Bio-PEPA with 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 biochemical systems, to handle *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 in order to keep the specification of the model as straightforward as possible. Some maps are defined from Bio-PEPA with events to analysis tools. Specifically, we map our language to *Hybrid Automata (HA)* and we consider a modification of Gillespie's algorithm for stochastic simulation. In order to test our approach, we present the translation in Bio-PEPA of a biochemical network describing the functional properties of the Acetylcholine receptor with the addition of an event that causes the inactivation of some reactions at a given time.

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 (e.g. [27,25,8,9]). In most cases the analysis is performed using Gillespie's stochastic simulation algorithm [18]. Other possibilities exist, such as the mapping to differential equations [7].

Many biological models need to capture both discrete and continuous phenomena [1,4,23]. These models are called *hybrid systems*. A first example of a hybrid system describes 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. Other examples describe some experiments, where it may be necessary to render the possible change to the system, due, for instance, to the introduction or the removal of some reagents.

In this work we present an extension of Bio-PEPA [9,10], 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 [22]. These kinds of

C. Priami et al. (Eds.): Trans. on Comput. Syst. Biol. XI, LNBI 5750, pp. 45–68, 2009.

events can be found in biochemical networks, such as the ones in the BioModels database [24] or defined in some experimental settings.

The idea underlying our work is the following:
Biochemical networks with events $\Longrightarrow$ *Bio-PEPA with events* $\Longrightarrow$ *Analysis*

Starting from a biochemical network with one or more events, we want to map it into a Bio-PEPA system. From that, we can then consider different kinds of analysis. In this, view Bio-PEPA is a formal, intermediate, compositional representation of the biochemical network. This idea is the one proposed for (the standard) Bio-PEPA.

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, we keep the specification of the model as simple as possible. Second, this approach is appropriate when we study the same biochemical system but with different experimental regimes as 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)* [19]. 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 [18] in order to tackle events.

A preliminary version of this work has been presented in [11]. Here we add some definitions concerning the kind of events and further details concerning the mappings from Bio-PEPA with events to the Hybrid Automata and Gillespie's algorithm. Furthermore, we consider more general kinds of events, such as simultaneous events or events with a delay different from zero.

The rest of the paper is organised as follows. Section 2 reports a description of Bio-PEPA. In Section 3 we define the events we are considering in this work and then we extend Bio-PEPA in order to handle them. Section 4 describes the mapping from our language to Hybrid Automata. The mapping to stochastic simulation is reported in Section 5. After that, Section 6 illustrates the modelling in Bio-PEPA of a biochemical network describing the functional properties of the Acetylcholine receptor with an event that is triggered at a given time and causes the inactivation of some reactions. In Section 7 we overview some related work. Finally, in Section 8, some conclusions are reported.

2 Bio-PEPA

Bio-PEPA [9,10] is a language for the modelling and analysis of biochemical networks. The syntax of Bio-PEPA is defined as:

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

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

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 (α, κ) $\texttt{op}$ 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 $C \overset{def}{=} S$. The parameter $x \in \mathbb{R}^+$ in $S(x)$ represents the initial quantity (for instance the concentration) of the species. Finally, the process $P \underset{\mathcal{I}}{\bowtie} Q$ denotes the cooperation between components: the set $\mathcal{I}$ 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 discretized 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 $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Compon-ents, P \rangle$, 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 [9,10].

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 define two relations. The former, called *capability relation*, is indicated by $\xrightarrow{\theta}_c$. The label θ is of the form (α, w), where $w := [S : op(l, \kappa)] \mid w :: w$, with S a species component, op a symbol representing the role of the species in the reaction, l the level and κ the stoichiometry coefficient. This relation is defined as the minimum relation satisfying the rules reported in Table 1.

Table 1. Axioms and rules for Bio-PEPA

`prefixReac`	$((\alpha,\kappa)\downarrow S)(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])}_c S(l-\kappa) \quad \kappa \le l \le N$
`prefixProd`	$((\alpha,\kappa)\uparrow S)(l) \xrightarrow{(\alpha,[S:\uparrow(l,\kappa)])}_c S(l+\kappa) \quad 0 \le l \le (N-\kappa)$
`prefixMod`	$((\alpha,\kappa)\, op\, S)(l) \xrightarrow{(\alpha,[S:op(l,\kappa)])}_c S(l)$ with $op = \odot, \oplus, \ominus$ and $0 < l \le N$ if $op = \oplus$, $0 \le l \le N$ otherwise
`choice1`	$\dfrac{S_1(l) \xrightarrow{(\alpha,w)}_c S_1'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)}_c S_1'(l')}$
`choice2`	$\dfrac{S_2(l) \xrightarrow{(\alpha,w)}_c S_2'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)}_c S_2'(l')}$
`constant`	$\dfrac{S(l) \xrightarrow{(\alpha,S:[op(l,\kappa)])}_c S'(l')}{C(l) \xrightarrow{(\alpha,C:[op(l,\kappa)])}_c S'(l')}$ with $C \overset{def}{=} S$
`coop1`	$\dfrac{P_1 \xrightarrow{(\alpha,w)}_c P_1'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2}$ with $\alpha \notin \mathcal{L}$
`coop2`	$\dfrac{P_2 \xrightarrow{(\alpha,w)}_c P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w)}_c P_1 \underset{\mathcal{L}}{\bowtie} P_2'}$ with $\alpha \notin \mathcal{L}$
`coop3`	$\dfrac{P_1 \xrightarrow{(\alpha,w_1)}_c P_1' \quad P_2 \xrightarrow{(\alpha,w_2)}_c P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,w_1::w_2)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2'}$ with $\alpha \in \mathcal{L}$

The latter relation, called *stochastic relation*, is $\longrightarrow_s \subseteq \tilde{\mathcal{P}} \times \Gamma \times \tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}}$ is the set of well-defined Bio-PEPA systems[1] and Γ is the set of labels $\gamma = (\alpha, r)$, with α the action type and r the associated rate. This relation is defined as the minimal relation satisfying the rule:

[1] In a *well-defined* Bio-PEPA system each element has to satisfy some conditions. For instance, we have that each species component $C \in \mathit{Comp}$ must have subterms of the form "(α,κ) `op` C" and the action types in each single component must be all distinct. Furthermore, the model component P must be defined in terms of the species components defined in Comp and, for each cooperation set $\mathcal{L}_j$ in P, $\mathcal{L}_j \subseteq \mathcal{A}(P)$. For details see [12].

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

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 the *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 with levels, stochastic simulation and PRISM [26]. Some tools for the analysis of Bio-PEPA system have been implemented [3]. 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 further details and the other mappings see [10].

2.1 From Bio-PEPA to ODE System (π_{ODE})

Let π_{ODE} be the mapping from Bio-PEPA system to the associated ODE system.

The mapping π_{ODE} entails three steps:

1. definition of the stoichiometry $(n \times m)$ matrix $\mathbf{D}$, where n is the number of species and m is the number of reactions;
2. definition of the *kinetic 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 $\mathbf{D} = \{d_{ij}\}$. 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 from the functional rates and its definition is straightforward.

The ODE system thus obtained has the form:

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

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

2.2 Example

In order to show how to model biochemical systems in Bio-PEPA we consider the network presented in Fig. 1 and we translate it into Bio-PEPA. This network is then used as a running example in the rest of the paper.

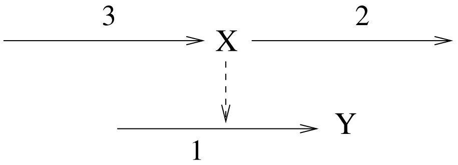

Fig. 1. Biochemical network composed of two proteins X and Y. The numbers indicate the reactions. Reaction 1 is the translation of Y enhanced by X, reaction 2 is the degradation of X and reaction 3 the translation of X.

The network is composed of two proteins, X and Y. These are involved in the following interactions:

- Translation of Y enhanced by X (reaction 1): $X \xrightarrow{r_1} X + Y$.
  The kinetic law is mass-action with constant parameter $r_1 = 0.01$;
- Degradation of the protein X (reaction 2): $X \xrightarrow{r_2} \emptyset$.
  The kinetic law is mass-action with constant parameter $r_2 = 0.02$;
- Translation of the protein X (reaction 3): $\emptyset \xrightarrow{r_3} X$.
  The kinetic law is mass-action with constant parameter $r_3 = 0.01$.

Each reaction i is represented by an action type α_i. The kinetic laws are represented by the following functional rates:

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

where $fMA(r)$ stands for mass-action kinetic law with rate r.

The Bio-PEPA species components[2] corresponding to the two proteins are:

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

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

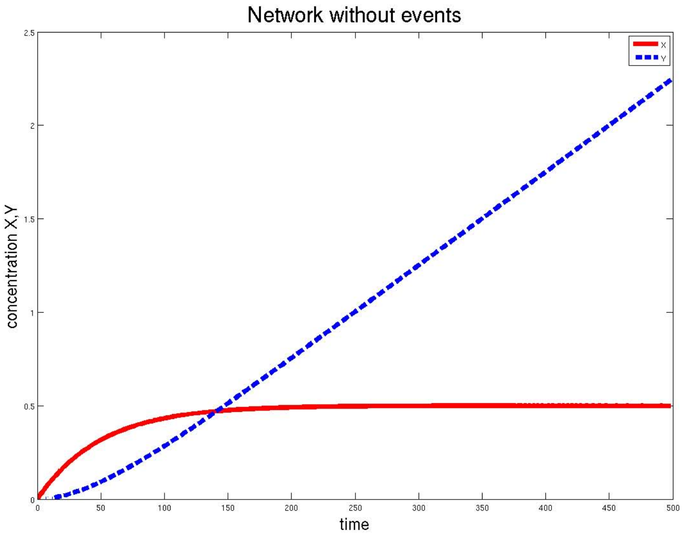

Fig. 2. ODE integration results for the network

whereas the model component is:

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

where the initial values are zero for both the proteins.

The set of compartments and the set $\mathcal{N}$ are not reported.

Applying the mapping π_{ODE} we obtain the ODE system:

$$\begin{aligned} \frac{dx}{dt} &= -0.02 \cdot x + 0.01 \\ \frac{dy}{dt} &= 0.01 \cdot x \end{aligned}$$

where x and y are the two variables describing X and Y. The result of ODE integration is reported in Fig. 2. The protein X reaches a steady-state whereas Y increases infinitely.

3 Bio-PEPA with Events

3.1 SBML-Like Events: Some Definitions

In this work we consider events as defined in the SBML specification [22]. 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. An SBML-like event is *immediate* if *delay* is equal to zero. Otherwise, the event is called *delayed*.

The definition of sequential and simultaneous events is reported below.

Definition 2. *Two or more SBML-like events are sequential if they are fired one after the other in a given order. They are said to be simultaneous if they happen at the same point in time.*

In most biochemical systems which we are interested in we have sequential events. In the general situation of simultaneous events, sometimes some *tie-breaking* rules are necessary to decide which of any set of events is simulated first. The most common way to do this is to assign a *priority* to each event [13]: when there are two or more simultaneous events, the event with the highest priority is defined to be the next event to fire. However, the order in which a set of simultaneous events is fired is not always important, for instance when the assignments of the events influence different variables. We have the following definition:

Definition 3. *Two simultaneous events are independent if their event assignments do not effect each other. Otherwise, they are called dependent.*

If we have simultaneous independent 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 independent events are dealt with similarly to sequential ones.

3.2 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), compartments (size), parameters (values) and functional rates (function definitions).
2. Triggers are deterministic, i.e. when they become true they are fired.
3. Triggers are only unidirectional, i.e. describing the change from one mode to another, but not vice versa. Bidirectional triggers can be decomposed into two unidirectional triggers.
4. Events are either sequential or simultaneous and independent.

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

3.3 The Definition of the Language

We can add events to a Bio-PEPA system 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 the Bio-PEPA model and time, *event_assignment* is a list of assignments, *delay* is 0 (*immediate events*) or positive real value (*delayed events*). Formally, we have the following definitions:

$$
\begin{array}{l}
\text{trigger} \;::=\; \text{cond} \mid \text{cond } \textbf{or} \text{ cond} \mid \text{cond } \textbf{and} \text{ cond} \mid \textbf{not} \text{ cond} \\
\text{cond} \;::=\; t \text{ eq value} \mid \mathit{expression}(\bar{C}, \bar{k}) \text{ eq value} \mid \\
\qquad\qquad \mathit{expression}(\bar{C}, \bar{k}) \text{ eq } \mathit{expression}(\bar{C}, \bar{k}) \\
\text{eq} \;::=\; = \mid \neq \mid > \mid < \mid \leq \mid \geq \\
\text{delay} \;::=\; \text{value} \\
\text{event\_assignment} \;::=\; \text{assignment ; event\_assignment} \\
\text{assignment} \;::=\; \text{k} \leftarrow \text{value} \mid C \leftarrow \text{value} \mid f_\alpha \leftarrow \mathit{expression}(\bar{C}, \bar{k}) \\
\qquad\qquad\qquad V \leftarrow \text{value} \mid \text{t} \leftarrow \text{value} \\
\text{event} \;::=\; \text{(id, trigger, event\_assignment, delay)}
\end{array}
$$

where C stands for any species component, k for any parameter and V for any compartment, f_α is the functional rate associated with the action type α, the variable $t \in \mathbb{R}^+$ represents the global time of the system, $\mathit{expression}(\bar{C}, \bar{k})$ 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. Note that in the assignment definition, C indicates the concentration of the associated species component and V the size of the associate compartment. The assignment involving time is just auxiliary to express delayed events (see Sec. 4.2).

The set of events is then defined as:

$$Events \;::=\; [\,] \;\mid\; event :: Events$$

Definition 4. *A Bio-PEPA system with events $\mathcal{P}$ is an 8-tuple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}, Comp, P, Events, t \rangle$, where Events is the set of events, $t \in \mathbb{R}^+$ is the variable expressing time and the other elements are as in 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 [10].

Definition 5. *The set 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, compartments, 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.

3.4 Example (Continued)

Consider the simple network described in Sect. 2.2. We can assume that the translation of the protein X is possible only when the concentration of Y is less than 0.8. This is expressed in Bio-PEPA by the following immediate event involving concentrations:

$$(event_1,\ Y = 0.8,\ r_3 \leftarrow 0,\ 0)$$

where r_3 is the constant rate for the translation of protein X. When the concentration of Y reaches the value 0.8, the value of r_3 becomes 0 and therefore the creation of X is not possible anymore.

In addition to this events, we can assume that when the concentration of X is less or equal than a given value (0.2, for instance) and the concentration of Y greater than 0.8 the creation of X is enabled again but with a smaller rate than before ($r_3 = 0.005$).

This is expressed in Bio-PEPA by the following immediate event:

$$(event_2,\ Y > 0.8 \textbf{ and } X = 0.2,\ r_3 \leftarrow 0.005,\ 0)$$

These two events are sequential and clearly satisfy the assumptions discussed above.

4 Mapping to Hybrid Automata

4.1 Hybrid Automata

Hybrid automata (HA) [19] 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-valued 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* [15]) and model checking (see *HyTech* [20]). In this work we limit our attention to simulation. For a formal definition and details of the formalism see [19].

4.2 Definition of the Mapping

Here we present the map from Bio-PEPA 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 $\mathcal{P}_0 = \langle \mathcal{V}_0, \mathcal{N}_0, \mathcal{K}_0, \mathcal{F}_0, Comp_0, P_0, Events, t_0 \rangle$ 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 $\{X_1, X_2, \cdots, X_{N_{Comp}}, t\}$, 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 independent 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 assignment 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 $N_{jump} + 1$. σ is used to indicate a mode and the Σ the set of all modes. 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 modification 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 assignments are executed. This information cannot be directly translated in any of the components of standard HA. In the following we report as we handle the delay in HA.

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.

Second, 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. $(id_2, t = t_{mode} + delay, event\_assignment, 0)$.

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

4.3 Example (Continued 2)

Consider the network presented in Sect. 2.2 with the addition of the set of events (see Sect. 3.4):

$$[(event_1,\ Y = 0.8,\ r_3 \leftarrow 0,\ 0)].$$

A schema of the HA associated with this network is reported in Fig. 3. The set of variables is $\{x,\ y,\ t\}$, where x and y are the two variables representing the two proteins. The initial concentrations, derived from the initial condition in the Bio-PEPA model, are $x = 0$, $y = 0$ and $t = 0$. We have just one event so we have two modes and the jump condition (guard) is $y = 0.8$. The former mode is described by the invariant condition $y < 0.8$ and the latter by $y \geq 0.8$.

The ODE system corresponding to the initial mode $(S1)$ is derived by applying the mapping π_{ODE} to the initial Bio-PEPA system $(\mathcal{P}_0)$ and is:

$$\begin{aligned} \tfrac{dx}{dt} &= -0.02 \cdot x\ +\ 0.01 \\ \tfrac{dy}{dt} &= 0.01 \cdot x \end{aligned}$$

For the second mode, the ODE system $(S2)$ is obtained as $\pi_{ODE}(\mathcal{P}_0[r_3 \leftarrow 0])$ and is:

$$\begin{aligned} \tfrac{dx}{dt} &= -0.02 \cdot x \\ \tfrac{dy}{dt} &= 0.01 \cdot x \end{aligned}$$

If we consider both $event_1$ and $event_2$, we have the HA represented in Fig. 4. There are three modes, representing the network at the initial state, when $y < 0.8$ and when $y \geq 0.8$ and $x \geq 0.2$. Two jumps conditions are defined in terms of the trigger conditions.

The ODE systems describing the first and second modes are as above, whereas the ODE system for the third mode $(S3)$ is obtained from $\pi_{ODE}(\mathcal{P}_1[r_3 \leftarrow 0.005])$ (where $\mathcal{P}_1$ is the Bio-PEPA system corresponding to the second mode) and is:

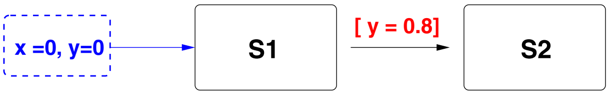

Fig. 3. HA representation for the network composed of the two proteins X and Y and with an event involving concentrations

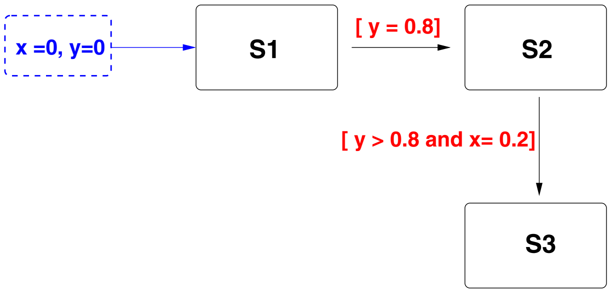

Fig. 4. HA representation for the network composed of the two proteins X and Y and with two sequential events

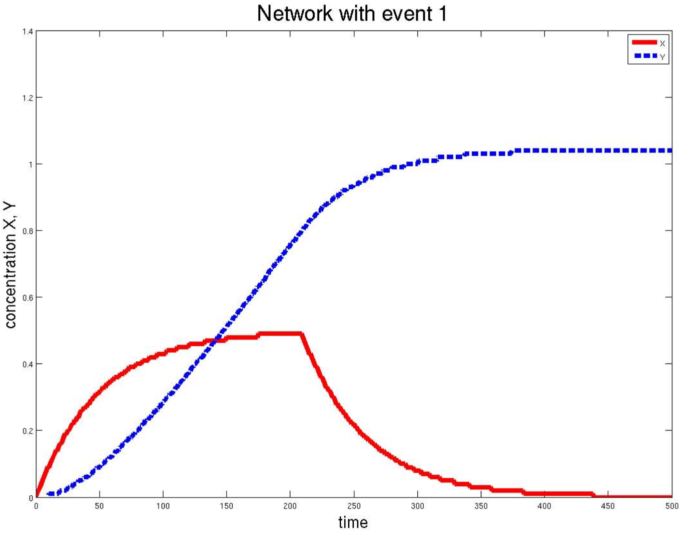

Fig. 5. Simulation results for the network composed of the two proteins X and Y and with the addition of $event_1$

$$\begin{aligned}\frac{dx}{dt} &= -0.02 \cdot x \; + \; 0.005 \\ \frac{dy}{dt} &= 0.01 \cdot x\end{aligned}$$

Some results for the network with just $event_1$ are reported in Fig. 5. The protein X increases until time 200 s when Y reaches the value 0.8 and then decreases to 0. The protein Y increases, but after the event, its rate of increase is much lower than the case without the event.

The results for the network with both $event_1$ and $event_2$ is reported in Fig. 6. In this case, when the second event is fired, the protein X starts to increase again and this has effect on the production of Y as well.

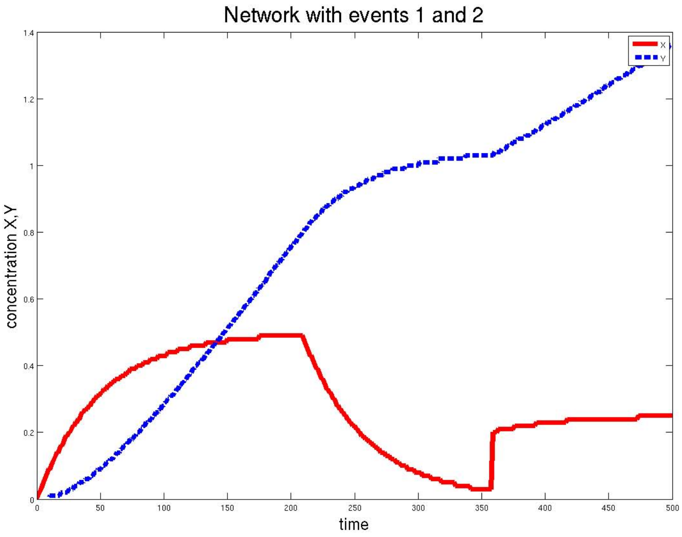

Fig. 6. Simulation results for the network composed of the two proteins X and Y and with the addition of $event_1$ and $event_2$

5 Stochastic Simulation by Gillespie's Algorithm

One of the possible kinds of analysis supported by Bio-PEPA is stochastic simulation using Gillespie's algorithm [10]. 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 in its initial conditions. We assume that initially all the triggers evaluate to false. When one of the conditions is satisfied, the simulation stops and the system is modified 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. The use of triggers involving time can be challenging since it can happen that the time of the event does not coincide with any of the simulation time points. Our approach to deal with this case is discussed below.

Note that if the events involve species concentrations, we have to change concentrations into number of molecules for stochastic simulation. Specifically, we have to multiply each concentration by $Na\,V$, where Na is the Avogadro number[3] and V is the volume of the compartment. In the the rest of this section we assume that the events are in terms of number of molecules.

[3] This is the number of "entities" (atoms or molecules) in one mole of substance. Its value is $6.022 \times e{+}23\ (mol)^{-1}$.

We propose the following procedure for each simulation run.

1. Let $\mathcal{P}_0$ be the initial Bio-PEPA system and t_s the maximum simulation time.
2. While $t < t_s$ and $trigger_i = false$ for $i = 1, 2, ..., N_{Events}$, simulate.
3. If $t \geq t_s$ then stop.
4. If $t < t_s$ and there exists a $trigger_i$ such that it is true, we have that:
   (a) if $delay = 0$ modify 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 until 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 independent simultaneous events we proceed as observed in Section 3.1: we can abstract these events as a single event, whose trigger is defined in terms of the triggers of the two events and the event assignments are the union of the assignments. Therefore, we modify the system according to the assignments associated with all the events involved.
- When we have an event with a trigger involving time $t = \tilde{t}$, the time value $\tilde{t}$ may not correspond to any of the simulation time point obtained by using Gillespie's simulation algorithm. Specifically, there exist two consecutive simulation time points t_j and t_{j+1} such that $t_j < \tilde{t} < t_{j+1}$. If this happens, we have to decide when the system has to be modified. In order to handle this situation we consider the following approach:

  1. if $t_j < \tilde{t} + delay < t_{j+1}$ with $delay \geq 0$ consider the system at time $\tilde{t} + delay$ and modify it at that time point. The simulation restarts from $\tilde{t} + delay$.
  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 run the simulation until t_h. Then, modify the system at time $\tilde{t} + delay$ and restart the simulation from that time point.

6 The Acetylcholine Receptor Model

This example concerns the functional properties of the *nicotin Acetylcholine Receptors (nAChR)*. These are transmembrane proteins that mediate interconversions between open and closed channel states under the control of neurotransmitters. The detailed description of the model is reported in [16].

A schema of the model is shown in Figure 7. 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 are the number of ligands (denoted X) bound to a receptor. In the model the ligands are not modelled explicitly. 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

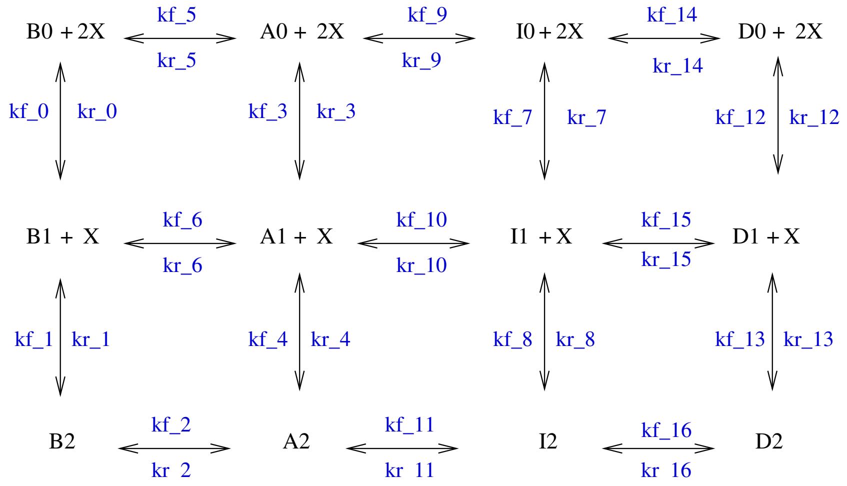

Fig. 7. Schema of the Acetylcholine receptor model

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

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				

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 direction kr_i.

In addition to these elements, 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 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 $kf_0 \leftarrow 0,\ kf_1 \leftarrow 0,\ kf_3 \leftarrow 0,\ kf_4 \leftarrow 0,\ kf_7 \leftarrow 0\ , kf_8 \leftarrow 0\ , kf_{12} \leftarrow 0\ , kf_{13} \leftarrow 0$.

The Bio-PEPA system associated with the Acetylcholine receptor model. In the following we report briefly the definition of the Bio-PEPA system representing the Acetylcholine receptor 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 is $1.66e$-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 $1.66e$-5 and coincides with the initial concentration of channels at the basal state. Note that the information about the step size and the number of levels is not used in this work, as we do not consider CTMC with levels, however we define them for completeness.
- *Definition of functional rates ($\mathcal{F}_R$) and parameters ($\mathcal{K}$).* Each reversible reaction i, $i = 0, 1, 2, \cdots, 16$, is decomposed in two irreversible reactions, f_i and r_i, representing the forward and inverse directions respectively. The associated kinetic laws are $f_{\alpha\_f_i} = fMA(kf\_i)$; and $f_{\alpha\_r_i} = fMA(kr\_i)$, where fMA denotes mass-action. All the parameters are defined in the set $\mathcal{K}$. The values are the ones reported in the paper [16].
- *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.

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

  The system is described as:

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

  where L_i, $i = 1, ..., 10$ are the cooperation sets and the initial values for the species are 0 with the exception of the species $B0$.
- *Definition of events.* We have only one event, describing a change in the system at time 20 s:

  $$\begin{array}{r} [(event, t = 20,\ kf_0 \leftarrow 0;\ kf_1 \leftarrow 0;\ kf_3 \leftarrow 0\ kf_4 \leftarrow 0;\ kf_7 \leftarrow 0;\ kf_8 \leftarrow 0; \\ kf_{12} \leftarrow 0;\ kf_{13} \leftarrow 0,\ 0)] \end{array}$$

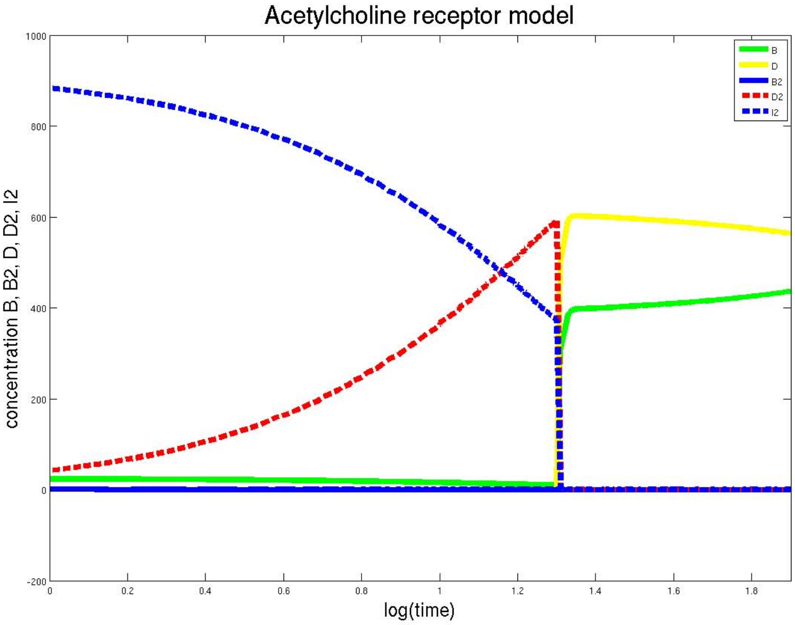

Fig. 8. Stochastic simulation results for the Acetylcholine receptor model (average over 100 runs)

Analysis results. The HA associated with the Acetylcholine receptor model is similar to the one for the network presented in Sect.2.2 with the addition of the set of events. We have two modes, described by two different sets of differential equations. The trigger condition involves time and is "$t = 20\ s$". The details of the two systems describing each mode are not reported.

Simulation results made using Gillespie's algorithm are reported in Fig. 8. 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. 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 [16]. 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.

7 Related Works

The use of mathematical formalisms in order to represent discrete changes in biological systems is not new [1,4,23,17,5,6]. 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 using *CHARON* [2], a language that allows formal description of hybrid systems. The authors of [23] discussed the use of discrete changes in biological systems and presented some examples using the formalism *HybridSAL* [21]. *Hybrid Concurrent Constraint Programming* is used to model some biological systems with both discrete and continuous changes in [4]. In [5] 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. A discussion of hybrid systems and biology is reported in [6]. Finally, in [17] the authors presented *HYPE*, a process algebra for the modelling of hybrid systems and used it to represent the *repressilator*, an artificial genetic network composed of three genes and their respective proteins with oscillatory behaviour. In none of these works are SBML-like events considered explicitly, but the focus is on general hybrid systems.

Events have been proposed in the *Beta Workbench (BetaWB)* [14] and in the associated programming language *BlenX* [28]. In BlenX 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 (*cond*) and an action (*verb*) and is associated with a rate. Conditions can involve number of entities, the simulation time or the simulation step. The possible actions are the join of two entities, the split of one entity into two, the update of a variable of the system and the deletion or the creation of a new entity.

The concept of events proposed in BlenX is quite similar to the one considered for Bio-PEPA. The BlenX condition and action correspond to the trigger and event assignment in Bio-PEPA events. However, rates in BlenX have a different meaning from the delay in Bio-PEPA. Indeed, in Bio-PEPA an events occurs when the trigger is satisfied and the role of the delay is to postpone when the event is executed. BlenX events with a finite rate can happen only when the trigger is satisfied but it is in competition with other actions that are enabled contemporaneously (race condition). BlenX events with infinite rate correspond to immediate events in Bio-PEPA.

In order to compare the definition of events in the two languages, we show how the events proposed in this paper can be described in BlenX. The event $event_1$ defined in Sec. 3.4 is represented in BlenX as:

$$\text{when } \ (Y \rightarrow value) \ \text{ update } \ (r_3, change\_par)$$

where *value* is $0.8 \cdot NaV$ molecules and the function *change_par* is defined as $change\_par \ : \ function = 0$. The operator "→" recognizes when the quantity bound to Y becomes greater than the specified value, whereas the action " update $(r_3, change\_par)$" means that the parameter r_3 is updated according to the function *change_par* (in our case it assigns the value 0). The rate associated with the update action is always infinite and not reported. Concerning the event in the Acetylcholine receptor model, it is not possible to represent this event in BlenX as conditions involving time are not allowed with the action update.

Note that BlenX events represent more general kinds of interactions than Bio-PEPA events. For instance, they are used to model the formation of a complex (by using the action join) or the split of a complex into two parts (by using the split action). These reactions (as all the other kinds) are represented in Bio-PEPA by synchronization of the species components over the action types abstracting the reactions. Bio-PEPA events have been introduced specifically to represent experimental situations when there is change in the system due to some conditions.

8 Conclusions

In this work we have presented 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.

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, Vashti Galpin and Adam Duguid for their helpful comments. The author is supported by the EPSRC under the CODA project "Process Algebra Approaches for Collective Dynamics" (EP/c54370x/01).

References

1. Alur, R., Belta, C., Ivancic, F., Kumar, V., Mintz, M., Pappa, G., Rubin, H., Schug, J.: Hybrid modeling and simulation of biomolecular networks. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 19–32. Springer, Heidelberg (2001)
2. Alur, R., Grosu, R., Hur, Y., Kumar, V., Lee, I.: Modular Specification of Hybrid Systems in CHARON. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, p. 6. Springer, Heidelberg (2000)
3. Bio-PEPA Workbench Home Page, `http://www.dcs.ed.ac.uk/home/stg/software/biopepa/`

4. Bockmayr, A., Courtois, A.: Using hybrid concurrent constraint programming to model dynamic biological systems. In: Stuckey, P.J. (ed.) ICLP 2002. LNCS, vol. 2401, p. 85. Springer, Heidelberg (2002)
5. Bortolussi, L., Policriti, A.: Hybrid Approximation of Stochastic Concurrent Constraint Programming. Constraints 13(1-2), 66–90 (2008)
6. Bortolussi, L., Policriti, A.: Hybrid Systems and Biology. Continuous and Discrete Modeling for Systems Biology. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 424–448. Springer, Heidelberg (2008)
7. Calder, M., Gilmore, S., Hillston, J.: Automatically deriving ODEs from process algebra models of signalling pathways. In: Proc. of CMSB 2005, pp. 204–215 (2005)
8. Calder, M., Gilmore, S., Hillston, J.: Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In: Priami, C., Ingólfsdóttir, A., Mishra, B., Riis Nielson, H. (eds.) Transactions on Computational Systems Biology VII. LNCS (LNBI), vol. 4230, pp. 1–23. Springer, Heidelberg (2006)
9. Ciocchetta, F., Hillston, J.: Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. In: Proc. of FBTC 2007. ENTCS, vol. 194(3), pp. 103–117 (2008)
10. Ciocchetta, F., Hillston, J.: Bio-PEPA: a framework for the modelling and analysis of biological systems. Theoretical Computer Science (to appear)
11. Ciocchetta, F.: Bio-PEPA with SBML-like Events. In: Proc. of the Workshop Computational Models for Cell Processes, TUCS general publication, vol. 47 (2008)
12. Ciocchetta, F., Hillston, J.: Bio-PEPA: a framework for the modelling and analysis of biological systems. School of Informatics University of Edinburgh Technical Report, EDI-INF-RR-1231 (2008)
13. Cota, B.A., Sargent, R.B.: Simultaneous events and distributed simulation. In: Proc. of the Winter Simulation Conference (1990)
14. Dematté, L., Priami, C., Romanel, A.: The BlenX Language: a Tutorial. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 313–365. Springer, Heidelberg (2008)
15. Deshpande, A., Gollu, A., Semenzato, L.: SHIFT Programming Language and Run-Time System for Dynamic Networks of Hybrid Automata. PATH Report, `http://path.berkeley.edu/SHIFT/publications.html`
16. Edelstein, S.J., Schaad, O., Henry, E., Bertrand, D., Changgeux, J.P.: A kinetic mechanism for nicotin acetylcholine receptors based on multiple allosteric transitions. Biol. Cybern. 75, 361–379 (1996)
17. Galpin, V., Hillston, J., Bortolussi, L.: HYPE applied to the modelling of hybrid biological systems. ENTCS, vol. 218, pp. 33–51 (2008); Also in Proceedings of MFPS 2008
18. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry 81, 2340–2361 (1977)
19. Henzinger, T.A.: The Theory of Hybrid Automata. In: The proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS (1996)
20. Henzinger, T.A., Ho, P.-H., Wong-Toi, H.: HyTech: A Model Checker for Hybrid Systems. Software Tools for Technology Transfer 1, 110–122 (1997)
21. HybridSal home page, `http://sal.csl.sri.com/hybridsal/`
22. Hucka, M., Finney, A., Hoops, S., Keating, S., Le Novére, N.: Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions, `http://sbml.org/documents/`

23. Lincoln, P., Tiwari, A.: Symbolic systems biology: Hybrid modeling and analysis of biological networks. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 660–672. Springer, Heidelberg (2004)
24. Le Novére, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J.L., Hucka, M.: BioModels Database: a Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems. Nucleic Acids Research 34, D689–D691 (2006)
25. Priami, C., Quaglia, P.: Beta-binders for biological interactions. In: Danos, V., Schachter, V. (eds.) CMSB 2004. LNCS (LNBI), vol. 3082, pp. 20–33. Springer, Heidelberg (2005)
26. Prism web site, `http://www.prismmodelchecker.org/`
27. Priami, C., Regev, A., Silverman, W., Shapiro, E.: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Information Processing Letters 80, 25–31 (2001)
28. Dematté, L., Priami, C., Romanel, A.: The Beta Workbench: a computational tool to study the dynamics of biological systems. Briefings in Bioinformatics 9(5), 437–449 (2008)

A Appendix: Bio-PEPA System for the Acetylcholine Receptor Model

In this appendix we report the specification of the whole Acetylcholine receptor model in Bio-PEPA. Note that, in the definition of the species component, we use the following notation: >> indicates a product (it corresponds to the operator ↑ in the Bio-PEPA syntax) and << indicates a reactant (it corresponds to the operator ↓). This is the syntax used in the Bio-PEPA tools [3].

```
\\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,  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)]
```