid Petri Net is a 6-tuple* $\mathcal{GHPN}_{bio} = [P, T, A, F, V, m_0]$, *where* P, T *are finite, non-empty and disjoint sets.* P *is the set of places, and* T *is the set of transitions with:*

- $P = P_{disc} \cup P_{cont}$ *whereby* P_{disc} *is the set of discrete places to which non-negative integer values are assigned, and* P_{cont} *is the set of continuous places to which non-negative real values are assigned.*
- $T = T_D \cup T_{cont}$,
  $T_D = T_{stoch} \cup T_{im} \cup T_{timed} \cup T_{scheduled}$ *with:*
  1. T_{stoch} *is the set of stochastic transitions, which fire randomly after exponentially distributed waiting time.*
  2. T_{im} *is the set of immediate transitions, which fire with waiting time zero; they have higher priority compared with other transitions.*
  3. T_{timed} *is the set of deterministically delayed transitions, which fire after a deterministic time delay.*
  4. $T_{scheduled}$ *is the set of scheduled transitions, which fire at predefined time points.*
  5. T_{cont} *is the set of continuous transitions, which fire continuously over time.*
- $A = A_{disc} \cup A_{cont} \cup A_{inhibit} \cup A_{read} \cup A_{equal} \cup A_{reset} \cup A_{modifier}$ *is the set of directed arcs, with:*
  1. $A_{disc} \subseteq ((P \times T) \cup (T \times P))$ *defines the set of discrete arcs.*

2. $A_{cont} \subseteq ((P_{cont} \times T) \cup (T \times P_{cont}))$ *defines the set of continuous arcs.*
3. $A_{read} \subseteq (P \times T)$ *defines the set of read arcs.*
4. $A_{inhibit} \subseteq (P \times T)$ *defines the set of inhibits arcs.*
5. $A_{equal} \subseteq (P_{disc} \times T)$ *defines the set of equal arcs.*
6. $A_{reset} \subseteq (P \times T_D)$ *defines the set of reset arcs,*
7. $A_{modifier} \subseteq (P \times T)$ *defines the set of modifier arcs.*

– *the function* F

$$F : \begin{cases} A_{cont} \rightarrow D_q, \\ A_{disc} \rightarrow D_n, \\ A_{read} \rightarrow D_q, \\ A_{inhibit} \rightarrow D_q, \\ A_{equal} \rightarrow D_n, \\ A_{reset} \rightarrow \{1\}, \\ A_{modifier} \rightarrow \{1\}. \end{cases}$$

assigns a marking-dependent function to each arc, where D_n *and* D_q *are sets of functions defined as follows:*

$$D_n = \{d_n | d_n : \mathbb{N}_0^{|^\bullet t_j|} \rightarrow \mathbb{N}, t_j \in T\},$$

$$D_q = \{d_q | d_q : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{Q}^+, t_j \in T\}.$$

– V *is a set of functions* $V = \{g, d, w, f\}$ *where :*

1. $g : T_{stoch} \rightarrow H_s$ *is a function which assigns a stochastic hazard function* h_{s_t} *to each transition* $t_j \in T_{stoch}$*, whereby* $H_s = \{h_{s_t} | h_{s_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{stoch}\}$ *is the set of all stochastic hazard functions, and* $g(t_j) = h_{s_t}, \forall t_j \in T_{stoch}$*.*
2. $w : T_{im} \rightarrow H_w$ *is a function which assigns a weight function* h_w *to each immediate transition* $t_j \in T_{im}$*, such that* $H_w = \{h_{w_t} | h_{w_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{im}\}$ *is the set of all weight functions, and* $w(t_j) = h_{w_t}, \forall t_j \in T_{im}$*.*
3. $d : T_{timed} \cup T_{scheduled} \rightarrow \mathbb{R}_0^+$*, is a function which assigns a constant time to each deterministically delayed and scheduled transition representing the (relative or absolute) waiting time.*
4. $f : T_{cont} \rightarrow H_c$ *is a function which assigns a rate function* h_c *to each continuous transition* $t_j \in T_{cont}$*, such that* $H_c = \{h_{c_t} | h_{c_t} : \mathbb{R}_0^{|^\bullet t_j|} \rightarrow \mathbb{R}_0^+, t_j \in T_{cont}\}$ *is the set of all rates functions and* $f(t_j) = h_{c_t}, \forall t_j \in T_{cont}$*.*

– $m_0 = m_{disc} \cup m_{cont}$ *is the initial marking for both the continuous and discrete places, whereby* $m_{cont} \in \mathbb{R}_0^{|P_{cont}|}$, $m_{disc} \in \mathbb{N}_0^{|P_{disc}|}$.

Here, $\mathbb{N}$ *denotes the set of natural numbers excluding 0,* $\mathbb{N}_0$ *denotes the set of non-negative integer numbers,* $\mathbb{R}_0$ *denotes the set of non-negative real numbers,* $\mathbb{Q}^+$ *denotes the set of positive rational numbers, and* ${}^\bullet t_j$ *denotes the set of pre-places of a transition* t_j.

□

A distinguishing feature of $\mathcal{GHPN}_{bio}$ compared with other hybrid Petri net classes is its support of the full interplay between stochastic and continuous transitions. Such interplay is implemented by updating and monitoring the rates of stochastic transitions while numerically solving the set of ODEs induced by the continuous transitions (For more details see [10]). By this way, accurate results are obtained during simulation.

4 The Model

Figure 6 shows the hybrid Petri net model which has been developed based on the previous one introduced by Kar et al. in [13]. Proteins, genes, and mRNAs are represented by places, reactions by transitions. We use the same kinetic parameters and initial values as in [13]. For the sake of space we do not repeat the kinetic parameters, but the initial marking is shown on the places. Moreover, we use Snoopy's logical node feature to simplify the connections between nodes. For example, place X and Y are involved in many reactions which decreases the network's readability. We repeat those nodes multiple times with the same names to keep the model understandable (logical places). Likewise, the transition *divide* is a logical transition. Furthermore, the increase of the cellular volume is intuitively represented using a continuous transition with a rate $\mu \cdot V$, where μ is the growth factor and V is the cellular volume, modelled as a continuous place.

The model contains three different transition types: continuous, stochastic, and immediate. Continuous transitions simulate the corresponding reactions deterministically, while stochastic transitions carry them out stochastically. The latter transitions are responsible for molecular fluctuations. Immediate transitions monitor the model evolution and perform the division when the free number of molecules of *Cdh1_APC* reaches a certain threshold ($\hat{Y} = Y + Y\_X + X\_Y$).

In the sequel we discuss in more detail some of the model's key components and the corresponding $\mathcal{GHPN}_{bio}$ representations.

4.1 Decision to Perform Division

In this section we consider the process of division in more detail. When the number of molecules of $\hat{Y}$ becomes greater than a certain threshold (in our case

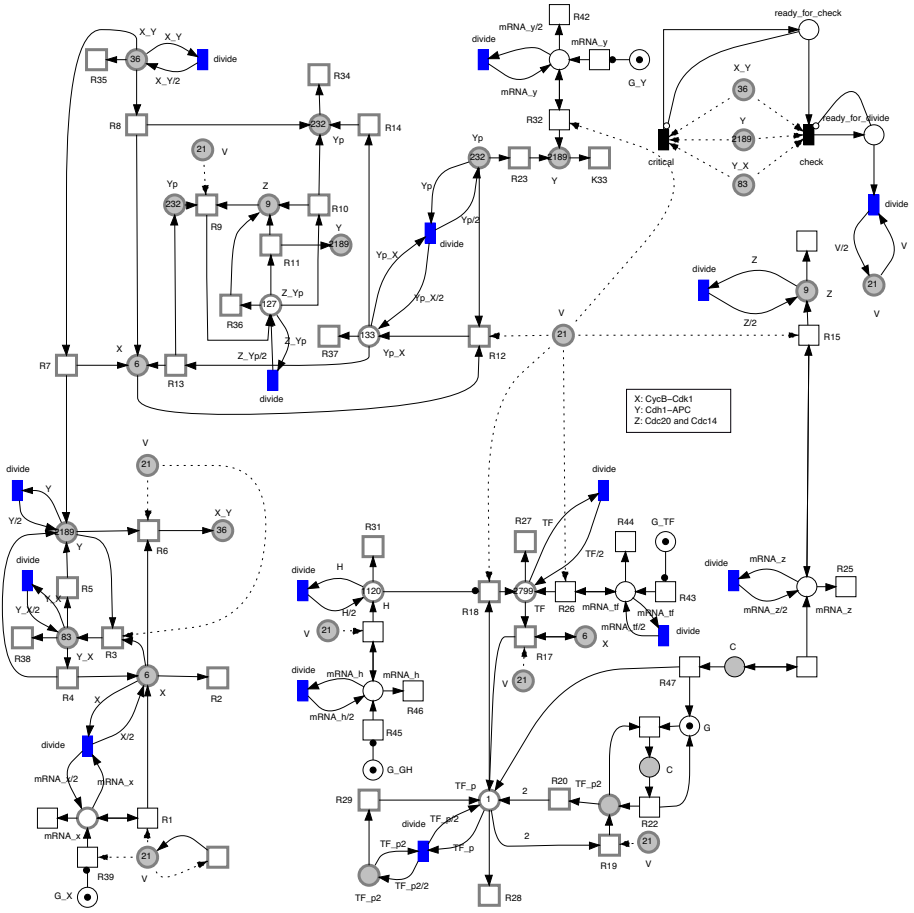


Fig. 6. A $\mathcal{GHPN}_{bio}$ representation of the eukaryotic cell cycle. The model employs different types of transitions: continuous, stochastic and immediate. All reactions affecting mRNAs are represented and simulated stochastically. Repetitive nodes (places and transitions) with same names are logical nodes. When the immediate transition *divide* fires, it divides the current place marking more or less equally. Equal division means that the cellular volume of the daughter cell is always half of its parent. This model could be easily extended to permit unequal division, where a random variation in the cellular volume is possible, by having arc weights with random functions. The unequal division type will reproduce extrinsic noise. The type of division (equal, or unequal) depends on the outgoing arc weight and its effect is implemented by marking-dependent arc weights.

1200), the cell divides the cellular volume and other components (e.g., mRNAs) between the two daughter cells. In Figure 7a, this process is represented by the immediate transition *check* with a weight defined by the Boolean expression $\hat{Y} > threshold$ (the weight is 0 if the Boolean expression yields false, and 1 for the result true). Recall that weights of immediate transitions determine the firing frequencies of immediate transitions in the case of conflicts. A weight of zero means that a transition cannot fire at all. However, when the transition *check* has a weight of one, it adds a token to the place *ready_to_divide* which triggers the transition *divide* to carry out the division. To give the transition *divide* a chance to fire before re-checking the value of $\hat{Y}$, an inhibitor arc is used as constraint. Please note that the transitions *critical* and *check* need the current marking of the places X_Y, Y, and Y_X only to calculate the term $\hat{Y}$ in the transitions' weight. Therefore, modifier arcs are used to fulfil this requirement.

An interesting characteristics of the model is the division process. Although the division can take place when the value of $\hat{Y}$ is greater than a certain threshold, it does not do that all the times. For example, at the beginning of the simulation, the initial value of $\hat{Y}$ satisfies the division criteria. However; the cell should not divide because it is still at G1 phase which means that it has to replicate itself before it can divide. We model these cases by adding a new immediate transition which detects the critical value of $\hat{Y}$, before checking for division. Therefore the transition *critical* monitors the value of $\hat{Y}$. When the value of $\hat{Y}$ goes below a certain threshold, it enables the division process.

4.2 Cell Division and Marking-Dependent Arc Weights

When a cell divides, it splits all of its components more or less evenly between the two daughter cells. This is most naturally expressed with marking-dependent arc weights [26]. In Figure 7a, when the transition *divide* fires, it removes all of the current marking of the place *V* and adds *V/2* to it. To permit uneven division of the cell volume and other components, arc weights can be a function which operates on the current place marking [18]. However, we restrict the places used in arc weights to a transition's pre-places to keep the locality principle Petri nets are famous for.

Figure 7b illustrates the process of cell division graphically by showing a simulation trace.

Moreover, all proteins and mRNAs have to undergo such division. This means the transition *divide* has to be connected with each place in the net that represents a protein or mRNA. The ingoing arc weight of such a connection is equal to the pre-place's current marking, while the outgoing arc weight is equal to half of the pre-place's current marking. Furthermore, the markings of discrete places are rounded after the division process to preserve the discrete representation of the molecular species.

4.3 Transition Partitioning

The model in Figure 6 contains transitions which fire at different rates. For instance, transition R_3 fires more frequently than R_1 as illustrated in Figure

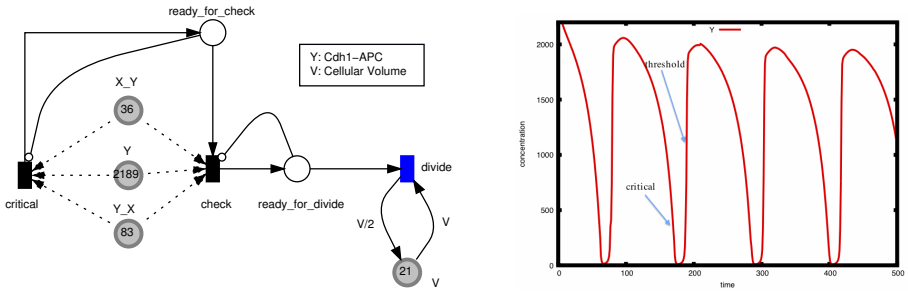


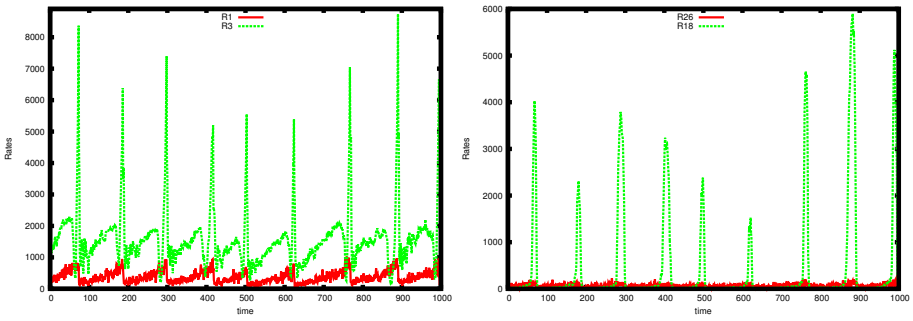
Fig. 7. Cell Division (a) A sub-net for modelling the decision of the division process (see also upper right corner of Figure 6). The transition *critical* monitors the value of $\hat{Y}$ and adds a token to *ready_for_check* when $\hat{Y} < 300$. Later, when the value of $\hat{Y}$ increases and becomes greater than a certain threshold (1200), the transition *check* fires and adds a token to *ready_for_divide* which signals the transition *divide* to perform the division. Inhibitor arcs are used as checkpoints for the sequence of events: critical $\rightarrow$ check $\rightarrow$ divide. (b) Hybrid simulation trace of cell division.

8a. Slow transitions should be simulated stochastically to account for molecular fluctuations, while fast transitions need to be simulated continuously for the sake of numerical efficiency. Indeed, transitions of the latter type consume the majority of computational resources.

In this model, transitions are statically partitioned before the simulation starts. The transition type is determined by executing a single run and analysing the results as shown in Figure 8. Increasing (decreasing) the accuracy of the simulation results involves converting more continuous (stochastic) transitions into stochastic (continuous) ones.

Another approach for partitioning is to perform it dynamically during the simulation. Using this technique, a transition changes its type from stochastic to continuous or vice versa according to the current firing rate. $\mathcal{GHPN}_{bio}$ provide the user with a trade-off between efficiency and accuracy by permitting the user to specify two thresholds: $a_{0_{min}}$ and $a_{0_{max}}$, the minimum and maximum cumulative propensity (i.e., the total rates of stochastic transitions), respectively. Moreover, two other thresholds are required to perform dynamic partitioning: the place marking threshold and the transition rate threshold. The former is used to ensure that species concentrations are large enough to be simulated continuously, while the latter is used to partition transitions into fast and slow based on their rates. A transition is simulated continuously, if its rate exceeds the rate threshold and the marking of all its pre-place is greater than the marking threshold.

Nevertheless, cell growth has to be represented and simulated continuously in both partitioning approaches. Using off-line partitioning, this can be easily communicated to the simulator by drawing a continuous transition. However, in the case of dynamic partitioning, the transition rate threshold had to be set less than the smallest expected rate of cell growth which makes the latter approach unsuitable for the cell cycle model.



(a) (b)

Fig. 8. Example of different transition firing rates. (a) transition R_3 fires more frequently than the transition R_1, and (b) transition R_{18} fires much more often than R_{26}.

5 Simulation Results

In this section we show some simulation results of the model in Figure 6 using Snoopy's hybrid simulator. Figures 9 - 12 present time course simulation results of some model species of continuous and hybrid trajectories.

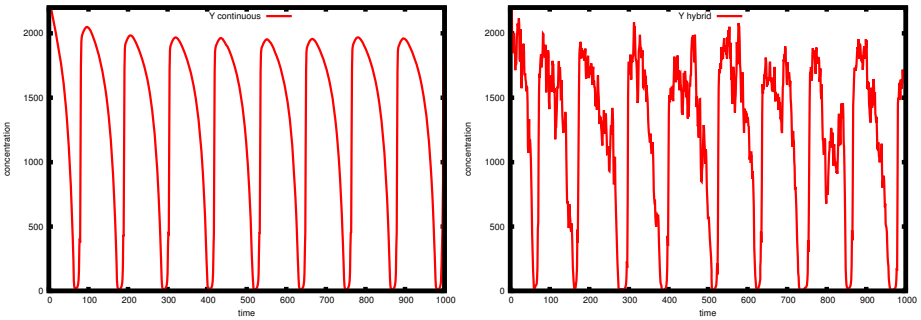
In the hybrid setting, species of low numbers of molecules are simulated using the stochastic regime, e.g., $mRNA_x$ and $mRNA_z$; thus, their numbers of molecules show variability. Such variability is due to the intrinsic noise which is captured by the stochastic simulation algorithm.

Figure 12 compares continuous and hybrid simulation results of the cellular volume (V). Using continuous simulation, parent cells divide all the time equally, and the model does not produce variability in its volume size. Contrary, hybrid simulation does show variability in the cellular volume because species of low numbers of molecules (e.g., mRNAs) are simulated stochastically.

The variability behaviour in the cellular volume, which is produced by the hybrid simulation, is close to the biological model behaviour. For example, the Fission yeast cells have at division a Coefficient of Variation (CV) of the cellular volume of about 6% [13]. The CV is a normalised measure of dispersion of a probability distribution. It is used to judge the variability of a result and it is defined as the ratio of the standard deviation σ to the mean μ, i.e, $CV = \frac{\sigma}{\mu}$.

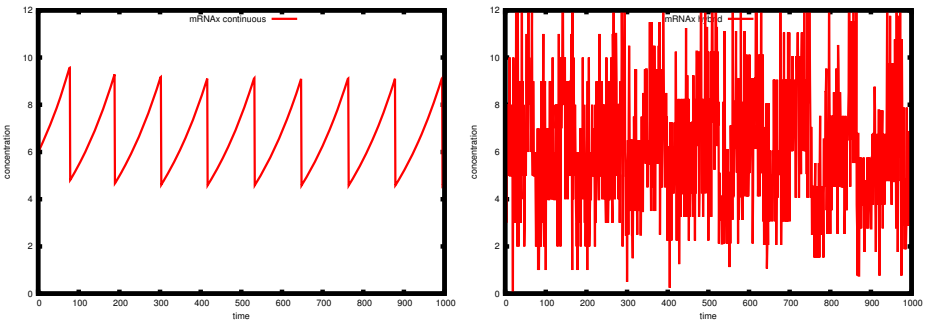
Table 1 compares the CV and mean values of the deterministic, stochastic, and hybrid simulation results as well as the experimental data of the Fission yeast (wild-type). The continuous and hybrid results are computed by exporting the Snoopy simulation output to a comma-separated values format (CSV). Then a tiny script extracts the different statistics, i.e., μ and CV.

As expected, the CVs of continuous simulation results are zero. This means that continuous simulation does not exhibit any variability in the cellular volume. Moreover, the stochastic and hybrid statistics are similar, but not the same. The variability of cellular volumes of cells simulated via the hybrid version is slightly less than the corresponding stochastic simulation. However, this is an expected behaviour since some of the transitions are continuously simulated.



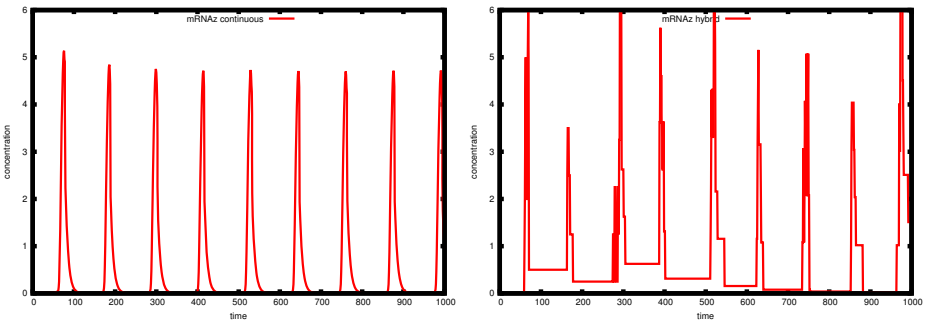
(a) (b)

Fig. 9. Time course result of Y; (a) continuous, and (b) hybrid



(a) (b)

Fig. 10. Time course result of *mRNAx*; (a) continuous, and (b) hybrid



(a) (b)

Fig. 11. Time course result of mRNAz; (a) continuos, and (b) hybrid

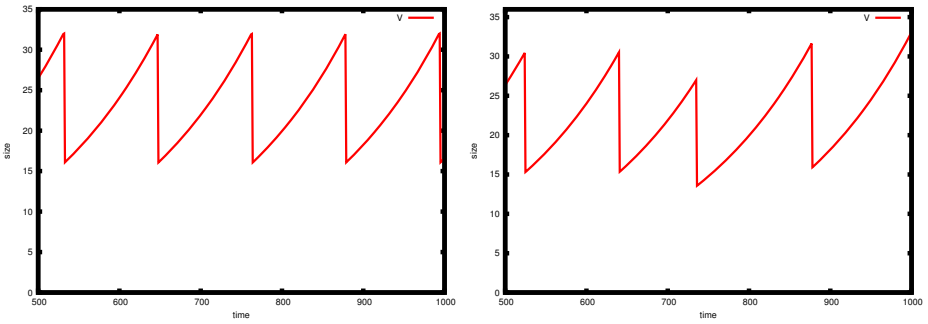


Fig. 12. Time course results of the cellular volume (V); (a) continuous, and (b) hybrid simulation

Table 1. Comparison of the continuous, stochastic, and hybrid simulation results of the model in Figure 6. The volume size is given in fl (femtolitre).

No.	simulator	cell age, min		size at division, fl		size at birth, fl		reference
		mean	CV%	mean	CV %	mean	CV%	
1	Fission yeast	148	10.8	14.4	5.9	8.2	6.3	[23]
2	deterministic	115.9	0	30.9	0	15.9	0	-
3	stochastic	115.5	13	29.1	8.2	14.5	8.2	[13]
4	hybrid	115.5	12	29.9	7.4	15	7.4	-

6 Conclusions and Outlook

In this paper we have shown a class of biological models that can appropriately be modelled using hybrid Petri nets. As an example we have presented and discussed a hybrid Petri net model of the eukaryotic cell cycle. This specific model can be executed using either continuous or hybrid simulators. It employs continuous, stochastic and immediate transitions to intuitively represent the entire model logic. Generally, depending on the type of model, a $\mathcal{GHPN}_{bio}$ model can be simulated continuously, stochastically or in a hybrid way.

The model is implemented using Snoopy. The model itself and the tool are available at `http://www-dssz.informatik.tu-cottbus.de/`. Marking-dependent arc weights are a new feature recently added to Snoopy which is currently not available in the official Snoopy release. However, the under-development version is freely available on request.

Comparing the simulation results we notice that hybrid simulation produces results close to the stochastic ones (in terms of the resulting CVs), while simulation efficiency could be preserved. Indeed, the reactions of this model can easily be split into slow and fast reactions, which makes it an ideal case study for hybrid simulation algorithms.

Marking-dependent arc weights are of paramount importance to model such biological scenarios since they provide a direct tool to program certain biological phenomenon (e.g., cell division). Therefore, we intend to add even more functionalities into this direction to permit more user-defined operators depending on the transition's pre-places, e.g., random function.

So far the partitioning of the reactions into stochastic and deterministic ones is carried out using a heuristic approach (see Section 4.3). However, (as suggested by one of the reviewers) a more sophisticated partitioning could be performed. For instance, the fast processes could be described by a quasi (or pseudo) steady state approach, assuming that they reach equilibrium rapidly. In other words, they could be better described by setting the corresponding ODEs to zero and solving them. In contrast, continuous dynamics could be seen as more appropriate for abundant molecules whose concentration display a small coefficient of variation, and stochastic dynamics for those molecules evolving at low copy numbers.

Finally, the model presented in this paper could be viewed as a sub-net in a bigger network of reactions (e.g., modelling budding yeast cell cycle or Fission yeast cells). Snoopy's hierarchical nodes might simplify such task as they provide an easy tool to insert a sub-net into a bigger one.

Acknowledgements. This work was done during the stay of Mostafa Herajy at the Brandenburg University of Technology at Cottbus, Germany, supported by the GERLS (German Egyptian Research Long Term Scholarships) program, which is administered by the DAAD in close cooperation with the MHESR and German universities.

References

1. Chen, K., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., Tyson, J.: Integrative analysis of cell cycle control in budding yeast. Mol. Biol. Cel. 5(8), 3841–3862 (2004)
2. Fujita, S., Matsui, M., Matsuno, H., Miyano, S.: Modeling and simulation of fission yeast cell cycle on hybrid functional Petri net. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E87-A(11), 2919–2927 (2004)
3. Gilbert, D., Heiner, M.: From Petri nets to differential equations - an integrative approach for biochemical network analysis. In: Donatelli, S., Thiagarajan, P.S. (eds.) ICATPN 2006. LNCS, vol. 4024, pp. 181–200. Springer, Heidelberg (2006)
4. Gillespie, D.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Comput. Phys. 22(4), 403–434 (1976)
5. Gillespie, D.: Stochastic simulation of chemical kinetics. Annual Review of Physical Chemistry 58(1), 35–55 (2007)
6. Haseltine, E., Rawlings, J.: Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. J. Chem. Phys. 117(15), 6959–6969 (2002)
7. Heiner, M., Herajy, M., Liu, F., Rohr, C., Schwarick, M.: Snoopy – a unifying Petri net tool. In: Haddad, S., Pomello, L. (eds.) PETRI NETS 2012. LNCS, vol. 7347, pp. 398–407. Springer, Heidelberg (2012)
8. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets for systems and synthetic biology. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 215–264. Springer, Heidelberg (2008)

9. Heiner, M., Lehrack, S., Gilbert, D., Marwan, W.: Extended stochastic Petri nets for model-based design of wetlab experiments. In: Priami, C., Back, R.-J., Petre, I. (eds.) Transactions on Computational Systems Biology XI. LNCS, vol. 5750, pp. 138–163. Springer, Heidelberg (2009)
10. Herajy, M., Heiner, M.: Hybrid representation and simulation of stiff biochemical networks. Nonlinear Analysis: Hybrid Systems 6(4), 942–959 (2012)
11. Herajy, M.: Computational Steering of Multi-Scale Biochemical Reaction Networks. Ph.D. thesis, Brandenburg University of Technology Cottbus - Computer Science Institute (2013)
12. Herajy, M., Heiner, M.: Hybrid representation and simulation of stiff biochemical networks through generalised hybrid Petri nets. Tech. Rep. 02/2011, Brandenburg University of Technology Cottbus, Dept. of CS (2011)
13. Kar, S., Baumann, W.T., Paul, M.R., Tyson, J.J.: Exploring the roles of noise in the eukaryotic cell cycle. Proceedings of the National Academy of Sciences of the United States of America 106(16), 6471–6476 (2009)
14. Kiehl, T., Mattheyses, R., Simmons, M.: Hybrid simulation of cellular behavior. Bioinformatics 20, 316–322 (2004)
15. Liu, Z., Pu, Y., Li, F., Shaffer, C., Hoops, S., Tyson, J., Cao, Y.: Hybrid modeling and simulation of stochastic effects on progression through the eukaryotic cell cycle. J. Chem. Phys. 136(34105) (2012)
16. 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. Methods in Molecular Biology, ch. 21, vol. 804, pp. 409–437. Humana Press (2012)
17. Matsui, M., Fujita, S., Suzuki, S., Matsuno, H., Miyano, S.: Simulated cell division processes of the xenopus cell cycle pathway by genomic object net. Journal of Integrative Bioinformatics, 0001 (2004)
18. Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., Miyano, S.: Biopathways representation and simulation on hybrid functional Petri net. Silico Biology 3(3) (2003)
19. Mura, I., Csikász-Nagy, A.: Stochastic Petri net extension of a yeast cell cycle model. Journal of Theoretical Biology 254(4), 850–860 (2008)
20. Sabouri-Ghomi, M., Ciliberto, A., Kar, S., Novak, B., Tyson, J.J.: Antagonism and bistability in protein interaction networks. Journal of Theoretical Biology 250(1), 209–218 (2008)
21. Singhania, R., Sramkoski, R.M., Jacobberger, J.W., Tyson, J.J.: A hybrid model of mammalian cell cycle regulation. PLoS Comput. Biol. 7(2), e1001077 (2011)
22. Steuer, R.: Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations. Journal of Theoretical Biology 228(3), 293–301 (2004)
23. Sveiczer, Á., Novák, B., Mitchison, J.: The size control of fission yeast revisited. J. Cell Sci. 109, 2947–2957 (1996)
24. Tyson, J., Novak, B.: Regulation of the eukaryotic cell cycle: Molecular antagonism, hysteresis, and irreversible transitions. Journal of Theoretical Biology 210(2), 249–263 (2001)
25. Tyson, J., Novak, B.: A Systems Biology View of the Cell Cycle Control Mechanisms. Elsevier, San Diego (2011)
26. Valk, R.: Self-modifying nets, a natural extension of Petri nets. In: Ausiello, G., Böhm, C. (eds.) Proceedings of the Fifth Colloquium on Automata, Languages and Programming, vol. 62, pp. 464–476. Springer, Heidelburg (1978)
27. Wikipedia: Wikipedia website (2012), `http://www.wikipedia.org/` (accessed: September 20, 2012)