Brandenburg
University of Technology
Cottbus

Faculty of Mathematics,
Natural Sciences and
Computer Science

Institute of Computer Science

COMPUTER SCIENCE REPORTS

Report 2/11

July 2011

HYBRID REPRESENTATION AND SIMULATION OF STIFF BIOCHEMICAL NETWORKS THROUGH GENERALISED HYBRID PETRI NETS

MOSTAFA HERAJY
MONIKA HEINER

Computer Science Reports
Brandenburg University of Technology Cottbus
ISSN: 1437-7969

Send requests to: BTU Cottbus
Institut für Informatik
Postfach 10 13 44
D-03013 Cottbus

Mostafa Herajy, Monika Heiner
herajmos@tu-cottbus.de, monika.heiner@informatik.tu-cottbus.de,
www-dssz.informatik.tu-cottbus.de

Hybrid Representation and Simulation of Stiff Biochemical Networks through Generalised Hybrid Petri Nets

Computer Science Reports
02/11
July 2011

Brandenburg University of Technology Cottbus

Faculty of Mathematics, Natural Sciences and Computer Science

Institute of Computer Science

Computer Science Reports
Brandenburg University of Technology Cottbus
Institute of Computer Science

Head of Institute:
Prof. Dr. Hartmut König koenig@tu-cottbus.de
BTU Cottbus
Institut für Informatik
Postfach 10 13 44
D-03013 Cottbus

Research Groups:	Headed by:
Computer Engineering	Prof. Dr. H. Th. Vierhaus
Computer Network and Communication Systems	Prof. Dr. H. König
Data Structures and Software Dependability	Prof. Dr. M. Heiner
Database and Information Systems	Prof. Dr. I. Schmitt
Programming Languages and Compiler Construction	Prof. Dr. P. Hofstedt
Software and Systems Engineering	Prof. Dr. C. Lewerentz
Theoretical Computer Science	Prof. Dr. K. Meer
Graphics Systems	PD Dr. D. Cunningham
Systems	Prof. Dr. R. Kraemer
Distributed Systems and Operating Systems	Prof. Dr. J. Nolte
Internet-Technology	Prof. Dr. G. Wagner

CR Subject Classification (1998): I.6.5, I.6.8, D.2.2, J.3

Printing and Binding: BTU Cottbus

ISSN: 1437-7969

Hybrid Representation and Simulation of Stiff Biochemical Networks through Generalised Hybrid Petri Nets

Mostafa Herajy, and Monika Heiner
Data Structures and Software Dependability,
Computer Science Department
Brandenburg University of Technology
Cottbus, Germany

Abstract

With the progress of computational modelling and simulation of biochemical networks, there is a need to manage multi-scale models, which may contain species or reactions at different scales. A visual language like Petri nets can provide a valuable tool for representing and simulating such stiff biochemical networks. In this paper we introduce a new Petri nets class, Generalised Hybrid Petri Nets ($GHPN_{bio}$) tailored to the specific needs for modelling and simulation of biochemical networks. It provides rich modelling and simulation functionalities by combining all features of Continuous Petri Nets and Generalised Stochastic Petri Nets, including three types of deterministic transitions. In this paper, we focus on modelling and simulation of stiff biochemical networks, in which some reactions are represented and simulated stochastically, while others are carried out deterministically. Two related simulation algorithms are presented, supporting static (off-line) partitioning and dynamic (on-line) partitioning. We discuss three case studies, demonstrating the use of $GHPN_{bio}$ and the efficiency of the developed simulation algorithms.

1 Introduction

Computer simulation is an essential tool for studying biochemical systems. The deterministic approach is the traditional way of simulating biochemical pathways [Pah09, WUKC04]. In this approach, reactions and their influence on the concentrations of the involved species are represented by a set of ordinary differential equations (ODEs). While this approach has the advantage of a well established mathematical basis and strong documentation, it lacks to capture the phenomena which may occur due to the underlying discreteness and random fluctuation in molecular numbers [LCPG08, Pah09], especially in situations where the number of molecules is small.

The stochastic approach [Gil76] overcomes the drawbacks of deterministic simulations and provides a very natural way of simulating biochemical pathways, since it can

$$R_1 : S_1 \xrightarrow{c_1} S_2$$
$$R_2 : S_2 \xrightarrow{c_2} S_1$$
$$R_3 : S_1 + S_2 \xrightarrow{c_3} S_3$$

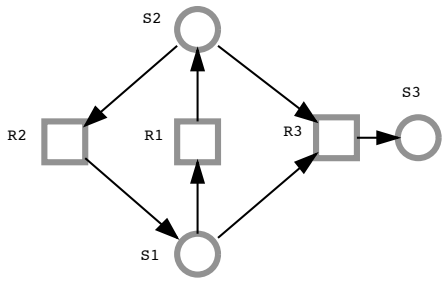

Figure 1: An example of a stiff biochemical network: Reaction set (left) [RPCG03] and its Petri nets representation (right). Assuming mass-action kinetics with $c_1 = c_2 = 10^5$ and $c_3 = 0.0005$ and the initial state $x(0) = (10000, 10000, 100)$, reaction R_3 is much slower than R_1 and R_2.

successfully capture the fluctuations of the underlying model. Furthermore it deals correctly with the problem of extremely low numbers of molecules [ACT+05, MA99]. Nevertheless, a major drawback of the stochastic simulation is that it is computationally expensive when it comes to simulate larger biological models [ACT+05, LCPG08, Pah09], especially when there are large numbers of molecules of some chemical species.

The situation becomes even more complicated for stiff models, i.e. models which combine slow and fast reactions and/or species with small and large numbers of molecules. Figure 1 shows an example for this case [RPCG03].

In this situation, neither stochastic nor deterministic simulation are appropriate to efficiently analyse it, because stochastic simulations will be very slow and the continuous ones will fail to capture the fluctuation caused by species with low copies of molecules. Hybrid simulation of biochemical networks has been previously studied in, e.g., [ACT+05, GCPS06, HR02, KMS04, SK05]. To overcome the problem of stiffness, sets of reactions are divided into two subsets: slow and fast. The slow set is simulated stochastically, while the fast one is simulated deterministically – either using the ODE system or stochastically using the chemical Langevin equation.

On the other side, the tight analogy between Petri nets and biochemical reactions makes them a natural choice to model chemical reaction networks [HGD08, RML93]. Being bipartite and inherently concurrent are common properties shared by Petri nets and biochemical reaction networks. Qualitative Petri net [HGD08] can be used to analyse the biochemical systems qualitatively, while stochastic and continuous Petri nets are used to simulate them quantitatively.

Continuous Petri Nets provide a way for modelling systems in which states change continuously over time. In systems biology, biochemically interpreted continuous Petri nets (CPN_{bio}) provide a convenient means of describing ODEs in a structure-oriented manner. Pre-places of the transitions represent reactants and the marking of places stands for the species' concentrations. Each transition t is associated with a rate function $v(t)$ which defines the kinetic rate. The corresponding ODE which represents the change

of the concentration of the species p is generated by (1), see e.g. [GH06],

$$\frac{dp}{dt} = \sum_{t \in {}^\bullet p} f(t,p)v(t) - \sum_{t \in p^\bullet} f(p,t)v(t) \tag{1}$$

where $f(t,p)$ is the arc weight connecting transition t with place p, likewise $f(p,t)$, and ${}^\bullet p, p^\bullet$ are the pre- and post-transitions of place p, respectively. Note that place names are here read as real-valued variables.

In contrast to continuous Petri nets, stochastic Petri nets preserve the discrete state description as it is the standard in Petri nets. Biochemical models are simulated stochastically by associating a probability-distributed firing rate (waiting time) with each transition. To extend the modelling capabilities of stochastic Petri nets (SPN) for the appropriate modelling of biological system and experimental conditions in the wet-lab, two extensions of SPN have been introduced – biochemically interpreted Generalised Stochastic Petri Nets ($GSPN_{bio}$), and Extended Stochastic Petri Nets ($XSPN_{bio}$) [HLGM09]. The extensions include inhibitor and read arcs and deterministically time-delayed transitions.

In this paper, we introduce a new class of Petri nets which combines the power of CPN_{bio} and $XSPN_{bio}$ – Generalised Hybrid Petri Nets ($GHPN_{bio}$). They are particularly well suited to represent and simulate stiff biochemical networks. The modelling power of this class of Petri nets allows the combination of both discrete and continuous network parts in one model, which permits to represent, e.g., a biological switch in which continuous elements are turned on/off by discrete elements. The models can be simulated using static partitioning in which the partitioning is done off-line before the simulation starts, or using dynamic partitioning in which the partitioning is done on-line during the simulation.

This paper is organized as follows: we start off with recalling some related work, specifically Petri nets classes which have been recently used in the context of biochemical modelling. Afterwards the theoretical background for hybrid simulation of stiff biochemical networks is presented. Then, we introduce $GHPN_{bio}$ and show how they can be simulated using static or dynamic partitioning. Three examples demonstrate how stiff biochemical networks can be conveniently modelled and efficiently be simulated using $GHPN_{bio}$. We conclude with a brief discussion and conclusions.

2 Related Work

Hybrid Petri nets [AD98] incorporate both continuous and discrete capabilities and can be used to model systems which contain both discrete and continuous elements. Many variations of hybrid Petri nets have been introduced during the last two decades, with different modelling goals.

Hybrid Dynamic Nets (HDN) [Dra98] allow any function for defining state-dependent transition rates, without structural restrictions. In our net class, we restrict the domain of rate functions to the transitions' pre-places. This constraint is very useful in the biological context and crucial for the efficiency of our tools, since the reactions' propensities

(i.e the transition's rates) are calculated in terms of the reactions' reactants. We provide a special arc type called modifier to allow any place in the transitions' rate functions.

Hybrid Functional Petri Nets (HFPN) were introduced in [MTA+03] to allow any function to be assigned as input/output weight or as transition delay. Hybrid Functional Petri Nets with extension (HFPNe) [NDMM04] extend HFPN by generic entities and generic data types. However, dynamic partitioning of transitions into discrete and continuous ones is not considered. In [YLL09], transitions can be simulated in an adaptive way, but distinction between discrete and continuous places is not supported. Other transition types (immediate transitions) are not supported neither.

Contrary, Fluid Stochastic Petri Nets (FSPNs) [TK93] combine both stochastic and continuous net parts into one net class. However, they suffer from unclear and inconsistent graphical representations [HK99] which make them inappropriate for our purpose of representing and simulating biochemical networks. More importantly, they do not support the full range of deterministically delayed transitions as we do.

Hybrid simulation of biochemical networks using a combination of both stochastic and deterministic reactions was introduced in [HR02], however time-dependent propensities are not considered, and the reactions are partitioned statically. In [ACT+05, GCPS06], time-dependent propensities are considered when determining the reaction type and the next time, at which a stochastic reaction will occur. A survey of different hybrid simulation methods can be found in [Pah09]. In our paper we consider time-dependent propensities when locating stochastic events as well as other event types like firing of immediate, deterministic, or scheduled transitions.

In [ACT+05, GCPS06, YLL09], reactions are partitioned dynamically based on two thresholds: one for transitions and the other one for places. In our approach we are going to use two additional thresholds to decide the repartitioning time. This will answer the question of when we need to reconsider repartitioning of the biochemical networks. Unlike most of the previous works of studying multi-scale biochemical networks which concentrated on the simulation aspect only [ACT+05, HR02, GCPS06, WGMH10, HMMW10], we pay attention to the representation aspect as well.

The specific contribution of our paper is the provision of a new class of Petri nets, Generalised Hybrid Petri Nets, to support stiff biochemical networks both on representation and simulation level. The models can be simulated using both static and dynamic partitioning. Additionally, $GHPN_{bio}$ inherits from Snoopy [RMH10] – a tool to design and animate or simulate hierarchical graphs, among them qualitative, stochastic, continuous and hybrid Petri nets – some very useful features which do not exist in any of the previously mentioned Petri nets classes like logical nodes and hierarchies which are crucial means for modelling large scale biochemical networks.

3 Theory

Consider a well mixed system of N chemical species $S_1, \ldots, S_N$, which interact using M chemical reactions $R_1, \ldots, R_M$. The state of the system at any time t, can be represented by an N-vector $\mathbf{X}(t) = \mathbf{X}_1(t), \ldots, \mathbf{X}_N(t)$, where $\mathbf{X}_i(t)$ gives the number of molecules of

specie S_i at time t. The goal is to find an estimated evolution of the vector $\mathbf{X}$ over the time t, starting from an initial state $\mathbf{X}(t_0)$ [Gil07]. In the following we often refer to a reaction R_i by just giving its index i, as it is habit in related literature.

If the thermodynamic limit condition holds (i.e. the number of molecules and the volume of the system approach infinity), then the evolution of the above system can be represented as a set of ordinary differential equations(ODEs) [HR02, WUKC04] in the form of (2), where the concentration of species S_i is denoted by $[S_i]$. Please note that equation (1) – which is used to generate the ODEs from Petri nets – has also the same form as (2), where the function $v(t)$ is a state dependant transition rate.

$$\frac{d[S_i]}{dt} = f_i([S_1], \ldots, [S_N]) \tag{2}$$

However if the system contains some species with low numbers of molecules, then the thermodynamic limit condition will be violated and deterministic simulation will not reflect the actual model behaviour [Gil76]. In this case, stochastic simulation can be used to simulate the model at the molecular level which takes into account the inherently discrete and stochastic nature of chemical reactions [MA99].

Gillespie [Gil76, Gil77] derived two Monte Carlo based simulation algorithms to simulate Markov systems. The direct and first reaction methods are two variations to simulate a set of coupled reactions.

According to the direct method [Gil76, Gil77], the next time τ a reaction will occur is specified by

$$\tau = -\frac{1}{a_0(\mathbf{x})} \ln r_1 \,, \tag{3}$$

and the reaction, R_μ to occur is determined by

$$\sum_{i=1}^{\mu-1} a_i(\mathbf{x}) < r_2 a_0(\mathbf{x}) \leq \sum_{i=1}^{\mu} a_i(\mathbf{x}) \,, \tag{4}$$

where r_1 and r_2 are two random numbers which are generated from a uniform distribution $(0, 1)$, $a_i(\mathbf{x})$ is the propensity of reaction R_i at a state $\mathbf{X}(t) = \mathbf{x}$, and $a_0(\mathbf{x}) = \sum_{i=1}^{M} a_i(\mathbf{x})$ is the total (cumulative) propensity [Pah09].

However, neither stochastic nor deterministic approaches are appropriate to simulate stiff models due to the aforesaid reasons. One choice in such a situation is hybrid simulation. In hybrid simulation, reactions are divided into two groups: fast and slow. In the former case the macroscopic conditions are fulfilled and they can be simulated using an ODE system, while in the latter case the set of slow reactions are simulated using stochastic methods [HR02, SK05, Gou05, GCPS06]. Figure 2 summarizes the relationship between stochastic, deterministic and hybrid simulation.

Due to the combination of both deterministic and stochastic reactions in the hybrid simulation approach, the propensities of the stochastic reactions depend on the state

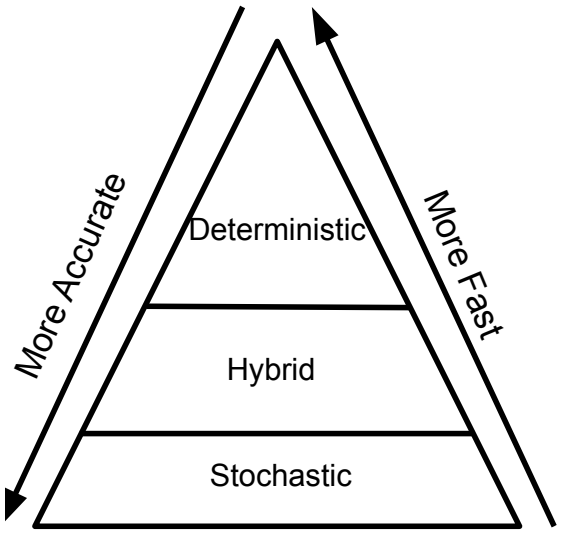

Figure 2: The relationship between hybrid simulation and other methods of simulating biochemical networks.

change of deterministically simulated reactions [HR02, SK05, Pah09]. Gillespie [Gil91] derived the correct reaction probability density function for this case as

$$P(\tau, \mu|\mathbf{X}(t), t) = a_\mu(\mathbf{X}(t+\tau)) \exp\left(-\int_t^{t+\tau} a_0(\mathbf{X}(t))dt\right), \tag{5}$$

In [HR02], fast reactions are represented by a continuous Markov process being coupled to Markov jump process for slow reactions where the CTMC is approximated by ODEs. However, they do not consider time varying propensities for slow reactions; instead a probability is introduced that no reaction occurs to decrease the approximation error [Pah09]. Other hybrid methods, for example in [ACT$^+$05, GCPS06], consider time-varying propensities of slow reactions using (6).

$$g(\mathbf{x}) = \int_t^{t+\tau} a_0^s(\mathbf{x})dt - \xi = 0\,, \tag{6}$$

where ξ is a random number exponentially distributed with a unit mean, and $a_0^s(\mathbf{x})$ is the cumulative propensity of slow reactions.

Using (6), the hybrid simulation algorithm can switch between deterministic and stochastic simulation by integrating the set of ODEs representing fast reactions along with the cumulative propensity, $a_0^s(\mathbf{x})$, till (6) is satisfied, which means that a stochastic event has occurred. Then, a stochastic reaction R_μ is selected such that

$$\sum_{i=1}^{\mu-1} a_i^s(\mathbf{x}) < r_2 a_0^s(\mathbf{x}) \leq \sum_{i=1}^{\mu} a_i^s(\mathbf{x})\,, \tag{7}$$

where $a_i^s(\mathbf{x})$ is the propensity of the i^{th} slow reaction, and r_2 is a random number uniformly distributed in (0,1).

In simulating $GHPN_{bio}$'s models, we not only have to detect stochastic events, but also other event types such as immediate and deterministic events. Immediate events represent a firing of an immediate transition while deterministic events represent a firing of a deterministically delayed time transition. The detailed algorithm is presented in 4.3.

4 Generalised Hybrid Petri Nets

In this part, we discuss in more detail the different aspects of the Generalised Hybrid Petri Nets class. We start by its modelling capabilities for biological systems, specifically in simulating stiff biochemical networks, and explain how $GHPN_{bio}$ models can be simulated.

4.1 Modelling

To model stiff biochemical networks, Generalised Hybrid Petri Nets ($GHPN_{bio}$) combine both stochastic and continuous elements in one and the same model. Indeed, continuous and stochastic Petri nets complement each other. The fluctuation and discreteness can be powerfully modelled using stochastic simulation and at the same time, the computationally expensive parts can be simulated deterministically using ODE solvers. Modelling and simulation of stiff biochemical networks is one of the outstanding functionalities that GHPNs can provide in the area of systems biology.

Generally speaking, biochemical systems can involve reactions from more than one type of biological networks, for example regulatory, metabolic or transduction pathways. Incorporating reactions which belong to distinct (biological) networks, tends to result into stiff systems. This follows from the fact that regulatory networks' species may contain a few number of molecules, while metabolic networks' species may contain a large number of molecules [KMS04].

In the rest of this section, we will discuss in more details the newly introduced net class in terms of the graphical representation of its elements as well as the firing rules and connectivity between the continuous and stochastic net parts.

4.1.1 Graphical Representation

$GHPN_{bio}$ contains two types of places: discrete and continuous. Discrete places (single line circle) contain non-negative integer numbers which represent the number of molecules in a given specie. On the other hand, continuous places - which are represented by shaded line circle - contain non-negative real numbers which represent the concentration of a certain species. Furthermore $GHPN_{bio}$ contains five transition types: continuous, stochastic, deterministic, immediate, and scheduled transitions [HGD08]. Continuous transitions (shaded line square) fire continuously in the same way like in

continuous Petri nets. Their semantics are governed by ordinary differential equations. Their ODEs define the changes in the transitions' pre- and post-places.

Stochastic transitions which are drawn in Snoopy as a square, fire randomly with an exponentially distributed random delay. The user can specify a set of firing rate functions, which determine the random firing delay. The transition's pre-places can be used to define the firing rate functions of the stochastic transitions. Deterministic transitions (represented as black squares) fire after a specified constant time delay, immediate transitions (black bar) fire with zero delay, and have always higher priority in the case of conflicts with other transitions. They may carry weights which specify the relative firing frequency in the case of conflicts between more than one immediate transition. Scheduled transitions (grey squares) fire at user-specified absolute time points. More details about the biochemical interpretation of deterministic, scheduled, and immediate transitions can be found in [HLGM09].

The connection between those two types of nodes (places and transitions) takes place using a set of different arcs. $GHPN_{bio}$ contains six types of arcs: standard, inhibitor, read, equal, reset and modifier arcs. Standard arcs connect transitions with places or vice versa. They can be continuous, i.e carry non-negative real-valued weights (stoichiometry in the biochemical context), or discrete i.e carry non-negative integer-valued weights. Special arcs like inhibitor, read, equal and reset arcs can only be used to connect places to transitions, but not vice versa. Further arc types – like modifier and equal arcs – simplify the modelling task. For example, a modifier arc permits to include any place in the transitions' rate functions and simultaneously preserves the net structure restriction. The connection rules and their underlying semantics are discussed in more details below. Figure 3 provides a graphical illustration of those elements. Although this graphical notation is the default one, they can be easily customised using our Petri nets editing tool, Snoopy.

As a simple example for the above discussion, consider again the stiff biochemical network in Figure 1. Using $GHPN_{bio}$, the slow reaction R_3 can be modelled using a stochastic transition while the other two fast reactions, R_1 and R_2, can be represented using continuous transitions. Places are partitioned into discrete and continuous based on the connection rules which will be discussed next.

4.1.2 Connection Rules

A critical question arises when considering the combination of discrete and continuous elements: how are these two different parts connected with each other? Figure 4, provides a graphical illustration of how the connection between different elements of the introduced $GHPN_{bio}$ takes place. We denote here by discrete transitions: stochastic, immediate, deterministic or scheduled transitions.

Firstly, we will consider the connection between continuous transitions and the other elements of $GHPN_{bio}$. Continuous transitions can be connected with continuous places in both directions using continuous arcs (i.e arc 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.

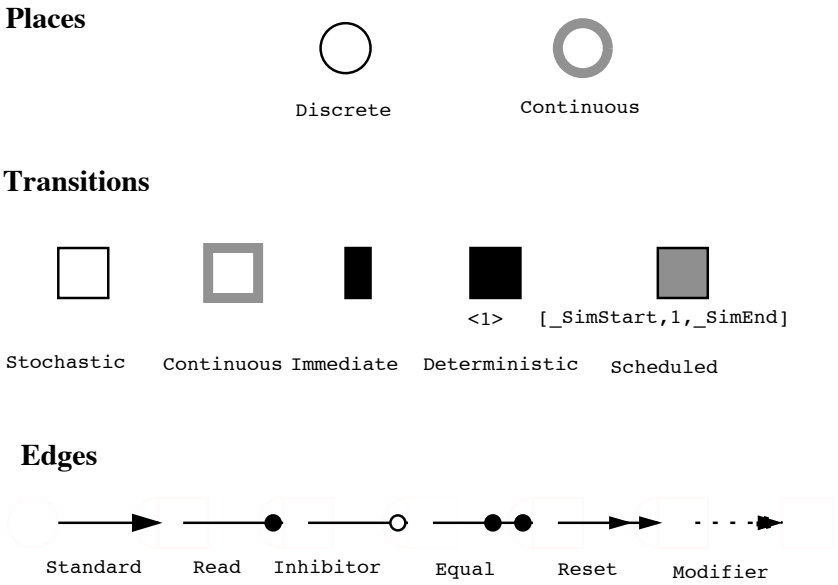

Figure 3: Graphical representation of the $GHPN_{bio}$'s elements. Places are classified as continuous and discrete, transitions as continuous, stochastic, immediate, deterministic, and scheduled, and arcs as standard, inhibitor, read, equal, and modifier.

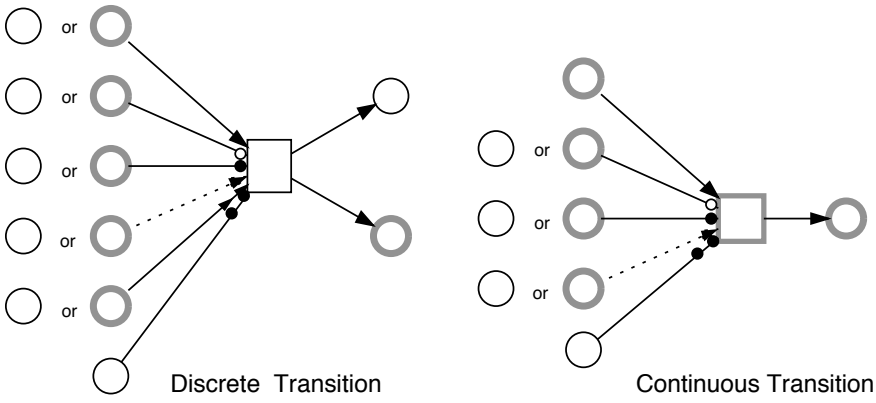

Figure 4: Possible connections between $GHPN_{bio}$'s elements: Discrete places can not be connected with continuous transitions using standard arcs.

Continuous transition can also be connected with discrete places, but only by one of the extended arcs (inhibitor, read, equal, and modifier). Read arcs allow to specify positive side conditions, while inhibitor arcs allow to specify negative side conditions. It is worth being mentioned that the markings of the transition's pre-places connected by these special arcs do not change when the transition fires. This type of connection allows a connection between discrete and continuous parts of the 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 can not take place in the discrete world. Discrete transitions (stochastic, deterministic, immediate and scheduled ones) can be connected with discrete or continuous places in both directions using standard arcs. However, the arc's weight should be considered, i.e 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 the type of the transition's pre/post-places.

Connecting continuous places and discrete transitions will result in a model like in [TK93], in which changes in continuous places are governed by firing of stochastic transitions. Discrete transitions can also have discrete or continuous pre-places using special arcs.

4.2 Formal Definition

In a more formal way, Generalised Hybrid Petri Nets are a 5-Tuple, $GHPN_{bio} = [P, T, A, V, m_0]$ where: P, T are finite, non-empty and disjoint sets. P is the set of places, and T is the set of transitions with:

- $P = P_{cont} \cup P_{disc}$ whereby P_{cont} is the set of continuous places to which non-negative real values are assigned, and P_{disc} is the set of discrete places to which non-negative integer values are assigned.
- $T = T_{cont} \cup T_{stoch} \cup T_{im} \cup T_{timed} \cup T_{scheduled}$ with:
  1. T_{cont} is the set of continuous transitions, which fire continuously over time.
  2. T_{stoch} is the set of stochastic transitions, which fire stochastically with exponentially distributed waiting time.
  3. T_{timed} is the set of deterministic transitions, which fire with a deterministic time delay.
  4. $T_{scheduled}$ is the set of scheduled transitions, which fire at predefined firing time points.
  5. T_{im} is the set of immediate transitions, which fire with waiting time zero; they have higher priority compared with other transitions.

- $A = A_{cont} \cup A_{disc} \cup A_{inhibit} \cup A_{read} \cup A_{equal} \cup A_{reset}$ $\cup A_{modifier}$ is the set of directed edges, whereby:

  1. $A_{cont} : ((P_{cont} \times T) \cup (T \times P_{cont})) \rightarrow \mathbb{R}_0^+$ defines the set of continuous, directed arcs, weighted by non-negative real values.
  2. $A_{disc} : ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}_0$ defines the set of discrete, directed arcs, weighted by non-negative integer values.
  3. $A_{read} : (P \times T) \rightarrow \mathbb{R}_0^+$ if $P \in P_{cont}$, or $A_{read} : (P \times T) \rightarrow \mathbb{N}_0$ if $P \in P_{disc}$ defines the set of read arcs.
  4. $A_{equal} : (P_{disc} \times T) \rightarrow \mathbb{N}_0^+$, defines the set of equal arcs.
  5. $A_{inhibit} : (P \times T) \rightarrow \mathbb{R}_0^+$ if $P \in P_{cont}$, or $A_{inhibit} : (P \times T) \rightarrow \mathbb{N}_0$ if $P \in P_{disc}$ defines the set of inhibits arcs.
  6. $A_{reset} : (P \times T_{discrete}) \rightarrow \{0, 1\}$ defines the set of reset arcs, where $T_{discrete} = T_{stoch} \cup T_{im} \cup T_{timed} \cup T_{scheduled}$ is the set of discrete transitions.
  7. $A_{modifier} : (P \times T) \rightarrow \{0, 1\}$ defines the set of modifier arcs.

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

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

- $m_0 = m_{cont} \cup m_{disc}$ 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, $\mathbb{N}_0$ denotes the set of non-negative integer numbers, $\mathbb{R}_0^+$ denotes the set of non-negative real numbers, and $^\bullet t$ denotes the pre-places of a transition t.

The formal definition above covers the syntactical aspects of Generalised Hybrid Petri Nets. Their operational semantics will be formally defined in the next section, when we discuss their simulation.

4.3 Simulation of GHPN

After the modelling aspects of $GHPN_{bio}$ have been presented in the previous section, we discuss here the approach which is used to simulate $GHPN_{bio}$. The key idea behind simulation of Generalised Hybrid Petri Nets is to numerically solve the set of ordinary differential equations generated by the continuous transitions until a discrete event occurs. The event type is dispatched, and afterwards the continuous simulation is resumed. We start by discussing the simulation of statically partitioned $GHPN_{bio}$, then the dynamic partitioning of $GHPN_{bio}$ is presented, which substantially simplifies the modelling of biochemical networks using $GHPN_{bio}$.

4.3.1 Simulation of Statically Partitioned $GHPN_{bio}$

In the following we illustrate how $GHPN_{bio}$ can be simulated using an extended version of the algorithms which are discussed in [ACT$^+$05, GCPS06, HR02]. Algorithm (1) summarizes the steps which are needed to simulate Generalised Hybrid Petri Nets. Starting from an initial marking which corresponds to the initial state of a biochemical system, the algorithm computes state changes over time t, which is represented by the current marking $m(CurrentTime)$. Initially the current marking is set to the initial marking, and the individual propensities $a(t)$ as well as the cumulative propensity are calculated for both stochastic and continuous transitions (lines 3-5). Deterministically time delayed transitions do not have propensities, since they fire after a pre-defined time delay, likewise for scheduled transitions. Note that in the algorithm we consider scheduled transitions as deterministically time delayed transitions since they can be considered as a special case of them. Note that immediate transitions also do not have propensities associated with them (see Section 4.1.1).

If there is a non-stochastic transition in the underlying model, then the algorithm determines the next stochastic transition to fire by integrating the set of ODEs as well as the cumulative propensities until equation (6) is satisfied.

The numerical integrator stops when an event E_i occurs. The event may be an enabling of an immediate or deterministically time delayed transition, a deterministically time delayed transition has finished its delay, a stochastic event occurred, or the end of simulation time has been reached. Then, the appropriate action will be taken.

Line 11 updates all of the transitions' propensities that share a pre-place with a continuous transition. The function $IsEnabled(t)$ checks for enabling of a transition t, while $Fire(t)$ fires an enabled transition. The details of these functions are easy to be implemented, therefore they are not further considered here.

$CheckImmediateTransitions()$ checks if there is any immediate transition enabled. If such a transition is found, it will be fired. If there are several immediate transitions enabled then the first one to fire is selected based on their weights. More precisely, if an immediate transition k is enabled in the current marking m, then it fires with probability:

Algorithm 1 Simulating Static Partitioned $GHPN_{bio}$

1: $CurrentTime \leftarrow 0$;
2: $\xi \leftarrow exp(1)$ {Generate a random number exponentially distributed with a unit mean}
3: $m(CurrentTime) \leftarrow m(0)$; {current marking=initial marking}
4: $\forall t \in \{T_{cont} \cup T_{stoch}\}$ calculate $a(t)$;
5: $a_0 \leftarrow \sum a(t), \forall t \in T_{stoch}$;
6: **while** CurrentTime $< EndTime$ **do**
7: **if** There are non-stochastic transitions **then**
8: Initialize the ODE solver by $m(CurrentTime)$;
9: Simultaneously integrate the system of ODEs generated using (1) and $g(m(CurrentTime))$ until an event E occurs;
10: $CurrentTime$ = the current integrator time;
11: Update$(a(t_i), a_0)$, $\forall t_i : {}^\bullet t_i \cap \{{}^\bullet t_j \cup t_j^\bullet\} \neq \phi, \forall t_j \in T_{cont}$;
12: **if** E is: $\exists t \in T_{im}$ and IsEnabled(t) **then**
13: CheckImmediateTransitions();
14: **else if** E is: $\exists t \in T_{deter}$ and IsEnabled(t) **then**
15: CheckDeterministicTransitions();
16: **else if** E is: $\exists t \in T_{deter}$ and FireTime(t)= $CurrentTime$ **then**
17: CheckDeterministicTransitions();
18: **else if** E is: $g(m(CurrentTime)) \geq 0$ **then**
19: $g(m(CurrentTime)) \leftarrow 0$;
20: $\xi \leftarrow exp(1)$;
21: $t_{chosen} \leftarrow$ a transition index i satisfying (7);
22: Fire(t_{chosen});
23: Update$(a(t_i))$, $\forall t_i : {}^\bullet t_i \cap \{t_{chosen}^\bullet \cup {}^\bullet t_{chosen}\} \neq \phi$
24: **else if** E is: $CurrentTime \geq EndTime$ **then**
25: break;
26: **end if**
27: **else**
28: $CurrentTime = CurrentTime$+exp$(a_0)$ {See (3).}
29: **if** $CurrentTime < EndTime$ AND $a_0 > 0$ **then**
30: $t_{chosen} \leftarrow$ a transition index i satisfying (4);
31: Fire(t_{chosen});
32: Update$(a(t_i))$, $\forall t_i : {}^\bullet t_i \cap \{t_{chosen}^\bullet \cup {}^\bullet t_{chosen}\} \neq \phi$
33: **end if**
34: **end if**
35: **end while**

$$\frac{w(k,m)}{\sum\limits_{i \in T_{im} \wedge isEnabled(i)} w(i,m)}, \tag{8}$$

where $w(k, m)$ is the weight assigned to an immediate transition k in the current marking m. It is worth mentioning here that immediate transitions have priority over all other transitions in the case of conflict. The purpose of $CheckDeterministicTransitions()$ is twofold. Firstly, it checks if there are any enabled deterministic transitions; it puts them in the delay list along with their proposed time to fire. If there are transitions in the delay list which have finished their delay, then it fires them.

Lines 19-23 and lines 30-32 perform the same task, but for different conditions. In the former case, a stochastic transition is selected to be fired when the ODE integrator determines that a stochastic event has occurred. The stochastic transition is selected based on equation (7). While in the latter case, the model contains only stochastic transitions. Thus the next reaction time is computed based on equation (3), and the next transition to fire is selected based on (4).

When a transition fires, the propensity of this transition as well as of any other transitions that are affected by this firing are computed and the cumulative propensity is updated. The simulation ends when the current simulation time exceeds the simulation's end time which is specified by the user.

While this algorithm can simulate any $GHPN_{bio}$, it requires the user to specify the partitioning in advance. Sometimes it is not easy for naive users to do the partitioning off-line. It is also possible that a good partitioning changes dynamically over time. Therefore, we present in the next section an algorithm which supports on-line partitioning. In some cases, the cost of this dynamic partitioning is more computational overhead [Pah09].

4.3.2 Transition Partitioning

Static partitioning of Petri nets into stochastic and continuous net parts is not always appropriate. During simulation the transitions' rates can drastically vary between low and high. Furthermore off-line partitioning is not user friendly, since it is not easy for naive users to determine which transitions should be considered stochastically and which one continuously [Pah09]. The latter problem could be overcome by running stochastic simulation for only one trajectory in order to determine partitioning automatically [ACT+05]. Another solution is to use stochastic analysis techniques.

The partitioning of the reactions into slow and fast can be done through the use of two thresholds: one for the transitions' rates and the other one for the places' marking [ACT+05, GCPS06].

However one important question remains: when do we need to consider repartitioning? One solution to this problem is to reconsider repartitioning after a specific time period (for example every one or two seconds). However this will not correctly solve the problem since during this period there may be changes in some species' populations. Moreover it results in computational overhead when there is no need to repartition. In our partitioning approach we solve this problem by specifying two other thresholds: $a_{0_{max}}$, $a_{0_{min}}$.

Consider equation (3), which determines the next time point a stochastic event will occur. Larger values of a_0 will result in smaller time steps in stochastic simulation. On

the other hand, smaller values of a_0 will keep the time step small. In fact this also affects equation (6) which determines when we switch from deterministic to stochastic simulation. The same arguments holds for (6). The main idea here is that we can control the speed and accuracy of the hybrid simulation by specifying a lower and upper bound of a_0. Then the algorithm will realize that it needs to repartition the net when a_0 drops below $a_{0_{min}}$ or exceeds $a_{0_{max}}$.

Algorithm (2) summarizes the steps which are needed to carry out on-line partitioning of the network. It considers repartitioning if equation (9), (10) or both are violated.

$$a_{0_{min}} \leq a_0 \leq a_{0_{max}} \tag{9}$$

$$\#p \geq \Lambda, \forall p \in \{{}^\bullet t \cup t^\bullet\} \forall t \in T_{cont} \tag{10}$$

where Λ is a threshold for the number of tokens in a place p.

An inappropriate choice of the thresholds can result in unsuitable partitioning which may turn out to be more computational expensive than static partitioning.

The algorithm takes as inputs the stochastic and continuous transitions, a_{min}, a_{max} – the upper and lower bounds of the cumulative propensity, respectively, the transitions' rate threshold λ, and the places' marking threshold Λ. Note that the other transition types are not repartitioned. At the end of the partitioning the algorithm returns T'_{stoch} and T'_{cont} as the new partitioning.

The idea of repartitioning is then very easy. If one of the transitions violates the partitioning criterion, it will be added to the stochastic transitions, otherwise it will be added to the continuous one.

This algorithmic idea, together with the one which is presented in Section 4.3.1, provide a dynamic simulation of the Petri nets' elements which have been introduced in this paper.

Algorithm 2 Dynamic Partitioning of $GHPN_{bio}$

Input: a_{min}, a_{max}, λ, Λ, T_{stoch}, T_{cont}
Output: T'_{stoch}, T'_{cont}

if $a_0 < a_{min}$ OR $a_0 > a_{max}$ OR $\#p < \Lambda \ \forall p \in {}^\bullet T_{cont} \cup T_{cont}^\bullet$ **then**
  for all $t \in T_{stoch} \cup T_{cont}$ **do**
    if $a_t > \lambda$ AND $\forall p \in \{{}^\bullet t \cup t^\bullet\}, \#p > \Lambda$ **then**
      if $t \in T_{stoch}$ **then**
        $a_0 \leftarrow a_0 - a(t)$
        $T'_{cont} \leftarrow T'_{cont} \cup \{t\}$
        $T_{stoch} \leftarrow T_{stoch} - \{t\}$
      end if
    else
      if $t \in T_{cont}$ **then**
        $a_0 \leftarrow a_0 + a(t)$
        $T_{cont} \leftarrow T_{cont} - \{t\}$
        $T'_{stoch} \leftarrow T'_{stoch} \cup \{t\}$
      end if
    end if
  end for
  return T'_{stoch}, T'_{cont}
else
  return T_{stoch}, T_{cont} {No partitioning is needed}
end if

4.4 Implementation

The presented Petri nets class and its simulation are implemented in Snoopy [RMH10]. Snoopy is available free of charge for non-commercial use. It is platform independent and runs under Mac OS X, Windows and Linux (selected distributions). We implemented Gillespie's direct method [Gil76] to simulate stochastic transitions, while SUNDIALS CVODE [HBG+05] is used to integrate the resulting ODEs due to continuous transitions.

Snoopy supports also a dedicated net class to simulate Petri nets which contain only continuous elements and provides 14 different ODEs integrators. The addition of further stochastic simulators is easy due to the generic design of Snoopy (see future work). Snoopy supports also many other useful modelling features like hierarchy and logical nodes which are very useful tools when considering modelling and simulation of large scale biochemical networks. Furthermore, a model developed with Snoopy can be exported to a variety of analysis tools. Additionally, using $GHPN_{bio}$ the same model can be simulated continuously or stochastically independently of its original modelling method, thanks to dynamic partitioning.

5 Case Studies

In this section, we present three case studies to illustrate hybrid modelling and simulation of $GHPN_{bio}$: Goutsias model, Circadian oscillation model, and T7 Phage model. Using the first example we aim to demonstrate the speed up of the computation while preserving accuracy. In the second and third examples we use models where stochasticity plays a role when there are a few number of molecules. Stochasticity can also be preserved when GHPN is used.

5.1 Goutsias Model

This model has been used by Goutsias in [Gou05] as an example for systems that can be effectively partitioned into two distinct subsystems, one that comprises slow reactions and one that comprises fast reactions. It has been studied in [WGMH10] and [HMMW10] as example for hybrid numerical solutions of the chemical master equation. We use the same reactions which have been originally proposed by [Gou05], and the more challenging parameters which have been used in [HMMW10].

Figure 5 is a hybrid Petri net representation of Goutsias' model. The partitioning of transitions and places into discrete and continuous ones is based on running one trajectory of a fully stochastic simulation. R_1, R_3, R_9, and R_{10} are reactions with high rates compared to the other reactions. Thus this set of reactions is represented by continuous transitions which in turn are simulated by ODEs integrator. Note that places are partitioned into continuous and discrete ones according to the type of pre- and post-transitions. Places are considered as discrete ones if the adjacent arcs do not preclude this interpretation. Figure 6 is a time course result of the places DNA, DNA.D, and DNA.2D. The hybrid simulation result coincides with the stochastic one for the three species.

5.2 Circadian Oscillation

In some organisms, there is a control mechanism which is responsible for ensuring a periodic oscillation of certain molecular species [HL07]. This phenomenon is known as circadian rhythm and it can be found in many organisms (e.g Drosophila).

In this case study, we consider a simple model which demonstrates this phenomenon. The model consists of two genes which are represented in Figure 7 by two places G_1 and G_2. The model includes also one activator and one repressor which are represented by the places A and R, respectively. The activator and repressor control the two genes and their mRNAs, mRNA_G_1 and mRNA_G_2. A and R can be activated to form a complex A_R which takes place through reaction R_{12}.

The Petri net in Figure 7 contains 9 places and 12 transitions. Note that places with the same name are logical places. This feature is used to simplify the graphical representation of the Petri net. The complete list of reactions as well as the parameter list can be found in [HL07]. Figure 8 gives a time course simulation result of the $GHPN_{bio}$ in Figure 7. Using the parameter values given in Figure 8, continuous simulation produces

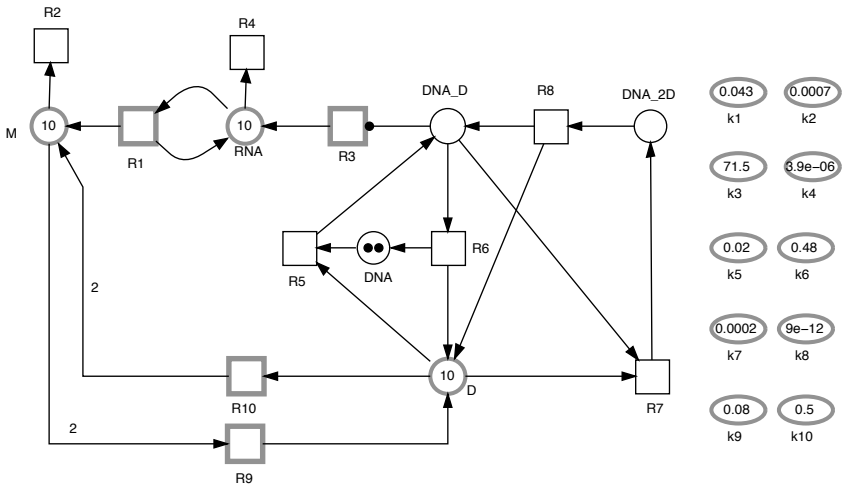

Figure 5: A $GHPN_{bio}$ representation of the Goutsias model. Numbers in ovals represent reaction rate constants.

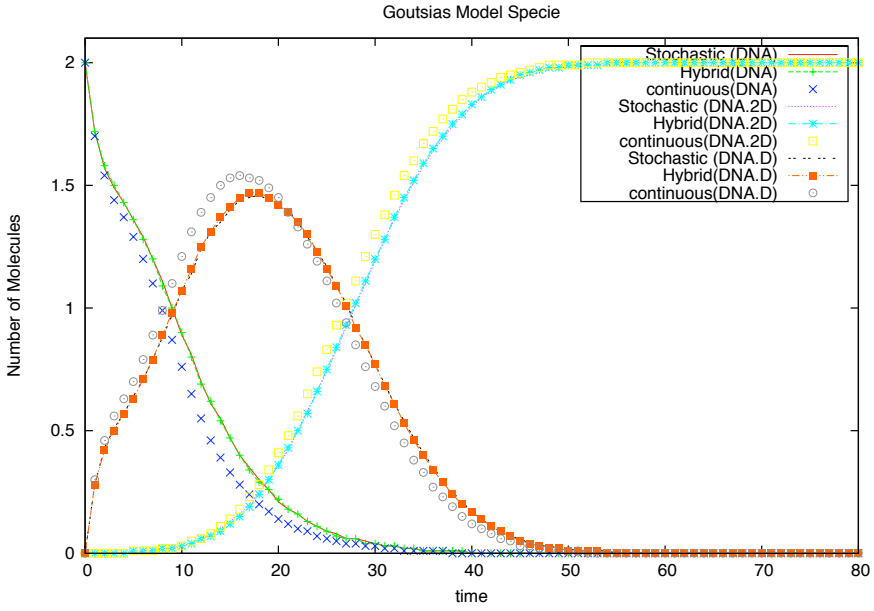

Figure 6: Continuous, stochastic and hybrid simulation result of the Goutsias model: hybrid simulation result is closer to stochastic than to continuous simulation result.

oscillations. However, if the rate constant of reaction R_{17}, i.e. $(k17)$, is changed from 0.2 to 0.08, the continuous simulation fails to produce the desired oscillation [HL07, JHNS02].

It is shown in [JHNS02] that stochastic simulation can still produce the expected

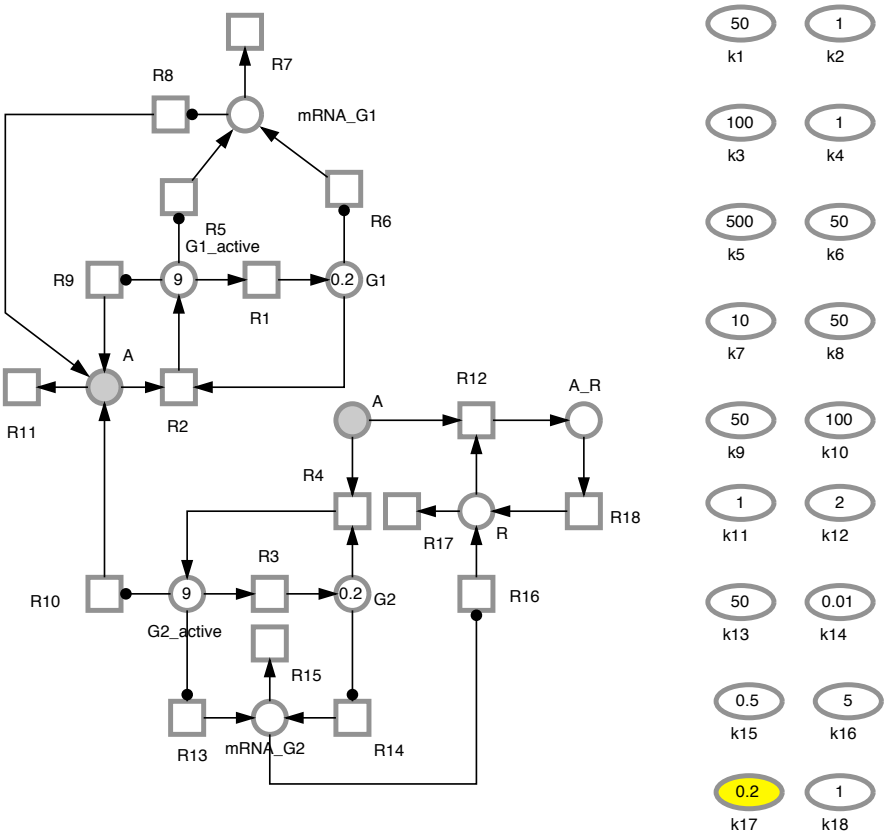

Figure 7: A $GHPN_{bio}$ representation of the circadian oscillation model. Places given in grey colour are logical places. The parameter k17, highlighted in yellow, is a key parameter in this model.

oscillation even if there are reactions with low rates.

Hybrid simulation can also produce such an oscillation of this model when species with low numbers of molecules or reactions of low rates are simulated stochastically, while the others are simulated continuously. However, static partitioning of the Petri net into continuous and stochastic parts will slow down the simulation, since the propensity values are changing during the simulation due to the oscillation.

Thus we opted to dynamically partition the model into fast and slow parts during the simulation. Figure 8 shows the simulation result when the Petri net in Figure 7 is simulated using continuous, stochastic, and hybrid methods with different values of k17 (0.2 and 0.08). The parameters in the partitioning algorithm are set to get a trade off between accuracy and speed.

5.3 T7 Phage Model

As a final example, we present the modelling of T7 phage viral kinetics [SYSY02] using Generalised hybrid Petri nets. Two different time scales can be distinguished in this model. One represents fast reactions and contains R_5 and R_6, and the other one comprises the slow reactions R_1, R_2, R_3 and R_4. Figure 9 gives a $GHPN_{bio}$ model of the six reactions which were published in [SYSY02]. The fast reactions are represented by continuous transitions, while slow reactions are modelled using stochastic transitions. Figure 10 shows a time course simulation result of the temp's population comparing between stochastic, continuous, and hybrid results. As a conclusion, hybrid simulation is closer to stochastic simulation rather than to the continuous one which means that stochasticity plays a role in this model because of low molecular species. Finally, static partitioning is chosen over dynamic one for simulating this model, because we can easily separate the reactions off-line into two subsets. In the next section we will discuss the issue of selecting between static and dynamic partitioning in more details.

Table 1: Comparison of continuous, stochastic and hybrid simulation run time (in seconds) for some $GHPN_{bio}$ models.

	Continuous	Stochastic	hybrid (static)	hybrid (dynamic)
Goutsias	0.01	0.972	0.014	0.138
Oscillator	0.258	5.995	4.21	1.991
T7 Phage	0.007	12.36	0.210	0.107

5.4 Discussion

Biological models vary in size and scales which explains the need for different tools to deal with such diversities. To illustrate this point consider for example Table 1 which compares continuous, stochastic, and hybrid simulation in terms of runtime needed for the three different models considered in our paper. The time is computed based on a

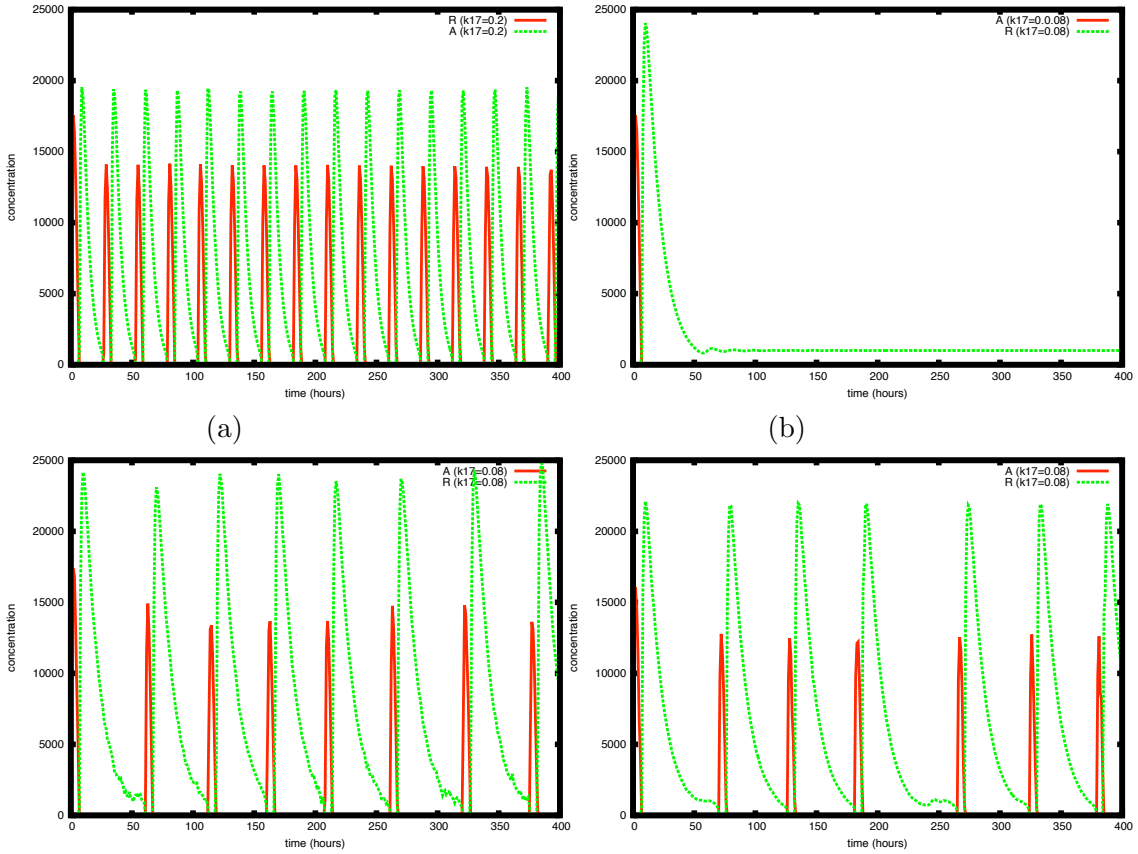

Figure 8: Simulation results of repressor (R) and activator(A) proteins of the circadian oscillation model for continuous, stochastic and hybrid methods with two different values of k17 (0.2 and 0.08). Continuous simulation produces sustained oscillation when k17=0.2 (a), but it fails when k17=0.08 (b). Contrary, stochastic simulation still produces oscillation in case of small parameter values (k17=0.08) (c), however it is computationally more expensive. Hybrid simulation is also able to produce oscillation when k17=0.08 (d), but with substantial improvement in speed.

single run for continuous simulation and as an average time for a single run for stochastic and hybrid simulation. For the stochastic and hybrid settings, 10^4 runs were used.

From this table we can conclude that there is no single optimal method (in terms of accuracy and speed) which always performs best for all models. Dynamic partitioning is more computational expensive for Goustias model than static partitioning, however dynamic partitioning is faster than static partitioning for T7 phage and circadian oscillator.

Stochastic simulation is always slower, but it is very accurate compared to continuous and hybrid simulation. This motivates us to provide with $GHPN_{bio}$ a unified framework

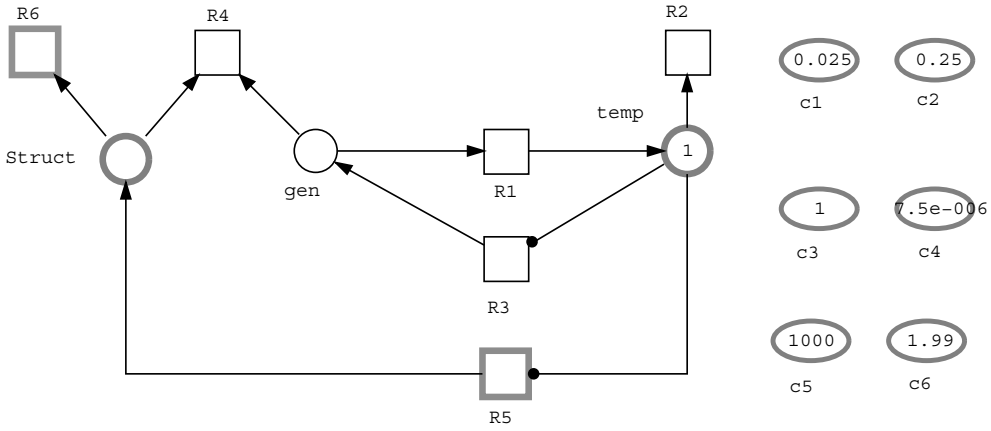

Figure 9: A $GHPN_{bio}$ representation of T7 phage model.

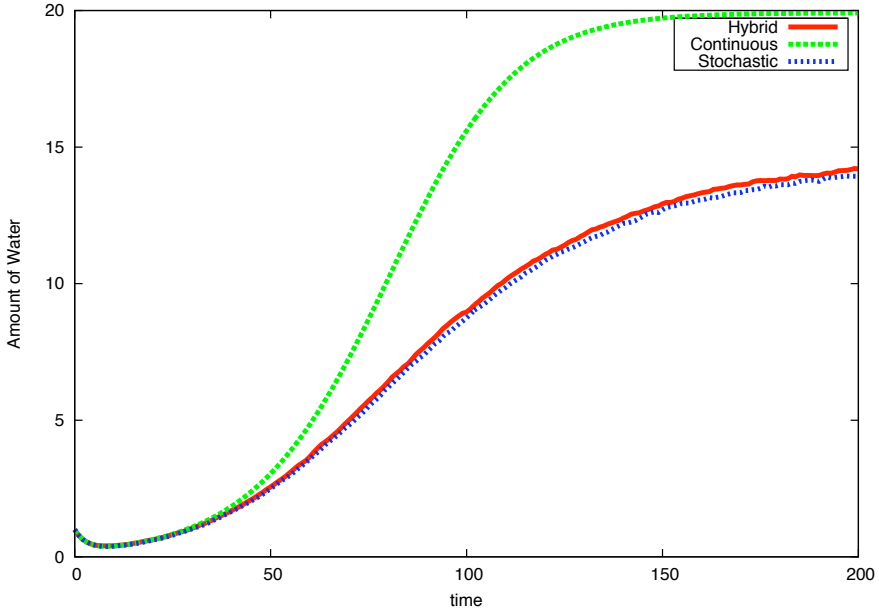

Figure 10: Time course simulation result of species tem in the T7 phage model.

to simulate one and the same model using different simulation methods which gives biologists a tool to easily try different methods and to choose the most suitable one.

6 Conclusions

In this paper, we have presented a new class of Petri nets which combines both Generalised stochastic Petri nets and continuous Petri nets into one net class, the Generalised Hybrid Petri Nets. The introduced net class has several functionalities which help biol-

ogists to model and simulate their biochemical networks through an easy to use visual language. $GHPN_{bio}$ models can be simulated statically through off-line partitioning or dynamically by on-line partitioning.

Deciding the partitioning off-line will save the dynamic partitioning overhead for certain biological applications. Providing an automatic way to achieve this goal could be important from the user's point of view. For this purpose, future investigation of analysis techniques of stochastic Petri nets might be of help. Another intended further extension is to support more than one stochastic simulator within Snoopy. This can be easily achieved due to the generic implementation of the simulator.

The cases studies which we have used in this paper have been chosen to illustrate different aspects of the hybrid approach. We are currently challenging Snoopy with other, more complex case studies than discussed in this paper.

7 Acknowledgements

Mostafa Herajy is 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

[ACT+05] A. Alfonsi, E. Cancès, G. Turinici, B. Ventura, and W. Huisinga. Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. *ESAIM: Proc.*, 14:1–13, 2005.

[AD98] H. Alla and R. David. Continuous and hybrid Petri nets. *J. Circ. Syst. Comp.*, 8(1):159 –188, 1998.

[Dra98] R. Drath. Hybrid object nets: an object oriented concept for modeling complex hybrid systems. In *3rd International Conference on Automation of Mixed Processes: Hybrid Dynamical Systems (ADPM)*, ADPM'98, pages 437–442, 1998.

[GCPS06] M. Griffith, T. Courtney, J. Peccoud, and W. Sanders. Dynamic partitioning for hybrid simulation of the bistable hiv-1 transactivation network. *Bioinformatics*, 22(22):2782–2789, 2006.

[GH06] D. Gilbert and M. Heiner. From Petri Nets to Differential Equations - An Integrative Approach for Biochemical Network Analysis. In Susanna Donatelli and P. Thiagarajan, editors, *Petri Nets and Other Models of Concurrency - ICATPN 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 181–200. Springer Berlin / Heidelberg, 2006.

[Gil76] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22(4):403 – 434, 1976.

[Gil77] D. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[Gil91] D. Gillespie. *Markov Processes: An Introduction for Physical Scientists.* Academic Press, October 1991.

[Gil07] D. Gillespie. Stochastic simulation of chemical kinetics. *Annual review of physical chemistry*, 58(1):35–55, 2007.

[Gou05] J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J. Chem. Phys.*, 122(184102):1–15, 2005.

[HBG+05] A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker, and C. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, 31:363–396, September 2005.

[HGD08] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer Berlin / Heidelberg, 2008.

[HK99] G. Horton and M. Kowarschik. Discrete-Continuous Modeling Using Hybrid Stochastic Petri Nets. In *European Simulation Symposium.* SCS Publishing House., 1999.

[HL07] A. Hellander and P. Lötstedt. Hybrid method for the chemical master equation. *J. Comput. Phys.*, 227:100–122, November 2007.

[HLGM09] M. Heiner, S. Lehrack, D. Gilbert, and W. Marwan. Extended Stochastic Petri Nets for Model-Based Design of Wetlab Experiments. In *Transactions on Computational Systems Biology XI*, pages 138–163. Springer, Berlin, Heidelberg, 2009.

[HMMW10] T. Henzinger, L. Mikeev, M. Mateescu, and V. Wolf. Hybrid numerical solution of the chemical master equation. In *Proceedings of CMSB '10*, CMSB '10, pages 55–65, New York, NY, USA, 2010. ACM.

[HR02] E. Haseltine and J. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.*, 117(15):6959–6969, 2002.

[JHNS02] J. Jose, H. Hao, N. Naama, and S. Stanislas. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):5988–5992, April 2002.

[KMS04] T. Kiehl, R. Mattheyses, and M. Simmons. Hybrid simulation of cellular behavior. *Bioinformatics*, 20:316–322, February 2004.

[LCPG08] H. Li, Y. Cao, L. Petzold, and D. Gillespie. Algorithms and software for stochastic simulation of biochemical reacting systems. *Biotechnol. Progr.*, 24(1):56–61, 2008.

[MA99] H. Mcadams and A. Arkin. It's a noisy business! *Trends in Genetics*, 15(2):65–69, 1999.

[MTA+03] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano. Biopathways representation and simulation on hybrid functional Petri net. *In silico biology*, 3(3), 2003.

[NDMM04] M. Nagasaki, A. Doi, H. Matsuno, and S. Miyano. A versatile Petri net based architecture for modeling and simulation of complex biological processes. *Genome informatics. International Conference on Genome Informatics*, 15(1):180–197, 2004.

[Pah09] J. Pahle. Biochemical simulations: stochastic, approximate stochastic and hybrid approaches. *Brief Bioinform*, 10(1):53–64, 2009.

[RMH10] C. Rohr, W. Marwan, and M. Heiner. Snoopy–a unifying Petri net framework to investigate biomolecular networks. *Bioinformatics*, 26(7):974–975, April 2010.

[RML93] V. Reddy, M. Mavrovouniotis, and M. Liebman. Petri net representations in metabolic pathways. In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, pages 328–336. AAAI Press, 1993.

[RPCG03] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. of Chem. Phys*, 119:12784–12794, 2003.

[SK05] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J. Chem. Phys*, 122(5), 2005.

[SYSY02] R. Srivastava, L. You, J. Summers, and J Yin. Stochastic vs. deterministic modeling of intracellular viral kinetics. *J. theor. Biol.*, 218(3):309–321, October 2002.

[TK93] K. Trivedi and V. Kulkarni. FSPNs: Fluid Stochastic Petri Nets. In *14th International Conference on Application and Theory of Petri Nets*, pages 24–31, 1993.

[WGMH10] V. Wolf, R. Goel, M. Mateescu, and T. Henzinger. Solving the chemical master equation using sliding windows. *BMC Syst. Biol.*, 4(1):42, 2010.

[WUKC04] O. Wolkenhauer, M. Ullah, W. Kolch, and K. Cho. Modeling and simulation of intracellular dynamics: choosing an appropriate framework. *IEEE Trans. Nanobiosci.*, 3(3):200–207, 2004.

[YLL09] H. Yang, C. Lin, and Q. Li. Hybrid simulation of biochemical systems using hybrid adaptive Petri nets. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, pages 410–420. ICST, 2009.