Bio-PEPA with SBML-like events

Federica Ciocchetta

Laboratory for Foundations of Computer Science, The University of Edinburgh,
Edinburgh EH8 9AB, Scotland

Abstract. In this work we present an extension of Bio-PEPA, a language recently defined for the modelling and analysis of biological systems, to handle *events*. Broadly speaking, events are constructs that represent changes in the system due to some trigger conditions. Some mappings from Bio-PEPA with events to analysis tools are reported. in order to test our approach, we present the translation of two biological models into Bio-PEPA with events.

1 Introduction

Computational models play an important role in systems biology. Indeed they help to study, analyze and predict the behaviour of biological systems. In recent years there have been some applications of process algebras for the analysis of biological systems [24, 22, 7, 8]. In most cases the analysis is performed using Gillespie's stochastic simulation algorithm [15]. Other possibilities exist, such as the mapping to differential equations [6].

Many biological models need to capture both discrete and continuous phenomena [1, 3, 20]. These models are called *hybrid systems*. A first example of hybrid system involves the activation of a certain activity when the concentration of enabling quantities is above the desired threshold. A second example considers a signal or stimuli that becomes null after some time leading to some changes in the interactions of the system.

In this work we present an extension of Bio-PEPA [8, 9], a language recently defined for the modelling and analysis of biological systems, to handle *events*. Broadly speaking, events are constructs that represent changes in the system due to some trigger conditions.

Here we are interested in simple forms of events. Specifically we refer to the definition of events reported in the SBML specification [19]. These kinds of events can be found in biochemical networks, such as the ones in BioModels database [21] or defined in some experimental settings. Indeed, in order to model some experiments, it may be necessary to render the possible change to the system, due, for instance, to the introduction of some reagents or the interruption of some external stimuli.

The idea underlying our work is the following:

Biological models with events $\Longrightarrow$ *Bio-PEPA with events* $\Longrightarrow$ *Analysis*

A first challenge concerns the modelling: we need to add events to the Bio-PEPA system. Events are added to our language as a set of elements and the rest of the syntax is unchanged. There are two motivations for this choice: first of all we keep the specification of the model as simple as possible, secondly this approach is appropriate when

we study the same biochemical system but with different experimental regimes. Indeed we can modify the list of events without any changes to the rest of the system.

A second aspect is the analysis. Some maps must be defined from Bio-PEPA to analysis tools. Specifically we map our language to Hybrid Automata (HA) [16]. HA are a formalism that consider both continuous and discrete changes. The continuous part is expressed by a set of variables evolving in each state according to a set of differential equations and the discrete dynamics is given by transitions between states, triggered by some conditions on variables. Furthermore, we can consider a modification of Gillespie's algorithm [15] in order to tackle events.

The rest of the paper is organised as follows. In the next Section we overview some related works. Section 3 reports a description of Bio-PEPA. In Section 4 we define the events we are considering in this work and then we extend Bio-PEPA in order to handle them. Section 5 describes the mapping from our language to Hybrid Automata. The mapping to stochastic simulation is reported in Section 6. After that, Section 7 illustrates the modelling in Bio-PEPA of some networks with events. Finally, in Section 8, some conclusions are reported.

2 Related works

The use of mathematical formalisms in order to represent discrete changes in biological systems is not new [1, 3, 20, 14, 4, 5]. In [1] the authors proposed a hybrid system approach to modelling an intra-cellular network using continuous differential equations to model some part of the system and mode-switching to describe the changes in the underlying dynamics. Some models with hybrid behaviour are presented and described by using CHARON [2], a language that allows formal description of hybrid systems. The authors of [20] discussed the use of discrete changes in biological systems and presented some examples by using the formalism HybridSAL [18]. Hybrid Concurrent Constraint Programming is used to model some biological systems with both discrete and continuous changes in [3]. In [4] the authors presented a map from stochastic Concurrent Constraint Programming (sCCP) to HA. The HA generated in this way are said to be able to capture some aspects of the dynamics which are lost if standard differential equations are used instead. A discussion about Hybrid Systems and Biology is reported in [5]. Finally, in [14] tha authors presented HYPE, a process algebra for the modelling of hybrid systems and used it to represent the repressilator. In none of these works SBML-like events are considered explicitly, but the focus is on general hybrid systems.

An approach to model events similar to the ones considered in this paper has been proposed in the *Beta Workbench (BetaWB)* [11] and in the associated programming language *BlenX* [25]. The BetaWB is a tool for modelling and simulating biological processes, based on Beta-binders, a recently introduced process algebra suitable for the biological applicative domain. The language allows us to represent some specific cases of events. Events can be considered as global rules of the environment, triggered only when the conditions associated with them are satisfied. Each event is the composition of a condition and an action verb. The possible actions are the join of two entities, the

split of one entity into two, the delete and the creation of a new entity. Each event is associated with a rate.

In BlenX more general events can be represented. As in the BetaWB, a single event is the composition of a condition and of an action, but the conditions can involve also the simulation time and the step size, in addition to the number of entities. Specifically, conditions are used to trigger the execution of an event when some elements are presents in the system, when a particular condition is met, with a given rate or at a precise simulation time or simulation step.

3 Bio-PEPA

Bio-PEPA [8, 9] is a language for the modelling and analysis of biochemical networks. The syntax of Bio-PEPA is defined as:

$$S ::= (\alpha, \kappa)\ \mathsf{op}\ S \mid S + S \mid C \qquad P ::= P \underset{I}{\bowtie} P \mid S(x)$$

where $\mathsf{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$.

The component S (*species component*) abstracts a biological species and the component P (*model component*) describes the system and the interactions among components. The prefix term $(\alpha, \kappa)\ \mathsf{op}\ S$ contains information about the role of the species in the reaction associated with the action type α: κ is the *stoichiometry coefficient* of the species and the *prefix combinator* "op" represents the role of the element in the reaction. Specifically, $\downarrow$ indicates a *reactant*, $\uparrow$ a *product*, $\oplus$ an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*. The operator "+" expresses choice between possible actions and the constant C is defined by an equation $C \stackrel{def}{=} S$. The parameter $x \in \mathbb{R}^+$ in $S(x)$ represents the concentration. Finally, the process $P \underset{I}{\bowtie} Q$ denotes the cooperation between components: the set I determines those activities on which the operands are forced to synchronize. In Bio-PEPA the rates are not expressed in the syntax of components but are defined as functional rates. These allow us to express any kind of kinetic law. Each action is associated with a specific functional rate.

A possible modelling style supported by Bio-PEPA is in terms of concentration levels. This is the style considered in the derivation of the transition system for Bio-PEPA. The species concentrations can be discretised into a number of levels. The granularity of the system is expressed in terms of the *step size* h, i.e. the length of the concentration interval representing a level. The information about the step sizes and the number of levels for each species is collected in a set $\mathcal{N}$. Specifically, the elements of the set $\mathcal{N}$ have the form: "$C : H = value\_H, N = value\_N, M = value\_M, V = value\_V, unit = value\_u$", where C is the species component name, H is the step size, N is the maximum level, M is the maximum concentration, V is the name of the enclosing compartment and unit is the unit for concentration.

In order to fully describe a biochemical network in Bio-PEPA we need to define structures that collect information about the compartments, the maximum concentrations, number of levels for all the species, the constant parameters and the functional rates. The Bio-PEPA system is defined in the following way:

Definition 1. *A Bio-PEPA system $\mathcal{P}$ is a 6-tuple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Components, P \rangle$, where: $\mathcal{V}$ is the set of compartments, $\mathcal{N}$ is the set of quantities describing species, $\mathcal{K}$ is the set*

of parameter definitions, $\mathcal{F}_R$ is the set of functional rates, Components is the set of definitions of sequential components, P is the model component describing the system.

For details see [8, 9]. The behaviour of the system is defined in terms of an operational semantics. This refers to the level-based modelling style and in this context the parameter in the species components stands for the concentration level. We defined two relations. The former, called *capability relation*, is indicated by $\xrightarrow{\theta}$. The label θ is of the form (α, w), where $w := [S : op(l, \kappa)] \mid w :: w$, with S a species component, l the level and κ the stoichiometry coefficient. This relation is defined as the minimum relation satisfying the rules reported in Table 1.

The latter relation, called *stochastic relation*, is $\mapsto \subseteq \tilde{\mathcal{P}} \times \Gamma \times \tilde{\mathcal{P}}$, where Γ is the set of labels and $\tilde{\mathcal{P}}$ is the set of well-defined Bio-PEPA systems. This relation is defined as the minimal relation satisfying the rule:

$$\texttt{Final} \quad \frac{P \xrightarrow{(\alpha_j, w)} P'}{\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P \rangle \xmapsto{(\alpha_j, r_\alpha[w, \mathcal{N}, \mathcal{K}])} \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P' \rangle}$$

The element $r_\alpha[w, \mathcal{N}, \mathcal{K}]$ is the rate associated with the action α and is defined as:

$$r_\alpha[w, \mathcal{N}, \mathcal{K}] = \frac{f_\alpha[w, \mathcal{N}, \mathcal{K}]}{h}$$

where h is the step size for the species involved in the reaction and the notation $f_\alpha[w, \mathcal{N}, \mathcal{K}]$ means that the function f_α is evaluated over w and the information about parameters and species components contained in the sets $\mathcal{N}$ and $\mathcal{K}$.

In this definition r_α represents the parameter of a negative exponential distribution. The dynamic behaviour of processes is determined by a *race condition*: all activities enabled attempt to proceed but only the fastest succeeds.

A *Stochastic Labelled Transition System (SLTS)* is defined for a Bio-PEPA system. From this we can obtain a *continuous time Markov Chain (CTMC)*. Both the SLTS and the CTMC derived from Bio-PEPA are defined in terms of levels of concentration. We call this Markov chain *CTMC with levels*.

Bio-PEPA can be seen as an *intermediate, formal, compositional* representation of biological systems, from which different kinds of analysis can be performed. We have defined some mappings from Bio-PEPA to ODEs, CTMC, Gillespie's model and PRISM [23]. A tool for the analysis of Bio-PEPA system is under implementation **(UPDATE)**.

In the following we report a brief description of the mapping from Bio-PEPA to ODE, as it is used later in the paper. For details about it and the other mapping see [9].

3.1 From Bio-PEPA to ODE system (π_{ODE})

Let π_{ODE} be the definition of the set of ODEs from a Bio-PEPA system.

π_{ODE} entails three steps: 1) definition of the stoichiometry $(n \times m)$ matrix D, where n is the number of species and m is the number of reactions; 2) definition of the *kinetic*

`prefixReac` $((\alpha,\kappa)\downarrow S)(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])} S(l-\kappa) \quad \kappa \le l \le N$

`prefixProd` $((\alpha,\kappa)\uparrow S)(l) \xrightarrow{(\alpha,[S:\uparrow(l,\kappa)])} S(l+\kappa) \quad 0 \le l \le (N-\kappa)$

`prefixMod` $((\alpha,\kappa)\,op\,S)(l) \xrightarrow{(\alpha,[S:op(l,\kappa)])} S(l)$ with $op = \odot, \oplus, \ominus$ and $0 < l \le N$ if $op = \oplus$, $0 \le l \le N$ otherwise

`choice1` $$\frac{S_1(l) \xrightarrow{(\alpha,w)} S_1'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)} S_1'(l')}$$

`choice2` $$\frac{S_2(l) \xrightarrow{(\alpha,w)} S_2'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)} S_2'(l')}$$

`constant` $$\frac{S(l) \xrightarrow{(\alpha,S:[op(l,\kappa)])} S'(l')}{C(l) \xrightarrow{(\alpha,C:[op(l,\kappa)])} S'(l')} \text{ with } C \stackrel{def}{=} S$$

`coop1` $$\frac{P_1 \xrightarrow{(\alpha,w)} P_1'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)} P_1' \underset{\mathcal{L}}{\bowtie} P_2} \text{ with } \alpha \notin \mathcal{L}$$

`coop2` $$\frac{P_2 \xrightarrow{(\alpha,w)} P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)} P_1 \underset{\mathcal{L}}{\bowtie} P_2'} \text{ with } \alpha \notin \mathcal{L}$$

`coop3` $$\frac{P_1 \xrightarrow{(\alpha,w_1)} P_1' \quad P_2 \xrightarrow{(\alpha,w_2)} P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w_1::w_2)} P_1' \underset{\mathcal{L}}{\bowtie} P_2'} \text{ with } \alpha \in \mathcal{L}$$

Table 1. Axioms and rules for Bio-PEPA.

law vector $(m \times 1)$ $\mathbf{v_{KL}}$ containing the kinetic laws of each reaction; 3) definition of the vector $(n \times 1)$ $\mathbf{x}$, with $\mathbf{x}^T = (x_1, x_2, ..., x_n)$.

A crucial part is the derivation of the stoichiometry matrix $D = \{d_{ij}\}$. The entries of the matrix are obtained as follows: for each sequential component C_i consider the prefix subterms C_{ij} representing the contribution of the species i to the reaction j. If the term represents a reactant we write the corresponding stoichiometry κ_{ij} as $-\kappa_{ij}$ in the entry d_{ij}. In the case of a product we write $+\kappa_{ij}$. All other cases are null. The kinetic law vector is derived form the functional rates and its definition is straigthforward.

The ODE system thus obtained has the form:

$$\frac{d\mathbf{x}}{dt} = D \times \mathbf{v_{KL}}$$

where the vector of initial concentrations is $\mathbf{x_0}$, with $x_{i,0}$ the intial concentration of the species i, as given in the specification of the system.

4 Bio-PEPA with events

4.1 SBML-like events: some definitions

In this work we consider events as defined in the SBML specification [19]. SBML events describe explicit discontinuous state changes in the model. Specifically, an SBML event has the following structure:

"*event_id*, if *trigger* then *event_assignment_list* with *delay*"

where:

- *event_id* is the event identifier,
- *trigger* is a mathematical expression that, when it is evaluated to true, makes the event fire. It can be composed of one or more conditions;
- *event_assignment_list* is a list of assignments that are made when the event is executed;
- *delay* is the length of time between the time when the event fires and the time when the event assignments are executed.

The trigger and the list of assignments are both mandatory and can involve parameters, species concentrations and compartment sizes. All the triggers are initially evaluated to false.

Some definitions concerning events are reported below.

Definition 2. *An SBML-like event "event_id, if trigger then event_assignment_list with delay" is immediate if delay is equal to zero. Otherwise the event is called delayed.*

A possible classification of events is reported in the following definition.

Definition 3. *Two or more SBML-like events are sequential if they are fired one after the other in given order. They are said to be simultaneous if they happen at the same point in time.*

In most biochemical systems we are interested in we have sequential events. In the general situation of simultaneous events sometimes it is necessary to give some *tie-breaking* rules to decide which of any set of events is simulated first. The most common way to do this is to use some rule that assigns a priority to each event [10]. When there are two or more simultaneous events, the event with the highest priority is defined to be the next event to fire. However not always the order in which a set of simultaneous events is fired is important, as in the case the assigments of the events influence different variables. We have the following definition:

Definition 4. *Two simultaneous events are independent if their event assignements do not effect each other. Otherwise they are called dependent.*

If we have simultaneous indipendent events we may abstract them as a single event and the system is reset according to the assignments of all the set of simultaneous events. Simultaneous indipendent events are dealt similarly to sequential ones.

4.2 Assumptions

We make the following assumptions for the events considered in this work.

1. Triggers can involve time and species components' names, while assignments can involve species components (concentrations/levels), compartments (size), parameters (values) and functional rates (function definitions);
2. The triggers are deterministic, i.e. when they become true they are fired;
3. The triggers are only unidirectional, i.e. describing the change from one mode to another, but non viceversa. Bidirectional triggers can be decomposed into two unidirectional triggers;
4. The events are either sequential or simultaneous and indipendent.

These assumptions are not restrictive. Indeed these events allow us to represent a large number of discontinuous changes that we can find in biological models.

4.3 The definition of the language

Generally speaking we can add events to the Bio-PEPA model by introducing a *set* of elements that have the form $(id, trigger, event\_assignment, delay)$, where *id* is the name of the event, *trigger* is a mathematical expression involving the components of Bio-PEPA model and time, *event_assignment* is a list of assignments, *delay* is 0 (*immediate events*) or non-negative real value (*delayed events*). Formally, we have the following definitions:

trigger ::= cond | cond **or** cond | cond **and** cond | **not** cond;
cond ::= t eq value | $exp(\bar{C}, \bar{k})$ eq value | $exp(\bar{C}, \bar{k})$ eq $exp(\bar{C}, \bar{k})$
eq ::= $= \mid \neq \mid > \mid < \mid \leq \mid \geq$ delay ::= value
event_assignment ::= assignment ; event_assignment
assignment ::= k $\leftarrow$ value | level(C) $\leftarrow$ value | $f_\alpha \leftarrow exp(\bar{C}, \bar{k})$
event ::= (id, trigger, event_assignment, delay)

where C stands for any sequential component and k for any parameter, the variable $t \in \mathbb{R}^+$ represents the global time of the system, $exp(\bar{C}, \bar{k})$ is an arithmetic expression involving a set of components (denoted $\bar{C}$) and a set of parameters (denoted $\bar{k}$), $value \in \mathbb{R}^+$ and id is a string indicating the event name. The function $level(C)$ associates a level with the component C. When we need the original value for the concentration, we write $C = value_C\{l_C\}$, where $value_C$ is the value of the concentration and l_C is the associated level. The set of events is then defined as:

$$Events \ ::= \ [] \ | \ event :: Events$$

Definition 5. *A Bio-PEPA system with events $\mathcal{P}_E$ is a 8-nuple $< \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P, Events, t >$, where Events is the set of events, $t \in \mathbb{R}^+$ is the variable expressing time and the other elements are as in the standard Bio-PEPA.*

A Bio-PEPA system is well-defined if all the elements are well-defined. The definition of well-definedness for all the elements, with the exception of events, is reported in [9].

Definition 6. *The set of events Events is well-defined if and only if the following conditions hold:*

- *triggers involve time or the name of species components, assignments involve species components, comaprtments, parameters and functional rates;*
- *all the elements used in the events are defined in the Bio-PEPA system;*
- *all the triggers are different and do not overlap in their values;*
- *given an event, the different assignments are independent (i.e. involve different elements).*

In the following we refer to Bio-PEPA with events simply as Bio-PEPA. Only well-defined Bio-PEPA systems are considered.

5 Mapping to Hybrid Automata

5.1 Hybrid Automata

Hybrid automata (HA) [16] combine discrete transition graphs with continuous dynamical systems. They are used to formally model hybrid systems, dynamical systems with both discrete and continuous components. An hybrid automaton consists of a finite set of *real-values variables* $\{X_1, X_2, ..., X_n\}$ and a finite labelled graph, whose vertices correspond to *control modes* (states), described by differential equations, and whose edges are *control switches*, corresponding to discrete events. In addition, we have some labels for the edges, specifying the *jump conditions* (activation conditions) and labels for the vertices, containing information about initial and invariant conditions. The variables evolve continuously in time, apart from some changes induced by events. When an event happens there is a change in the mode. The dynamic behaviour of each mode is described by a set of differential equations, generally different from mode to mode. We can use HA both for simulation (see for instance *the SHIFT language* [12]) and model checking (see *HyTech* [17]). In this work we limit our attention to simulation. For a formal definition and details about the formalism see [16].

5.2 Definition of the mapping

Here we present the map from Bio-PEPA with events to HA. First we limit our attention to the case of immediate events and then we show a way to represent delayed events. Indeed the translation of the delay associated with an event is not straightforward in the usual definition of HA.

Let $\mathcal{P}_0 =< \mathcal{V}_0, \mathcal{N}_0, \mathcal{K}_0, \mathcal{F}_0, Comp_0, P_0, Events, t_0 >$ be the initial Bio-PEPA system and let N_{events} be the number of events. We have the following correspondences:

1. Each species component C_i in $Comp$ is associated with a variable X_i. The set of variables is then given by: $\{X_1, X_2, ..., X_{N_{Comp}}, t\}$, where t is the variable expressing the time and N_{Comp} is the number of species components. The evolution of the variable t is described by the trivial differential equation $dt/dt = 1$.
2. The initial conditions of the variables are derived from the initial model component P_0. The variable t is initially set to 0.
3. For each event i in $Events$, we can consider the trigger tr_i. We use these triggers to define the jump conditions. In the case we have only sequential events, the number of possible jump conditions N_{jump} is just N_{Events}. If simultaneous indipendent events are possible, we may combine them together in order to define a new jump condition representing the union of the triggers of the simultaneous events. In this case, the system is reset according to the union of the assigment lists of the events involved.
4. Each mode is described by a specific instance of the Bio-PEPA system. Indeed modes are defined according to either the initial system or the system modified with the event assignments relative to a trigger. The number of modes is $N_{jump} + 1$. We indicate the modes with σ and the set of modes Σ. In each mode some invariant conditions are added in order to force the change of mode when the trigger becomes true. We have that:
   - The initial mode σ_0 is defined from the initial system $\mathcal{P}_0$. It is described in terms of an ODE system and this is derived from the Bio-PEPA model by considering the map π_{ODE}. Therefore we have $\sigma_0 = \pi_{ODE}(\mathcal{P}_0)$.
   - Given a mode $\sigma_i = \pi_{ODE}(\mathcal{P}_i)$, let tr_{ij} be one possible jump condition that can be satisfied from it. We define the Bio-PEPA system $\mathcal{P}_j = \mathcal{P}_i[event\_assignment_{ij}]$ as the reset of the previous system $\mathcal{P}_i$ according to the event assignments associated with the trigger. The mode σ_j is then defined as $\sigma_j = \pi_{ODE}(\mathcal{P}_j)$.

Case of delayed events The delay associated with an event represents the time interval between when the event is fired and when its assigments are executed. This information cannot be directly translated in any of the components of standard HA. A possible way to handle delay is reported below.

First we introduce a new variable t_{mode} representing the time when the system enters in a specific mode. It is initially set to zero. The differential equation associated with this new variable is $dt_{mode}/dt = 0$, i.e. this variable is constant in each mode.

Secondly, given an event $(id, trigger, event\_assignment, delay)$, we split it into two immediate events, defined as:

1. $(id_1, trigger, t_{mode} \leftarrow t, 0)$;

2. $(id_2, t = t_{mode} + delay, assigment, 0)$.

The role of the former event is to introduce the delay whereas the role of the second is to guarantee that the assigments of the initial event are executed after the given delay.

6 Stochastic simulation by Gillespie's algorithm

One of the possible kinds of analysis supported by Bio-PEPA is stochastic simulation by Gillepsie's algorithm [9]. When events are considered the algorithm has to be modified in order to handle them. Broadly speaking, events are tackled by adding some conditions and some checks along the simulation. We start at time $t = 0$, with the Bio-PEPA system at the initial conditions. We assume that initially all the triggers are evaluated to false. When one of the conditions is satisfied, the simulation stops and the system is reset according to the event assignments associated with the trigger. After that, the simulation can start again until another condition becomes true or the simulation time is reached.

We propose the following procedure for each simulation run.

1. Let $\mathcal{P}_0$ be the initial Bio-PEPA system and $time\_S$ the maximum simulation time.
2. While $t < time\_S$ and $trigger_i = false$ for $i = 1, 2, ..., N_{Events}$, simulate.
3. If $t \geq time\_S$ then stop.
4. If $t < time\_S$ and there exists a $trigger_i$ such that it is true, we have that:
   (a) if $delay = 0$ reset the Bio-PEPA system according to the event assignments associated with that trigger: $\mathcal{P}'(t) = \mathcal{P}(t)[event\_assignment_i]$. Go to (2).
   (b) if $delay > 0$ go on with the simulation till time $t + delay$ and then proceed as in (a).

Some final observations concern how to use the algorithm in two particular situations.

- In the case of two or more indipendent simultaneous events we proceed as observed in Section 4.1: we can abstract these events as a single event and therefore we modify the system according to the assigments associated with all the events involved.
- When we have an event with a trigger of type "$t = \tilde{t}$", the time $\tilde{t}$ may not correpond to any of the simulation time point. Specifically, there exist two consecutive time points t_j and t_{j+1} such that $t_j < \tilde{t} < t_{j+1}$. In order to handle this situation we consider the followingapproach:
  1. if $delay = 0$ or $t_j < \tilde{t} + delay < t_{j+1}$ with $delay > 0$ then consider the system at time t_j and reset it.
  2. If $delay > 0$ and $\tilde{t} + delay \geq t_{j+1}$ consider the last simulation time point $t_h \leq \tilde{t} + delay$ and reset the system at time t_h.

7 Examples

In this Section we present two examples to illustrate our approach.

7.1 Genetic network with an involving species concentration

This first example concerns a simple genetic network, composed of two genes. A gene X activates the expression of gene Y; above a certain threshold, gene Y inhibits expression of gene X. The reactions describing this situation are:

- activation of the gene Y: $X \xrightarrow{r_1} X + Y$, with $r_1 = 0.01$;
- degradation of the gene X: $X \xrightarrow{r_2} \emptyset$ with $r_2 = 0.02$;
- creation of the gene X: $\emptyset \xrightarrow{r_3} X$ with $r_3 = 0.01$, possible when the concentration of Y is less than 0.8.

The Bio-PEPA species components[1] corresponding to the two genes are:

$$X \stackrel{def}{=} (\alpha_1, 1) \oplus X + (\alpha_2, 1)\downarrow X + (\alpha_3, 1)\uparrow X; \qquad Y \stackrel{def}{=} (\alpha_1, 1)\uparrow Y;$$

whereas the model component is:

$$X(0) \underset{\{\alpha_1\}}{\bowtie} Y(0)$$

The initial values are zero for both the genes. The functional rates are all of kind mass-action:

$$f_{\alpha_1} = fMA(0.01); \quad f_{\alpha_2} = fMA(0.02); \quad f_{\alpha_3} = fMA(0.01);$$

Furthermore we have the following event involving concentrations:

$$[(event_1, Y = 0.8\{1\}, r_3 \leftarrow 0, 0)]$$

A description in terms of HA is reported in Fig.1. We have two modes, one describing the case $y < 0.8$ and the other the case $y \geq 0.8$. The systems in the two modes are the similar, but in the former case the reaction α_3 is activated, in the second case not. The guard to move from one mode to the other is "$y = 0.8$".

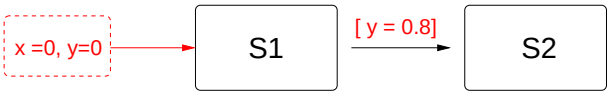

Fig. 1. HA representation for the network with an event involving concentration.

In the Figure 1 $S1$ and $S2$ represents the two ODE models representing the system when $y < 0.8$ and $y \geq 0.8$, respectively. These two systems are:

$$S1 = \begin{cases} \frac{dx}{dt} = -0.02 \cdot x + 0.01; \\ \frac{dy}{dt} = 0.01 \cdot x \end{cases}$$

[1] Note that we use X and Y (capital letters) to indicate the names of the species and the name of the Bio-PEPA component, whereas x and y indicate the associated species concentration.

$$S2 = \begin{cases} \frac{dx}{dt} = -0.02 \cdot x; \\ \frac{dy}{dt} = 0.01 \cdot x \end{cases}$$

The initial conditions are $x = 0,\ y = 0$.

Some simulation results are reported in Fig.2.

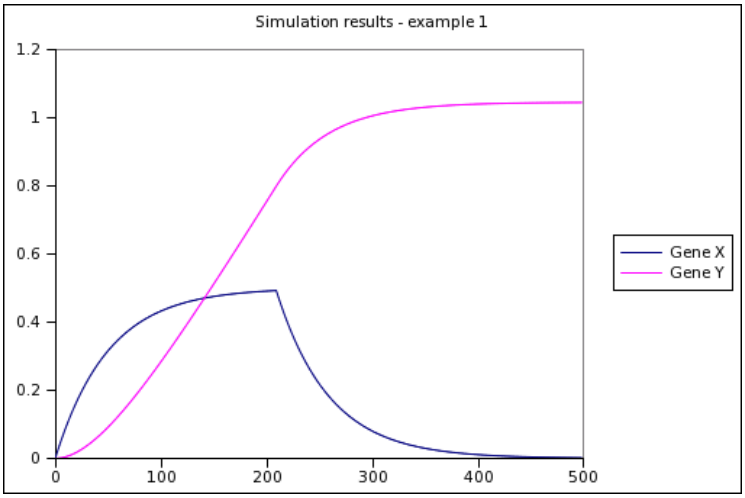

Fig. 2. Simulation results for the network with an event involving concentration.

7.2 The acetylcholine receptor model

This example concerns the functional properties of the *nicotin Acetylcholine Receptors* (nAChR). These are transmembrane proteins that mediate inter-conversions between open and closed channel states under the control of neurotransmitters. The detailed description of the model is reported in [13].

A schema of the model is shown in Figure 3. B (*Basal state*), A (*Active state*), D (*Desensitized state*) and I (*Inactivable state*) represent the different states of the acetylcholine receptors. The numbers 0, 1, 2 associated with the state represent the number of ligands (denoted X) bound to a receptor. In the model the ligands are not modelled explicetly. Each column corresponds to a series of ligand binding actions at two identical sites per receptor whereas each row corresponds to a series of transactions between conformational states. All the reactions are reversible and the dynamics are described by mass-action laws. For each reaction i, with $i = 1, 2, ...16$, the rate of the forward direction is kf_i and the rate of the reverse reaction kr_i.

In addition, there is an event to describe the recovery upon removal of free agonist at a given time. This is expressed by constraining the reaction rates of each second-order ligand-receptor reaction to zero. These constraints prevent ligand binding reactions from happening after that time, hence the states evolve as if the free ligands were

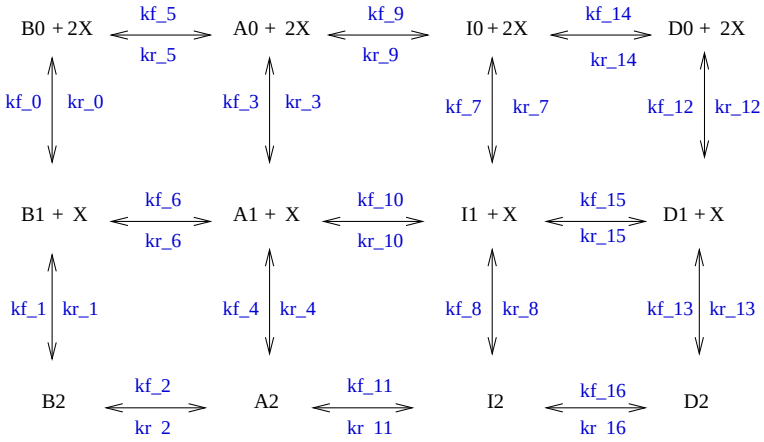

Fig. 3. Schema of the acetylcholine receptor model.

parameter	value	parameter	value	parameter	value	parameter	value
kf_0	3000	kr_0	8000	kf_1	1500	kr_1	16000
kf_2	30000	kr_2	700	kf_3	3000	kr_3	8.64
kf_4	1500	kr_4	17.28	kf_5	0.54	kr_5	10800
kf_6	130	kr_6	2740	kf_7	3000	kr_7	4
kf_8	1500	kr_8	8	kf_9	19.7	kr_9	3.74
kf_{10}	19.85	kr_{10}	1.74	kf_{11}	20	kr_{11}	0.81
kf_{12}	3000	kr_{12}	4	kf_{13}	1500	kr_{13}	8
kf_{14}	0.05	kr_{14}	0.0012	kf_{15}	0.05	kr_{15}	0.0012
kf_{16}	0.05	kr_{16}	0.0012				

Table 2. Acetylcholine receptor model. The list of parameters. The unit is s^{-1}.

completely removed from the system. The event is immediate, the trigger is "$t = t_2$", where $t_2 = 20\ s$, and the event assignments are $kf_0 = 0,\ kf_1 = 0,\ kf_3 = 0,\ kf_4 = 0,\ kf_7 = 0\ , kf_8 = 0\ , kf_{12} = 0\ , kf_{13} = 0$.

The Bio-PEPA system associated with the Edelstein's model In the following we report briefly the definition of the Bio-PEPA system representing the Edelstein's model. The complete system is reported in the Appendix A.

- *Definition of the compartment list $\mathcal{V}$*. In the model we have a single three-dimensional compartment, defined as "$comp1 :\ 1e\text{-}16,\ l$;", where l is litre.
- *Definition of the set $\mathcal{N}$*. Each species is associated with a species component. For each species component we have to declare the step-size, the number of levels, the initial and maximum concentrations and the compartment where the species is. The ligand is not represented explicitly. For instance, in the case of $B0$, $B1$ and $B2$ we have:

$$\begin{array}{l} B0 : H = h,\ N = N_{B0},\ M = M_{B0},\ V = comp1,\ unit = \mu M; \\ B1 : H = h,\ N = N_{B1},\ M = M_{B1},\ V = comp1,\ unit = \mu M; \\ B2 : H = h,\ N = N_{B2},\ M = M_{B2},\ V = comp1,\ unit = \mu M; \end{array}$$

  where the step size h is $1.66e\text{-}5$, the number of levels $N_{B0} = N_{B1} = N_{B2}$ is 1 (i.e. the species can be present, 1, or absent, 0), the maximum concentration $M_{B0} = M_{B1} = M_{B2}$ is $1.66e\text{-}5$ and coincides with the initial concentration of channels at the basal state.
- *Definition of functional rates ($\mathcal{F}_R$) and parameters ($\mathcal{K}$)*. Each reversible reaction in the model is decomposed in two irreversible reactions. For each reaction i with $i = 0, ..., 16$ we have the kinetic laws $f_{\alpha\_f_i} = fMA(kf\_i)$; and $f_{\alpha\_r_i} = fMA(kr\_i)$, where fMA stands for mass-action. All the parameters are defined in the set $\mathcal{K}$. The values are the ones reported in the paper.

- *Definition of species components (Comp) and of the model component (P).*
  In the following we report the definition for $B0$, $B1$ and $B2$; the other species are dealt with similarly.

$$\begin{array}{rcl} B0 & \stackrel{def}{=} & (\alpha\_f_0, 1)\downarrow B0 + (\alpha\_r_0, 1)\uparrow B0 + (\alpha\_f_5, 1)\downarrow B0 + (\alpha\_r_5, 1)\uparrow B0; \\ B1 & \stackrel{def}{=} & (\alpha\_f_0, 1)\uparrow B1 + (\alpha\_r_0, 1)\downarrow B1 + (\alpha\_f_6, 1)\downarrow B1 + (\alpha\_r_6, 1)\uparrow B1 + \\ & & (\alpha\_f_1, 1)\uparrow B1 + (\alpha\_r_1, 1)\downarrow B1; \\ B2 & \stackrel{def}{=} & (\alpha\_f_2, 1)\downarrow B2 + (\alpha\_r_2, 1)\uparrow B2 + (\alpha\_f_1, 1)\uparrow B2 + (\alpha\_r_1, 1)\downarrow B2; \end{array}$$

  The system is described as:

$$\begin{array}{c} B0(1.66e\text{-}5) \underset{L1}{\bowtie} B1(0) \underset{L2}{\bowtie} B2(0) \underset{L3}{\bowtie} A0(0) \underset{L3}{\bowtie} A1(0) \underset{L4}{\bowtie} A2(0) \underset{L5}{\bowtie} \\ I0(0) \underset{L6}{\bowtie} I1(0) \underset{L7}{\bowtie} I2(0) \underset{L8}{\bowtie} D0(0) \underset{L9}{\bowtie} D1(0) \underset{L10}{\bowtie} D2(0) \end{array}$$

  where L_i, $i = 1, ..., 10$ are the cooperation sets..
- *Definition of events*. We have only one event: $[(event_1, t = 20,\ kf_0 = 0;\ kf_1 = 0;\ kf_3 = 0\ kf_4 = 0;\ kf_7 = 0;\ kf_8 = 0;\ kf_{12} = 0;\ kf_{13} = 0, 0)]$

Analysis results The HA associated with the Edelstein's model is similar to the one for the first example 7.1. We have two modes, described by two different sets of differential equations. The trigger condition involves time and it is "$t = 20\ s$". The details of the two systems describing each mode are not reported.

Simulation results made by using Gillespie's algorithm are reported in Fig. 4. The initial number of molecules for $B0$ is given $M_0 \times V \times Na = (1.66e\text{-}5\ \mu M) \times (1.e\text{-}16\ l) \times (6.022 \times e{+}23\ (mol)^{-1}) = 1000$, where Na is the Avogadro number[2]. All the other species are initially null. The number of runs is 100. The graph reproduces results in agreement with the ones reported in the paper [13]. Following the ligand removal, the state $I2$ loses agonist molecules and is transformed to the state $B0$ very rapidly, while $D2$ loses ligand molecules to form $D0$. Since the data occur on a wide range of times we represent the time on a logarithmic scale.

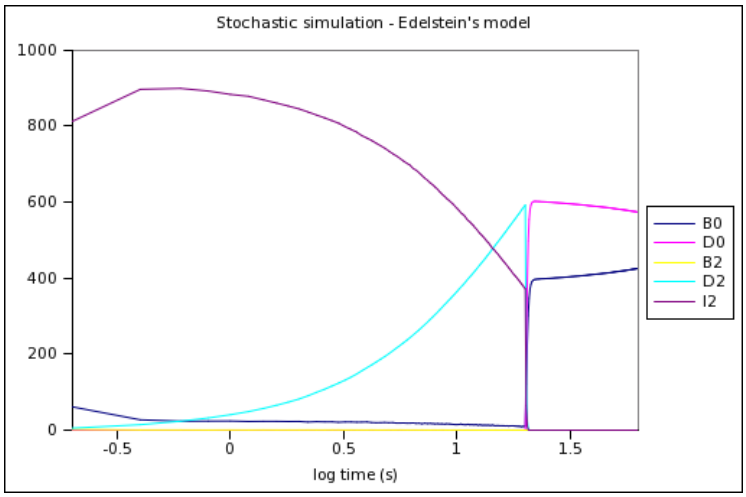

Fig. 4. Stochastic simulation results for the Edelstein's model.

8 Conclusions

In this work we present an extension of Bio-PEPA to handle *SBML-like events*. Events are constructs that represent changes in the system due to some trigger conditions. The events considered here are simple, but nevertheless able to describe most of the discontinuous changes in models and experiments. Events are added to our language without any modification to the rest of the syntax. The main motivation of this choice is that we want to keep the specification of the model as simple as possible. Furthermore this approach is appropriate when we study the same biochemical system but with different experimental regimes.

[2] It is the number of "entities" (atoms or molecules) in one mole of substance.

A topic for the future concerns the study of more general events and the possible extension to other kinds of hybrid systems in biology. Furthermore we plan to exploit the possible kinds of analysis involving hybrid systems in the context of systems biology. In this paper we focus on the mapping to Hybrid Automata and stochastic simulation by (a modification of) Gillespie's algorithm. Further investigation will concern the application of model checking for the study of the properties of biological systems.

Acknowledgements

The author thanks Jane Hillston and Vashti Galpin for her helpful comments. The author is supported by the EPSRC under the CODA project "Process Algebra Approaches for Collective Dynamics" (EP/c54370x/01 and ARF EP/c543696/01).

A Appendix: Bio-PEPA system for the Edelstein's model

In this appendix we report the specification of the whole Edelstein's model into Bio-PEPA. Note that, in the definition of the species component, we use the notation << to indicate that the species is a product (it corresponds to the operator $\uparrow$) and >> to indicate a reactant (it corresponds to the operator $\downarrow$).

```
\\Definition of compartments
comp1: 1e-16, l;

\\Definition information about species components
[ B0: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  B1: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  B2: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  A0: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  A1: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  A2: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  I0: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  I1: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  I2: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  D0: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  D1: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM;
  D2: H= 1.66-5, N= 2, M= 1.66-5, V= comp1, unit = muM ]

\\Definition of parameters
[ kf_0= 3000;  kr_0= 8000;  kf_1= 1500;  kr_1= 16000;
  kf_2=30000;  kr_2= 700;  kf_3= 3000;  kr_3= 8.64;
  kf_4= 1500;  kr_4= 17.28;  kf_5= 0.54;  kr_5= 10800;
  kf_6= 130;  kr_6= 2740;  kf_7= 3000;  kr_7= 4;
  kf_8= 1500;  kr_8= 8;  kf_9= 19.7;  kr_9= 3.74;
  kf_10= 19.85;  kr_10= 1.74;  kf_11= 20;  kr_11= 0.81;
  kf_12= 3000;  kr_12= 4; kf_13= 1500;  kr_13= 8;
  kf_14= 0.05;  kr_14= 0.0012; kf_15= 0.05; kr_15= 0.0012;
  kf_16= 0.05;  kr_16=  0.0012 ]
```

```
\\Definition of functional rates
\\all kinetic laws are MA
[ f_alpha_f_0= fMA(kf_0);  f_alpha_r_0=  fMA(kr_0);
  f_alpha_f_1= fMA(kf_1);  f_alpha_r_1=  fMA(kr_1);
  f_alpha_f_2= fMA(kf_2);  f_alpha_r_2=  fMA(kr_2);
  f_alpha_f_3= fMA(kf_3);  f_alpha_r_3=  fMA(kr_3);
  f_alpha_f_4= fMA(kf_4);  f_alpha_r_4=  fMA(kr_4);
  f_alpha_f_5= fMA(kf_5);  f_alpha_r_5=  fMA(kr_5);
  f_alpha_f_6= fMA(kf_6);  f_alpha_r_6=  fMA(kr_6);
  f_alpha_f_7= fMA(kf_7);  f_alpha_r_7=  fMA(kr_7);
  f_alpha_f_8= fMA(kf_8);  f_alpha_r_8=  fMA(kr_8);
  f_alpha_f_9= fMA(kf_9);  f_alpha_r_9=  fMA(kr_9);
  f_alpha_f_10= fMA(kf_10);  f_alpha_r_10=  fMA(kr_10);
  f_alpha_f_11= fMA(kf_11);  f_alpha_r_11=  fMA(kr_11);
  f_alpha_f_12= fMA(kf_12);  f_alpha_r_12=  fMA(kr_12);
  f_alpha_f_13= fMA(kf_13);  f_alpha_r_13=  fMA(kr_13);
  f_alpha_f_14= fMA(kf_14);  f_alpha_r_14=  fMA(kr_14);
  f_alpha_f_15= fMA(kf_15);  f_alpha_r_15=  fMA(kr_15);
  f_alpha_f_16= fMA(kf_16);  f_alpha_r_16=  fMA(kr_16);
]

\\Species components
B0 = (alpha_f_0,1)>>B0 + (alpha_r_0,1)<<B0  + (alpha_f_5,1)>> B0 +
     (alpha_r_5,1)<<B0;
B1 = (alpha_f_0,1)<<B1 + (alpha_r_0,1)>>B1 + (alpha_f_6,1)>>B1 +
     (alpha_r_6,1)>>B1+ (alpha_f_1,1)<<B1 + (alpha_r_1,1)<<B1;
B2 = (alpha_f_2,1)>>B2 + (alpha_r_2,1)<<B2 + (alpha_f_1,1)<<B2 +
     (alpha_r_1,1)>> B2;
A0 = (alpha_f_5,1)<<A0 + (alpha_r_5,1)>>A0 + (alpha_f_3,1)>>A0 +
     (alpha_r_3,1)<<A0 + (alpha_r_9,1)>>A0 + (alpha_f_9,1)<<A0;
A1 = (alpha_f_3,1)<<A1 + (alpha_r_3,1)>>A1 + (alpha_f_4,1)>>A1 +
     (alpha_r_4,1)<<A1 + (alpha_f_6,1)<<A1 + (alpha_r_6,1)>>A1 +
     (alpha_r_10,1)>>A1 + (alpha_f_10,1)<<A1;
A2 = (alpha_f_2,1)<<A2 + (alpha_r_2,1)>>A2 + (alpha_f_4,1)<<A2 +
     (alpha_r_4,1)>>A2 + (alpha_f_11,1)>>A2 + (alpha_r_11,1)<<A2;
I0 = (alpha_f_7,1)>>I0 + (alpha_r_7,1)<<I0 + (alpha_f_9,1)<<I0 +
     (alpha_r_9,1)>>I0 + (alpha_f_14,1)>>I0 + (alpha_r_14,1)<<I0;
I1 = (alpha_f_7,1)<<I1 + (alpha_r_7,1)>>I1 + (alpha_f_8,1)>>I1 +
     (alpha_r_8,1)>>I1 + (alpha_f_10,1)<<I1 + (alpha_r_10,1)<<I1 +
     (alpha_r_15,1)>>I1 + (alpha_f_15,1)<<I1;
I2 = (alpha_f_8,1)<<I2 + (alpha_r_8,1)>>I2 + (alpha_f_11,1)<<I2 +
     (alpha_r_11,1)>>I2 + (alpha_r_16,1)>>I2 + (alpha_f_16,1)<<I2;
D0 = (alpha_f_12,1)>>D0 + (alpha_r_12,1)<<D0 + (alpha_f_14,1)<<D0 +
     (alpha_r_14,1)<<D0;
D1 = (alpha_f_12,1)<<D1 + (alpha_r_12,1)>>D1 + (alpha_f_13,1)>>D1 +
     (alpha_r_13,1)>>D1 + (alpha_f_15,1)<<D1 + (alpha_r_15,1)<<D1;
D2 = (alpha_f_13,1)<<D2 + (alpha_r_13,1)>>D2 + (alpha_f_16,1)<<D2 +
     (alpha_r_16,1)>>D2;

\\Model components
```

```
B0(1.66e-5) <kf_0,kr_0> B1(0) <kf_1,kr_1> B2(0) <kf_5,kr_5>
A0(0) <kf_3,kr_3,kf_6,kr_6> A1(0) <kf_4,kr_4,kf_2,kr_2> A2(0) <kf_9,kr_9>
I0(0) <kf_7,kr_7> I1(0) <kf_8,kr_8,kf_10,kr_10> I2(0)<kf_14,kr_14>
D0(0) <kf_12,kr_12,kf_15,kr_15> D1(0) <kf_13,kr_13,kf_16,kr_16> D2(0)

\\Event
[(event_1, t=20, kf_0=0; kf_1=0; kf_3 =0 kf_4 = 0;
  kf_7=0; kf_8=0 ; kf_12=0; kf_{13}=0,0)]
```

References

1. R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappa, H. Rubin and J. Schug. Hybrid modeling and simulation of biomolecular networks. In Proc. of *4th International Workshop on Hybrid Systems: Computation and Control*, volume LNCS 2034, pages 19-32, 2001.
2. R. Alur, R. Grosu, Y. Hur, V. Kumar, I. Lee. Modular Specification of Hybrid Systems in CHARON. In Proceedings of the 3rd International Workshop on Hybrid Systems: Computation and Control, 2000.
3. A. Bockmayr and A. Courtois. Using hybrid concurrent constraint programming to model dynamic biological systems. In Proc. of the 18th *International Conference on Logic Programming*, volume 2401, Springer-Verlag, 2002.
4. L. Bortolussi and A. Policriti. Hybrid Approximation of Stochastic Concurrent Constraint Programming, Constraints, Volume 13, Iussue 1-2, pages 66–90, 2008.
5. L. Bortolussi and A. Policriti. Hybrid Systems and Biology. Continuous and Discrete Modeling for Systems Biology. Chapter for the tutorial of SFM-08:Bio, LNCS, volume 5016, 2008.
6. M. Calder, S. Gilmore and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways, Proc. of *CMSB'05*, pages 204–215, 2005.
7. M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *T. Comp. Sys. Biology*, VII, volume 4230 of LNCS, pages 1–23, Springer, 2006.
8. F. Ciocchetta, and J. Hillston. Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. Proc. of *FBTC 2007*, volume 194/3 of ENTCS, pages 103–117, 2008.
9. F. Ciocchetta, and J. Hillston. Bio-PEPA: a framework for the modelling and analysis of biological systems. Submitted to Theoretical Computer Science, 2008 **(UPDATE)**.
10. B.A. Cota and R. B. Sargent. Simultaneous events and distributed simulation. Proc. of the Winter Simulation Conference, 1990.
11. L. Dematté, C. Priami and A. Romanel. The BlenX Language: a tutorial. Chapter for the tutorial of SFM-08:Bio, LNCS, volume 5016, 2008.
12. A. Deshpande, A. Gll, L. Semenzato. SHIFT Programming Language and Run-Time System for Dynamic Networks of Hybrid Automata. PATH Report, available at http://path.berkeley.edu/SHIFT/publications.html.
13. S.J. Edelstein, O. Schaad, E. Henry, D. Bertrand and J.P. Changgeux. A kinetic mechanism for nicotin acetylcholine receptors based on multiple allosteric transitions. *Biol. Cybern.*, 75, pages 361–379, 1996.
14. V. Galpin, J. Hillston, and L. Bortolussi. HYPE applied to the modelling of hybrid biological systems. Electronic Notes in Theoretical Computer Science (Proceedings of MFPS 24), To appear, 2008. **(UPDATE)**.
15. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, volume 81, pages 2340–2361, 1977.

16. T.A. Henzinger. The Theory of Hybrid Automata. In the proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS'96, 1996.
17. T.A. Henzinger, P.-H. Ho and H. Wong-Toi. HyTech: A Model Checker for Hybrid Systems. *Software Tools for Technology Transfer*, volume 1, pages 110–122, 1997.
18. HybridSal home page, http://sal.csl.sri.com/hybridsal/.
19. M. Hucka, A. Finney, S. Hoops, S. Keating and Le Novére. Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions. Available at http://sbml.org/documents/.
20. P. Lincoln and A. Tiwari. Symbolic systems biology: Hybrid modeling and analysis of biological networks. In proc. *Hybrid Systems: Computation and Control 7th Intl. Workshop*, LNCS 2993, pages 660–672, 2004.
21. N. Le Novére, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J.L. Snoep, and M. Hucka. BioModels Database: a Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. *Nucleic Acids Research*, volume 34, pages D689–D691, 2006.
22. C. Priami and P. Quaglia. Beta-binders for biological interactions. Proc. of *CMSB'04*, Volume 3082 of LNCS, pages 20–33, Springer, 2005.
23. Prism web site. http://www.prismmodelchecker.org/
24. C. Priami, A. Regev, W. Silverman and E. Shapiro. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, volume 80, pages 25–31, 2001.
25. A. Romanel, L. Dematté and C. Priami. The Beta Workbench. *Technical report TR-03-2007, Microsoft Research-University of Trento Centre for Computational and Systems Biology.*