Process algebra modelling styles for biomolecular processes

Muffy Calder[1] and Jane Hillston[2]

[1] Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, Scotland.

[2] Laboratory for Foundations of Computer Science and Centre for Systems Biology at Edinburgh, The University of Edinburgh, Edinburgh EHA 9AB, Scotland

Abstract. We investigate how biomolecular processes are modelled in process algebras, focussing on chemical reactions. We consider various modelling styles and how design decisions made in the definition of the process algebra have an impact on how a modelling style can be applied. Our goal is to highlight the often implicit choices that modellers make in choosing a formalism, and illustrate, through the use of examples, how this can affect expressability as well as the type and complexity of the analysis that can be performed.

1 Introduction

Much recent research has considered the problem of providing suitable abstract models to allow biologists to construct mechanistic models to enhance understanding of biomolecular processes. Process algebras, formal modelling languages originally conceived for modelling concurrent computations, have been widely applied, most notably in the area of signalling pathways [RSS01,CGH06,TK08]. This is experimental science and we are currently evaluating the hypothesis that such formal models can add value to the mathematical analysis that is already undertaken within systems biology in terms of ordinary differential equation (ODE) models or stochastic simulations directly. In exploring this goal, even within work on process algebras, several different styles of modelling have emerged. Ultimately we hope to be able to give guidance on how to choose among these modelling styles, or on how to map molecular components and their interactions to processes, process communication and process composition. However, in the first instance we investigate how design decisions made in the definition of the language have an impact on how a modelling style can be applied, and highlight the often implicit choices that modellers make in choosing a formalism.

Recent research effort on process algebras for biomolecular processes, e.g. [CGH06,CVOG06,CH08,Car08], has focussed on defining alternative semantics, such as discrete-state (stochastic) or continuous-state (ODE) semantics. These provide important links with the work where mathematical representations are used directly and establish a valid foundation for process algebra models. Based on these semantics, analysis may be carried out by model-checking, stochastic simulation based on Gillespie's algorithm or ODE simulations. Our emphasis in this paper is different. Here we consider the forms of abstraction supported by process algebra and how the abstraction and the process algebra chosen affect the expressiveness of the model with respect to the biological processes, as well as the type and complexity of the analysis that can be performed.

We focus on one of the most important types of interaction between molecular components: chemical reactions. In chemical notation, these may be first order reactions, for example A degrades to B: $A \xrightarrow{k1} B$, or second order reactions, for example A and B combine to form C or C and D: $A + B \xrightarrow{k2} C$, or $A + B \xrightarrow{k3} C + D$. Typically, $k1 \ldots k3$ are rate constants for kinetic laws (e.g. mass action).

A fundamental aspect of the abstraction used in modelling is the nature of the process mapping. In the literature on process algebras for systems biology we find predominantly the *molecule-as-process* [RSS01,Car08] abstraction, but the *species-as-process* and *reaction-as-process* mappings have also been proposed [CGH06,CH08,BP08]. The distinction between the first two can be understood by appealing to ecology: the former is essentially individuals-based, whereas the latter is population-based. We note that this distinction is less common in distributed computing system modelling, the origins of process algebra, where population-based models are rarely considered.

Further stylistic differentiation was identified in [CGH06] where the concepts of *reagent-centric* and *pathway-centric* models are introduced, in the context of population-based modelling. Reagent-centric models map all reagents in a reaction to processes, whose variation reflect decrease through consumption and increase through product formation (consumers and producers). Reagents such as modifiers that do not vary species amounts can also be modelled in this approach. Reagent-centric models provide a fine-grained, distributed view of a system. Pathway-centric models provide a more abstract view of a system, tracking serialisations of events, which are then composed concurrently. Here, processes vary according to their biological state rather than their quantity. Whereas in a reagent-centric approach the processes may be molecules or molecular species, in the pathway-centric approach the processes are molecules or sub-pathways. Thus the interactions between processes are between flows of events corresponding to producers, i.e. components on the left hand sides of a reactions.

Most modelling approaches map *chemical reactions* to *events* in a straightforward way, and map (possibly a subset of) the *chemical components* to *processes*. Bortolussi and Policriti's work on sCCP, using the *reaction-as-process* abstraction, is an exception to this. When chemical components are mapped to processes within the reagent-centric approach there is a further choice: between associating processes with *all* components or only with the reagents on the *left hand side* of equations, i.e. those reagents that are the reactants of the reaction. To distinguish these two cases, we call the former reagent-centric and the latter *reactant-centric*. This modelling choice is often influenced by the form of synchronisation available within the algebra: binary or multi-way. If we have only the former, then only the reactant-centric approach is possible and we are left with an interesting dilemma when there are fewer components on the right hand side of the equation than on the left hand side, e.g. $A + B \xrightarrow{k2} C$.

In summary, a number of factors will influence the structure of a process algebra model of a biomolecular process:

– population-based or individuals-based,
– reagent-centric, reactant-centric, pathway-centric or reaction-centric,
– the form of synchronisation available in the algebra.

In this paper we investigate the interplay between these three factors. Our motivation is to explore the extent to which we can build clear and faithful models using current algebras and analysis techniques, and how design decisions with respect to the process algebra determine the mappings available to the modeller. We consider different combinations, investigating their advantages and disadvantages.

We will use five process algebras for illustration: π-calculus, Beta-binders, PEPA, Bio-PEPA, and sCCP; these are briefly outlined in the next section. These are chosen as they represent a spectrum of different modelling style, including languages that have been adapted (π-calculus, PEPA and sCCP) and designed (Beta-binders and Bio-PEPA) for biological modelling. This is by no means a comprehensive list of process algebras used in systems biology. In particular we do not include any of the process algebras designed to consider spatial aspects of biomolecular processes [CPR$^+$04,Car04,V07,BMMT06,CG09] as they are beyond the scope of this paper.

The remainder of the paper is organised as follows. Section 2 gives an overview of the process algebras and Section 3 describes the example pathway used throughout for illustration and comparison. In Sections 4 to 8 we consider modelling in PEPA, Bio-PEPA, π-calculus, Beta-binders and sCCP. We discuss the results in Section 9 and give our conclusions in Section 10.

2 Process algebras

Process algebras were originally defined to give semantics to concurrent processes in a computing context and have enjoyed considerable success over the three decades since they emerged. Classical process algebras such as CCS [Mil80] and CSP [Hoa85] focus on the functional capabilities of processes and all actions are atomic with only relative timing of actions captured. Subsequently there have been many extensions of process algebras to capture more information about the system being modelled, for example the relative probability of alternative actions (probabilistic process algebras) and the expected duration of actions (stochastic process algebras).

Each of the process algebras that we consider is based on three fundamental binary operators: action prefix, choice, which is associative and commutative, and synchronous composition, which is also associative and commutative. Note that in the following we omit the cooperation sets for composition in PEPA and Bio-PEPA and assume them to be the *intersection* of the alphabets of the processes involved (denoted $\underset{*}{\bowtie}$). We disregard quantitative aspects of actions, since the representation of kinetics is orthogonal to the expressiveness we consider here. Therefore in our examples, we will assume that the reaction rate for each considered reaction is unique and use this as the name of the corresponding reaction event, i.e. the reaction $A + B \xrightarrow{r_1} C + D$ in chemical notation maps to the process algebra event r_1.

In seminal work, Regev and Shapiro [RS01] suggested an abstraction of *cell-as-computation* and proposed that models formerly used in the study of interacting computational entities, such as Petri nets, process algebras and automata, could be usefully employed for the study of biological processes. In particular they focussed on the π-calculus [Mil99], and subsequently the stochastic π-calculus [Pri95] based on the *molecule-as-process* abstraction. This work has been hugely influential with many other authors following the same abstraction in their own work, even when the details of the process algebra differ.

However, the π-calculus has some particular characteristics that are independent of the *molecule-as-process* abstraction that also shape the style in which models are expressed. In this section we give a brief introduction to process algebras, focussing on the features which lead to different modelling paradigms.

2.1 Forms of synchronisation

The original process algebras, CCS and CSP, differ in their interpretation of actions and consequently the meaning of synchronisation. In CCS all actions are assumed to be communications, and therefore *conjugate*, i.e. actions are paired, corresponding to an input and an output. An action cannot be carried out without its partner, and the pairing of an input and an output becomes a private τ action. This has the consequence that the interaction, or synchronisation, between processes is strictly binary as once an input has been paired with an output both become unavailable for further interaction. In contrast, in CSP no distinction is made between inputs and outputs and there is no notion of complementarity between actions. Instead action type denotes ownership of a *channel* and synchronisation is assumed to take place whenever processes undertake actions of the same type, i.e. communication over the named channel. This is termed *multiway* synchronisation as there is no restriction on the number of processes that may own a channel and thus join a synchronisation. Note that in both these cases the parallel composition operator is generic: in CCS any complementary actions which are on either side of the parallel composition may synchronise; in CSP, processes composed by the parallel operator *must* synchronise on common actions.

Synchronisation in PEPA is a subtle variation of the CSP scheme. Here the parallel composition operator, termed *cooperation*, is decorated by a set of action types (the *cooperation set*) and processes are only forced to synchronise on action types within this set, being able to act concurrently and individually on other action types. Thus the parallel composition is not generic, but a family of parameterised operators. The characteristics of this multiway synchronisation are important in the biological context as they allow one copy of a process (molecule) within a set of identical processes to undertake a reaction individually, something that would not be possible in CSP.[3]

2.2 π-calculus

The π-calculus [Mil99] (and its stochastic form [Pri95]) was designed to express mobility, represented by the passing of channel names. It evolved from CCS [Mil80] and includes the operations of a constant, action prefix, choice, parallel composition, communication and scope restriction. There are variants of the syntax, here we use the following form with events π and processes P:

$$\pi ::= \tau \mid x \mid \overline{x} \mid x(y) \mid \overline{x}\langle y \rangle$$
$$P ::= \mathbf{0} \mid \pi.P \mid P|P \mid P + P \mid \nu x P$$

Following CCS [Mil80], τ is the unobservable event. All other events are observable and paired, e.g. $x(y)$ with $\overline{x}\langle y \rangle$, with $x(y)$ denoting input y on channel x, and $\overline{x}\langle y \rangle$ denoting output y on channel x. $\mathbf{0}$ is the inactive process and $\nu x P$ restricts the scope of the name x to P. In the stochastic form, rates are bound to channels, but as with the other process algebras, we will omit rates here.

A structural congruence, denoted $\equiv$, determines when two syntactic expressions are equivalent, and an operational semantics is given by a set of reaction rules that define how a system evolves following communication. We do not give the full definitions of the congruence and reaction rules, but note two distinguishing features. First, the constant, $\mathbf{0}$, is an identity for parallel composition, i.e. there is a *syntactic* equality $P \mid \mathbf{0} \equiv P$. Second, interaction only occurs when there is a complementary pair of input and output events. The relevant reduction rule is $(\ldots + \overline{x}\langle y \rangle.Q) \mid (\ldots + x(z).P) \rightarrow Q \mid P\{y/z\}$.

There have been numerous applications of the π-calculus to biomolecular processes, starting with the work of Regev *et al.* [RSS01]. An interesting aspect of the application of π-calculus is that it was designed to facilitate modelling mobility and name passing, thus in the original π-calculus events are parameterised, e.g. $x(y)$. Yet, most biological applications do not exploit mobility — the parameter is not relevant, except when modelling compartments, or internal communications. So, in many models unparameterised events are also permitted, e.g. x and $\overline{x}$, and we have also included them here. We note the recent work of Cardelli [Car08] on translations between process algebra and chemical reactions that introduces a subset of the π-calculus and CCS suitable for modelling chemical reactions. It is similar to the syntax above, but excludes event parameters and the ν operator. Additionally, it includes an expression of initial components.

A further distinctive aspect of the π-calculus/CCS paradigm for biomolecular modelling is the underlying assumption of two-way synchronous communication. This means that a a binary chemical reaction, e.g. of the form $A + B \rightarrow^r C$, is modelled by processes A and B offering events r and $\overline{r}$, whereas a unary chemical reaction, e.g. of the form $A \rightarrow^r B$, must be modelled by an unobservable τ event.

[3] This might explain why, to the best of our knowledge, there has been no work applying CSP to biomolecular modelling.

2.3 Beta-binders

Beta-binders [DPPQ06] is a process algebra based on the π-calculus, designed for modelling and simulation of biological processes. A biological process is modelled by a *bio-process*, which is a π-calculus process encapsulated in a box with interaction capabilities expressed as beta-binders. Each communication channel has a set of associated types and there are three kinds of binder: visible, hidden, and complexed. Additionally, there are rates, but these are omitted here. A bio-process is either a constant or pair of encapsulated π-calculus processes composed with a synchronous parallel operator.

The language has evolved over a number of years, here we use the following syntax for boxes B and beta-binders $\mathbf{B}$, assuming π-calculus processes P:

$$B ::= Nil \mid \mathbf{B}[P] \mid B \parallel B$$

$$\mathbf{B} ::= \beta(x, \Gamma) \mid \beta^h(x, \Gamma) \mid \beta^c(x, \Gamma)$$

Further, there is a additional syntactic category for *events*, which include functions on boxes to join, split, create and destroy boxes; these are called *join*, *split*, *new* and *delete*, respectively. These functions are only applied when a condition, defined over binders and π processes, is fulfilled.

Interaction is two-way and is either *intra-box*, in which case it is standard π-calculus interaction, or it is *inter-box* in which case it is specified by the beta-binders and it is between (visible) input/output pairs, but now the types have only to be compatible (rather than identical). There are additional actions (within boxes) that include changing the status of binders (e.g. unhide or change type). There are three structural congruences: $\equiv_p$, the standard congruence on π processes, $\equiv_b$, a congruence on boxes (e.g. $\parallel$ is associative, commutative), and $\equiv_e$, a congruence on events (e.g. join, split have substitution property).

2.4 PEPA

Performance Evaluation Process Algebra (PEPA) was introduced in the early 1990s as a formalism for building Markovian-based performance models of computer and communication systems [Hil96]. All actions in PEPA consist of an action type and a *rate*, which specifies the average duration of the action as an exponentially distributed random variable. The language has a small set of combinators (prefix, choice, parallel composition/cooperation, hiding and constant). Recursive behaviour is specified by mutually recursive definitions. As PEPA was designed for specifying ergodic continuous time Markov chains (CTMC), a restriction is often placed on model construction via a two level syntax, meaning that models consist of parallel compositions of sequential components (constructed using only prefix and choice):

$$S := \alpha.S \mid S + S \mid C$$

$$P := P \underset{L}{\bowtie} P \mid P/L \mid S$$

where S denotes a *sequential component*, P a *model component* and C is a constant defined by a declaration such as

$$C \stackrel{\mathit{def}}{=} S$$

$\alpha.S$ carries out activity α (with an exponentially distributed duration, but omitted here), and it subsequently behaves as S. As discussed above, PEPA supports multi-way cooperations between components: the result of synchronising on an activity α is thus another α, available for further synchronisation. We write $P \underset{L}{\bowtie} Q$ to denote cooperation between P and Q over L. The set which is used as the subscript to the cooperation symbol, the *cooperation set* L, determines those activities on which the *cooperands* are forced to synchronise. For action types not in L, the components proceed independently and concurrently with their enabled activities. We write $P \parallel Q$ as an abbreviation for $P \underset{L}{\bowtie} Q$ when L is empty. P/L denotes the component P in which all actions with types in L are *hidden* meaning that their type is no longer visible but is replaced by the distinguished type τ. We do not consider hiding in the remainder of this paper.

The stochastic nature of the actions means that the choice becomes a probabilistic choice governed by a *race condition* between the involved actions. Similarly actions of parallel components that are not forced to cooperate are also subject to a race condition. When components cooperate on actions but have different definitions of the rate of the action, the rate of the synchronised action is defined to be that of the slowest of the components. While these dynamic considerations do not concern us in this paper, and PEPA has been used for modelling a number of biological examples, we note that the form of the dynamics of synchronisation (the rate of the slowest component) is not always appropriate in this context.

2.5 Bio-PEPA

Bio-PEPA [CH08] is a newly defined modification of the PEPA formalism that has been specifically designed for modelling biochemical networks. It shares many features with PEPA but also has some characteristics to tailor it to the biological application.

Functional rates: In contrast to PEPA, individual processes are not able to define their own rates for actions. Instead the rate associated with an action is specified once, independently of the processes in which the action occurs. The value of this rate can be specified to be a function that depends on the current state of the system.

Stoichiometry: For each action, as well as its type, the stoichiometry or degree of involvement is also specified.

Parameterised processes: Bio-PEPA has been designed to support the population-based reagent-centric style of modelling and so a model consists of a number of sequential components each representing a distinct species which evolve quantitatively (increasing or decreasing amounts). Thus in order to capture the state of a system each component is parameterised recording its current level.

Differentiated prefix: For each action (reaction) that a component is involved in it records its *role* within that reaction, e.g. reactant, product, inhibitor etc. This enables the appropriate values to be used in the functional rate associated with this reaction.

As with PEPA, Bio-PEPA has a two level grammar. The syntax of the sequential (species) components is defined as:

$$S ::= (\alpha, \kappa)\ \mathsf{op}\ S \mid S + S \mid C \qquad\qquad \mathsf{op} ::= \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot.$$

In the prefix term (α, κ) op S, α is an action name and can be viewed as the name or label of a reaction, κ is the stoichiometry coefficient of the species and the prefix combinator op represents the role of the element in the reaction. Specifically, $\downarrow$ denotes the role of reactant, $\uparrow$ product, $\oplus$ activator, $\ominus$ inhibitor and $\odot$ generic modifier. The operator $+$ expresses the choice between possible actions and the constant C is defined by an equation $C \overset{def}{=} S$.

The syntax of model components is defined as:

$$P ::= P \underset{\mathcal{L}}{\bowtie} P \mid S(x)$$

The process $P \underset{\mathcal{L}}{\bowtie} Q$ denotes the synchronisation between components P and Q and the set $\mathcal{L}$ specifies those activities on which the components must synchronise. In the model component $S(x)$, the parameter $x \in \mathbb{R}$ represents the initial concentration by default, although according to the analysis to be carried out the parameter may also be interpreted as number of molecules or molecular level after appropriate conversion.

2.6 sCCP

In the Concurrent Constraint Programming (CCP) process algebra, rather than components and actions, there are components and constraints [BJG96]; there are also variables. The components evolve by adding constraints to a constraint store (tell) or checking the current state of the constraint store (ask). This leads to an asynchronous form of communication between components (via global variables in the constraint store) and there is no direct synchronisation. In addition to tell and ask components may also have choice, parallel composition, procedure call and local variables. In the stochastic form of CCP, sCCP [Bor06], a stochastic duration is associated with the ask and tell operators in a manner analogous to the durations of actions in other stochastic process algebras.

sCCP has been proposed as a modelling formalism for biological networks, and stochastic, deterministic and hybrid semantics have been associated with models in this context [BP08]. The style of modelling is similar to that of Bio-PEPA in that a population-based view is taken, although here explicit variables record the quantitative state of species, rather than parameterised components. At a high level the abstraction is that measurable entities (molecules etc.) are associated with stream variables, logical entities are associated with processes or control variables and reactions are associated with processes. In general a reaction is modelled as a sequence of interactions with the constraint store: first checking that there is sufficient amount of the substrates and then updating the amounts of the products. For mass action reactions the ask step of this sequence will be given a rate equal to the product of the kinetic constant and the amounts of the substrates; the tell step is assumed to be instantaneous. Thus an arbitrary mass action reaction

$$R_1 + \ldots + R_n \longrightarrow_k P_1 + \ldots + P_m$$

will be represented as

$$\begin{aligned}&\mathsf{reaction}(\mathsf{k}, [\mathsf{R_1}, \ldots, \mathsf{R_n}], [\mathsf{P_1}, \ldots, \mathsf{P_m}]) :- \\ &\qquad\qquad \mathsf{ask}_{\mathsf{r_{MA}(k,R_1,\ldots,R_n)}}\Big(\bigwedge_{\mathsf{i=1}}^{\mathsf{n}}(\mathsf{R_i} > 0)\Big). \\ &\qquad\qquad \Big(\,\|_{i=1}^{n}\ \mathsf{tell}_\infty(\mathsf{R_i}\ \$\!=\ \mathsf{R_i} - 1)\ \|_{\mathsf{j=1}}^{\mathsf{m}}\ \mathsf{tell}_\infty(\mathsf{P_j}\ \$\!=\ \mathsf{P_j} + 1)\Big)\end{aligned}$$

Here $\mathsf{R_i}$ and $\mathsf{P_j}$ are stream variables and r_{MA} is a predefined function with the obvious definition.

3 Example pathway

We refer to a small synthetic pathway when exploring how design decisions with respect to the the process algebra determine the mappings available to the modeller. The pathway consists of five representative reactions. The reactions are given in chemical notation in Figure 1, and presented graphically in Figure 2. While the pathway is a synthetic example, it is based on behaviour we have observed in various pathways, including the ubiquitous Raf/MEK/ERK signalling pathway.

$$\begin{array}{lll} A + B & \rightarrow^{r_1} & C \\ C & \rightarrow^{r_2} & A + B \\ B & \rightarrow^{r_3} & D \\ D + E & \rightarrow^{r_4} & B \\ & \rightarrow^{r_5} & E \end{array}$$

Fig. 1. Example pathway in chemical notation

The equations exhibit various combinations of increasing/decreasing/preserved reagents between the left and right hand sides. Specifically, r_1 and r_4 have a decreasing number of reagents, r_2 and r_5 have an increasing number of reagents, and r_3 has the same number of reagents on the left and right hand sides. Note that r_5 has no reagent on the left hand side; we might use a reaction like this

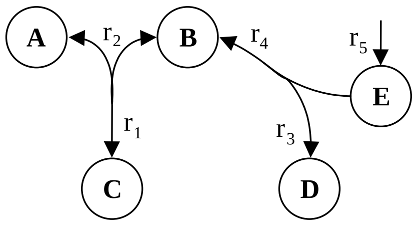


Fig. 2. Example pathway

to indicate that E is plentiful, or that it is produced by another pathway that is irrelevant to this abstraction. We will find it useful to refer to the *degree* of a chemical reaction, meaning the number of reactants that it has i.e. the number of reagents on the left hand side.

We have not included a homeo-reaction [Car08], where the components on the left hand side are identical, as it is only relevant to distinguish this case when rates are determined. In the example pathway, we assume initial concentrations of A, B and E, unless stated otherwise.

4 PEPA models

4.1 Reagent-centric style

In the reagent-centric view, first proposed in [CGH06], species concentrations are discretised into *levels*; the granularity of the system is determined by the number of levels n and the concentration *step size* h, where there is a given maximum concentration max, $h = max/n$. As the number of levels increases/step size decreases, the granularity of the model increases.

For each species, there is a family of processes, each defining the behaviour for that (abstraction of) concentration. The system is defined by the parallel composition of a number of initial components.

The simplest abstraction is obtained when the number of levels is two, so that for each species there are two processes, denoting behaviour in the *presence* and *absence* of that species, respectively. We often refer to this kind of model as the *high/low model*. For example, for species A, A_H denotes presence and A_L denotes absence (alternatively A_1 and A_0, respectively). Figure 3 gives the PEPA high/low model for the example pathway, consisting of a set of equations and a system definition. Figure 4 illustrates the state space for this model.

As an example of a model with a different granularity, Figure 5 contains a reagent-centric model with $n = 3$ (i.e. levels 0, 1, and 2). The state space is in Figure 6. Note that regardless of the number of levels, the number of (system) components is constant during system evolution, i.e. there are always five components (the number of species).

$$
\begin{aligned}
A_H &\overset{def}{=} r1.A_L & \qquad D_H &\overset{def}{=} r4.D_L \\
A_L &\overset{def}{=} r2.A_H & D_L &\overset{def}{=} r3.D_H \\
B_H &\overset{def}{=} r1.B_L + r3.B_L & E_H &\overset{def}{=} r4.E_L \\
B_L &\overset{def}{=} r2.B_H + r4.B_H & E_L &\overset{def}{=} r5.E_H \\
C_H &\overset{def}{=} r2.C_L \\
C_L &\overset{def}{=} r1.C_H
\end{aligned}
$$

$$
System \overset{def}{=} A_H \underset{*}{\bowtie} B_H \underset{*}{\bowtie} C_L \underset{*}{\bowtie} D_L \underset{*}{\bowtie} E_H
$$

Fig. 3. Example pathway: PEPA reagent-centric high/low model

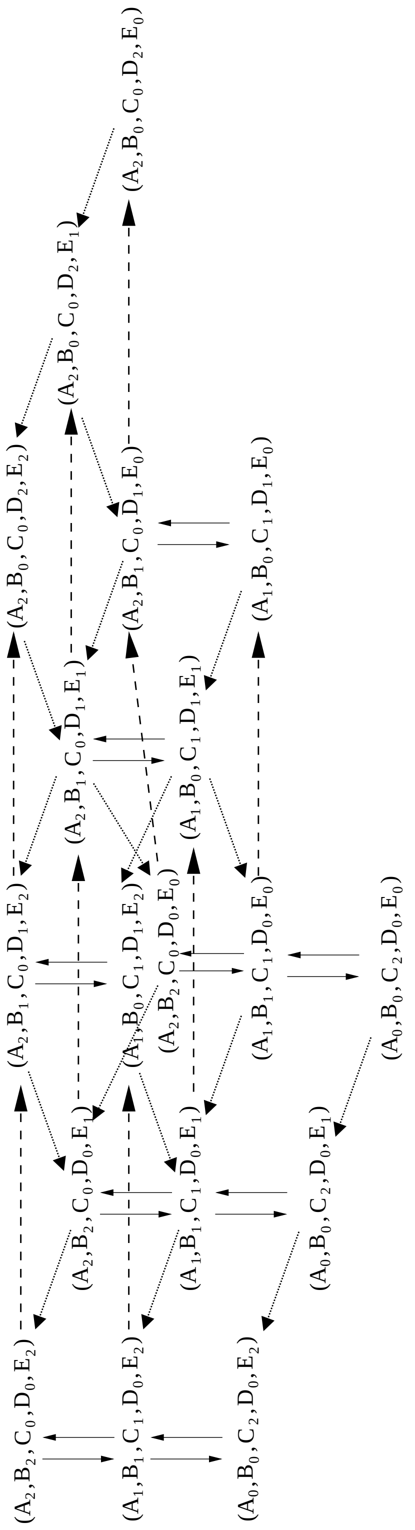
$$\begin{tikzcd}
(A_H, B_H, C_L, D_L, E_H) \arrow[d, bend right, "r_1"'] \arrow[rr, "r_3"] & & (A_H, B_L, C_L, D_H, E_H) \arrow[dl, "r_4"] & \\
(A_L, B_L, C_H, D_L, E_H) \arrow[u, bend right, "r_2"'] & (A_H, B_H, C_L, D_L, E_L) \arrow[ul, "r_5"'] \arrow[rr, "r_3"] \arrow[d, bend right, "r_1"'] & & (A_H, B_L, C_L, D_H, E_L) \arrow[ul, "r_5"'] \\
 & (A_L, B_L, C_H, D_L, E_L) \arrow[u, bend right, "r_2"'] \arrow[ul, "r_5"] & &
\end{tikzcd}$$

Fig. 4. State space of the PEPA high/low model. Note that we use $(A_X, B_X, C_X, D_X, E_X)$ to denote the state since the number of components is fixed and the synchronisation structure does not change.

$$
\begin{aligned}
A_0 &\overset{def}{=} r2.A_1 & \qquad D_0 &\overset{def}{=} r3.D_1 \\
A_1 &\overset{def}{=} r1.A_0 + r2.A_2 & D_1 &\overset{def}{=} r4.D_0 + r3.D_2 \\
A_2 &\overset{def}{=} r1.A_1 & D_2 &\overset{def}{=} r4.D_1 \\
B_0 &\overset{def}{=} r2.B_1 + r4.B_1 & E_0 &\overset{def}{=} r5.E_1 \\
B_1 &\overset{def}{=} r1.B_0 + r3.B_0 + r2.B_2 + r4.B_2 & E_1 &\overset{def}{=} r4.E_0 + r5.E_2 \\
B_2 &\overset{def}{=} r1.B_1 + r3.B_1 & E_2 &\overset{def}{=} r4.E_1 \\
C_0 &\overset{def}{=} r1.C_1 \\
C_1 &\overset{def}{=} r2.C_0 + r1.C_2 \\
C_2 &\overset{def}{=} r2.C_1
\end{aligned}
$$

$$
System \overset{def}{=} A_2 \underset{*}{\bowtie} B_2 \underset{*}{\bowtie} C_0 \underset{*}{\bowtie} D_0 \underset{*}{\bowtie} E_2
$$

Fig. 5. Example pathway: PEPA reagent-centric model with $n = 3$



(A$_2$,B$_2$,C$_0$,D$_0$,E$_2$)
(A$_1$,B$_1$,C$_1$,D$_0$,E$_2$)
(A$_0$,B$_0$,C$_2$,D$_0$,E$_2$)
(A$_2$,B$_2$,C$_0$,D$_0$,E$_1$)
(A$_1$,B$_1$,C$_1$,D$_0$,E$_1$)
(A$_0$,B$_0$,C$_2$,D$_0$,E$_1$)
(A$_2$,B$_1$,C$_0$,D$_1$,E$_2$)
(A$_1$,B$_0$,C$_1$,D$_1$,E$_2$)
(A$_2$,B$_2$,C$_0$,D$_0$,E$_0$)
(A$_1$,B$_1$,C$_1$,D$_0$,E$_0$)
(A$_0$,B$_0$,C$_2$,D$_0$,E$_0$)
(A$_2$,B$_1$,C$_0$,D$_1$,E$_1$)
(A$_1$,B$_0$,C$_1$,D$_1$,E$_1$)
(A$_2$,B$_0$,C$_0$,D$_2$,E$_2$)
(A$_2$,B$_1$,C$_0$,D$_1$,E$_0$)
(A$_1$,B$_0$,C$_1$,D$_1$,E$_0$)
(A$_2$,B$_0$,C$_0$,D$_2$,E$_1$)
(A$_2$,B$_0$,C$_0$,D$_2$,E$_0$)

Fig. 6. State space of the PEPA reagent-centric model with $n = 3$. To avoid clutter in the diagram reaction labels are omitted, but r_1 and r_2 are shown in solid lines, r_3 in dashed lines and r_4 and r_5 in dotted lines.

Process as molecule in reagent-centric style The granularity of the reagent-centric style depends on the step size h. In the limit, the finest grained model has a step size of one molecule. In general, it is impractical to increase n to its corresponding limit, but one alternative is to take a reagent-centric model with $n = 1$ and interpret each process as denoting the presence or absence of a *molecule*. An approach based on this abstraction has been used for studying the FGF pathway using stochastic model checking in [HKNT06]. For our example, for species A, A_H denotes presence of a molecule and A_L denotes absence. So, the population based high/low model model in Figure 3 can also be interpreted as an individuals model, with at most one molecule for each species. Similarly, a model consisting of (at most) two molecules for each species, is given by replacing the system definition of Figure 3 by the system definition:

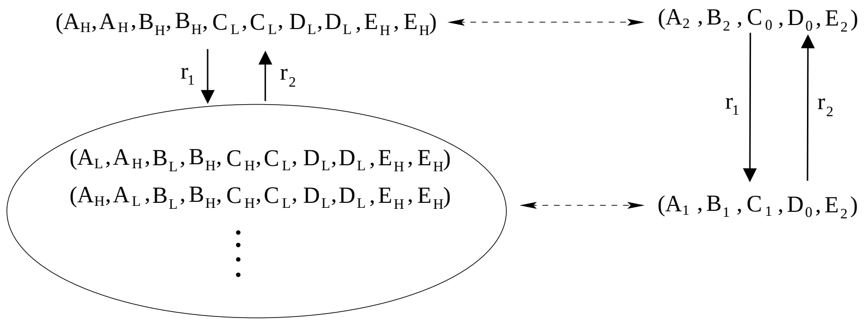
$$(A_H \parallel A_H) \underset{*}{\bowtie} (B_H \parallel B_H) \underset{*}{\bowtie} (C_L \parallel C_L) \underset{*}{\bowtie} (D_L \parallel D_L) \underset{*}{\bowtie} (E_H \parallel E_H).$$

Figure 7 illustrates a small portion of the corresponding state space (one transition step). Notice that this system describes the possible evolution of *every* molecule: it is very fine grained. For example, from the initial state there are 8 possible transitions for reaction r_1, because there are two possible molecules of A that can be consumed, two possible molecules of B that can be consumed, and two possible molecules of C that can be produced (2^3 combinations). Similarly, there are 4 possibilities for reaction r_3.

$$\begin{tikzcd}
(A_H, A_H, B_H, B_H, C_L, C_L, D_L, D_L, E_H, E_H) \arrow[r, "r_3"] \arrow[d, shift right=2, "r_1"'] & (A_H, A_H, B_L, B_H, C_L, C_L, D_H, D_L, E_H, E_H) \\
(A_L, A_H, B_L, B_H, C_H, C_L, D_L, D_L, E_H, E_H) \arrow[u, shift right=2, "r_2"'] & \vdots \\
\vdots &
\end{tikzcd}$$

Fig. 7. One transition step in PEPA reagent-centric process-as-molecule model with two molecules

In many cases this degree of granularity is inappropriate. By appealing to symmetry (i.e. composition is commutative), we can use a form of counter abstraction to represent the molecules $\underbrace{A_H \parallel \ldots \parallel A_H}_{n}$ by A_n, $\underbrace{A_H \parallel \ldots \parallel A_H}_{n-1} \parallel A_L$ by A_{n-1}, and so on. This counter abstraction involves identifying an *equivalence class* of states in a high/low model of m molecules, with a state in a model n levels, where $n = m$. In other words, we define the processes as in the high/low model of Figure 3, then compose multiple copies of each process and interpret A_n as representing n molecules. This is illustrated in Figure 8, for the example pathway with two molecules for each species. States in the fine-grained individuals model are quotiented and dashed lines indicate how the quotient class relates to a state in the counter abstraction model.



$(A_H, A_H, B_H, B_H, C_L, C_L, D_L, D_L, E_H, E_H)$
$(A_L, A_H, B_L, B_H, C_H, C_L, D_L, D_L, E_H, E_H)$
$(A_H, A_L, B_L, B_H, C_H, C_L, D_L, D_L, E_H, E_H)$
$(A_2, B_2, C_0, D_0, E_2)$
$(A_1, B_1, C_1, D_0, E_2)$
r_1
r_2

Fig. 8. One transition step in the state space of the PEPA counter abstraction model

4.2 Pathway-centric style

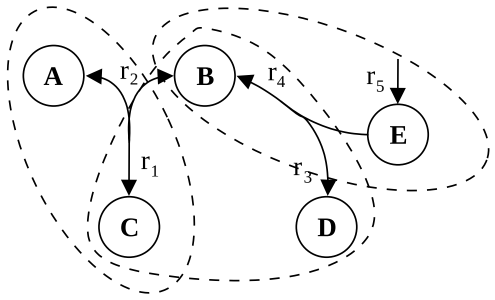


Fig. 9. Example set of reactions with pathways indicated

An alternative style of modelling that has been proposed in PEPA is the pathway-centric style. In this style, we specify the sub-pathways that consume and replenish the *initial species*, which are the species with significant initial concentrations. In the example pathway, this involves defining the sub-pathways starting from A, B, and E. Call these $Path_1$, $Path_2$ and $Path_3$, respectively. The example pathway is given in Figure 10, with corresponding state space in Figure 11.

Notice that although the system definition has only 3 components, this space is isomorphic to the high/low reagent-centric model (Figure 4). Notice also implicitly, the model has two levels. For example, $Path_1$ denotes high concentration of both A and B. We could make levels explicit in this style, by composing multiple copies of each pathway (with parallel composition, no synchronisation). For example the three level model would be:

$$(Path_1 \parallel Path_1) \underset{*}{\bowtie} (Path_2 \parallel Path_2) \underset{*}{\bowtie} (Path_3 \parallel Path_3)$$

$$\begin{aligned}
Path_1 &\overset{def}{=} r_1.r_2.Path_1 \\
Path_2 &\overset{def}{=} r_1.r_2.Path_2 + r_3.r_4.Path_2 \\
Path_3 &\overset{def}{=} r_4.r_5.Path_3
\end{aligned}$$

$$System \overset{def}{=} Path_1 \underset{*}{\bowtie} Path_2 \underset{*}{\bowtie} Path_3$$

Fig. 10. Example pathway: PEPA pathway-centric model

$$\begin{tikzcd}
(Path_1, Path_2, Path_3) \arrow[rr, "r_3"] \arrow[d, shift right, "r_1"'] & & (Path_1, r_4.Path_2, Path_3) \arrow[dl, "r_4"] & \\
(r_2.Path_1, r_2.Path_2, Path_3) \arrow[u, shift right, "r_2"'] & (Path_1, Path_2, r_5.Path_3) \arrow[ul, "r_5"] \arrow[rr, "r_3"] \arrow[d, shift right, "r_1"'] & & (Path_1, r_4.Path_2, Path_3) \arrow[ul, "r_5"'] \\
 & (r_2.Path_1, r_2.Path_2, r_5.Path_3) \arrow[ul, "r_5"] \arrow[u, shift right, "r_2"'] & &
\end{tikzcd}$$

Fig. 11. Pathway-centric model state space

In this case, similarly to the individuals reagent-centric style, there are more potential interleavings than in the reagent-centric population-based representation, and so the explicit state space here will be larger. However, again, by appealing to symmetry, we can work at the aggregate level. Thus for a given number of levels, the state space size and structure of both the pathway-centric and the reagent-centric models should be the same, as established in [CGH06]. Note that tools like the PEPA workbench [TDG09] can automatically detect such symmetries. We observe that assuming chemical reactions of at most degree two, we only require binary synchronisation, for this style of model.

5 Bio-PEPA

The Bio-PEPA formulation [CH08] of the reagent-centric style for the example pathway is given in Figure 12. This example does not fully exploit the power of Bio-PEPA, since the stochiometric coefficients are all simple (1) and the functional rates are omitted. However, it does illustrate how the language focuses on the role of each species, in each reaction. Initial concentrations are denoted A_0 for species A, etc. An integral part of a Bio-PEPA specification (omitted here) is a definition of h and n, for every species, as well as initial concentrations (expressed as levels).

The state space of this model depends upon the levels, for example, if the number of levels is uniformly 2, then the state space is the same as Figure 6. Note that in the corresponding PEPA model (i.e. Figure 5), the number of levels is "hardwired" into the equations, whereas in the Bio-PEPA model, it is given as a parameter offering more flexibility to the modeller. If the number of levels is

$$
\begin{aligned}
A &\overset{def}{=} (r1,1){\downarrow}A + (r2,1){\uparrow}A\\
B &\overset{def}{=} (r1,1){\downarrow}B + (r2,1){\uparrow}B + (r3,1){\downarrow}B\\
C &\overset{def}{=} (r2,1){\uparrow}C\\
D &\overset{def}{=} (r4,1){\downarrow}D + (r3,1){\uparrow}D\\
E &\overset{def}{=} (r4,1){\downarrow}E + (r5,1){\uparrow}E
\end{aligned}
$$

$$
\mathit{System} \overset{def}{=} A(A_0) \underset{*}{\bowtie} B(B_0) \underset{*}{\bowtie} C(C_0) \underset{*}{\bowtie} D(D_0) \underset{*}{\bowtie} E(E_0)
$$

Fig. 12. Example pathway: Bio-PEPA model

set sufficiently high the model has a state space corresponding to an individuals model (i.e. if n is chosen to be the number of molecules).

6 π-calculus

Models in the π-calculus and its stochastic variants predominantly follow the reactant style (e.g. [TK08]), based on the molecules-as-processes abstraction. Thus these are individuals based models. Figure 13 gives the π-calculus model in this style for the example pathway; since each reagent in the example also occurs on the left hand side of a chemical equation, there are processes for A ... E.

The example pathway highlights an interesting aspect of this style because in the biochemistry there are

1. equations with a decreasing number of components, and
2. an equation with no left hand side.

Consider the former case. Since synchronisations are between reagents on the left hand side of an equation only, there is an arbitrary (and inconsequential) choice between which component is output and which is input. Further, the components on the left hand side, when translated into processes, evolve into components on the right hand side. If the number of components decreases, then we have to nominate one or more to evolve to 0, the null process. For example, $A + B \rightarrow^r C$ could map to $A = r.C$ and $B = \overline{r}.0$; equally, it could map to $A = \overline{r}.0$ and $B = r.C$, or $A = r.0$ and $B = \overline{r}.C$, etc. Taking the first choice, $A \mid B$ evolves to $C \mid 0$. This is an example of a "trailing 0", which is removed through application of the *syntactic* equality $P \mid 0 \equiv P$, i.e. $A \mid B$ evolves to C.

Now consider the second case. We cannot model an equation without a left hand side explicitly, e.g. r_5, but since E is present initially, we could represent the infinite supply of E by a τ event, after offering the output event $\overline{r_4}$. However, this would constrain the creation of E to occur only after a molecule has been consumed in the reaction r_4. An alternative, which we use, is to introduce a representation of the environment Env and define it as follows:

$$
Env = \tau.Env \mid E
$$

This presents the possibility that an unbounded number of E molecules may be introduced into the system, which is true when we represent the system only qualitatively. In the biological reality and when quantitative information is included in the model in the form of rates the system will become *pragmatically bounded* meaning that the probability for E to grow unboundedly is extremely small.

$$\begin{aligned}
A &= r_1.C \\
B &= \overline{r_1}.0 \ + \ \tau.D \\
C &= \tau.A \mid B \\
D &= r_4.\ B \\
E &= \overline{r_4}.0 \\
Env &= \tau.Env \mid E \\
\\
System &= A \mid B \mid E \mid Env
\end{aligned}$$

Fig. 13. Example pathway: π-calculus model

Figure 14 illustrates possible evolutions for the system with one molecule of A, B and E initially, i.e. the evolution of $A \mid B \mid E \mid Env$. We have not labelled the transitions since events are either unobservable or become so after synchronisation. Notice that in this state space the number of system components fluctuates, it both increases and decreases. Moreover the state space is infinite due to the potentially unbounded number of E, although a graph isomorphic to the state space of the pathway-centric model is embedded within it. An alternative interpretation of this model is therefore a fine-grained pathway-centric view based on molecules. Or rather, it is a mixture of two styles: equations are defined for each reagent, but the system definition has the form of a pathway-centric model.

While this approach provides a faithful overall system model, it is not compositional. Specifically, one equation incorporates aspects of the initial system and it would be misleading to a reader who inspected the behaviour only of a process that arbitrarily terminates, e.g. B, which can evolve into 0. Moreover, some reactions are represented explicitly by named events, i.e. r_1 and r_4, whereas the unary or nullary reactions r_2, r_3 and r_5 are represented by the τ event. Thus, there are no occurrences of the reaction names r_2, r_3 and r_5 in the model.

7 Beta-binders

There are several ways to map a chemical reaction in this formalism. For example, we could define a mapping very similar to the π-calculus mapping, with boxes for the processes that are initial, i.e. the system is given by $[A] \parallel [B] \parallel [E]$, with suitable beta-binders defined for each box, and each encapsulated process is defined as in Figure 13. The authors recommend this mapping when the reaction

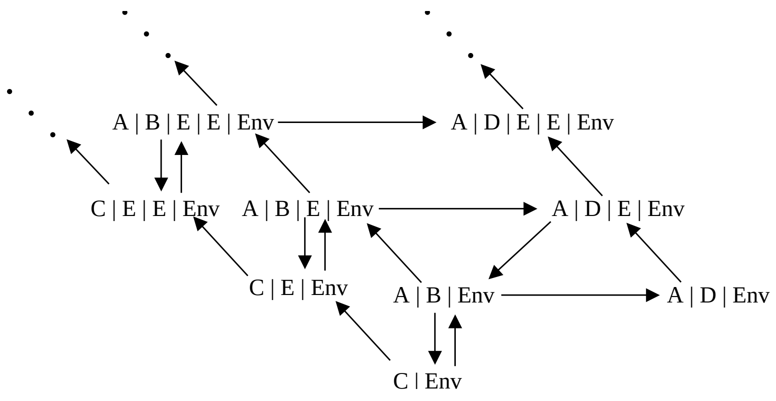


Fig. 14. π-calculus model state space

denotes a collision of entities, the collision being mapped to (inter-box) communication. However, if we use this mapping, we are left with boxes containing the π-calculus constant process (i.e. 0) and we cannot remove them by the structural congruences: we need to introduce an explicit *delete* event to remove them.

Alternatively, instead of representing reactions by inter-box communication, we could represent reactions by events, i.e. by the box operations. In this case, a reaction such as $A + B \rightarrow^r C$ maps to (A, B) *join* C, where A, B, and C are constant bio-processes. Figure 15 gives a Beta-binders model of the example pathway using events. Notice that there are four events and no communication: the encapsulated processes are constants, except for process B, which changes its interaction type (to that of D). The state space is given in Figure 16; the space is isomorphic to the π-calculus model, though we could bound the occurrences of *new* E with a condition. The model is also a mixture of styles: equations are defined for each reagent, but it is not reagent-centric: there is no communication and the system definition has the form of a reaction-centric model.

$$\begin{array}{lll}
(A,B)\ \mathit{join}\ C & \textbf{where} & A = \beta(x,\Gamma_A)\ [\mathit{nil}] \\
C\ \mathit{split}\ (A,B) & & B = \beta(x,\Gamma_B)\ [\mathit{chtype}(x,\Gamma_D).\ \mathit{nil}] \\
(D,E)\ \mathit{join}\ B & & C = \beta(x,\Gamma_C)\ [\mathit{nil}] \\
\mathit{new}\ E & & D = \beta(x,\Gamma_D)\ [\mathit{nil}] \\
 & & E = \beta(x,\Gamma_E)\ [\mathit{nil}]
\end{array}$$

Fig. 15. Example pathway in Beta-binders

There is a third possible mapping when the reaction denotes a binding (e.g. ligand to receptor); this is usually written in chemical notation as: $A{+}B \rightarrow^r [A{+}B]$. In this case we could we use the complex/decomplex beta-binder operations to create and delete dedicated communication channels between boxes $[A]$ and

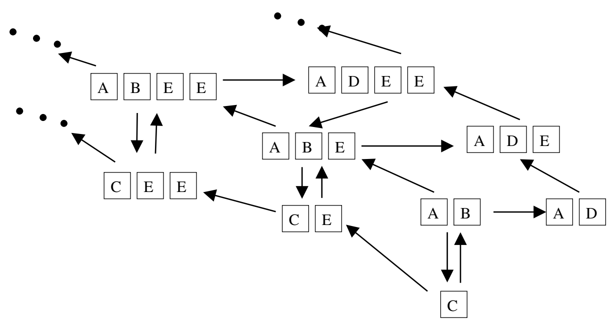


Fig. 16. State space of Beta-binders model. Note that following the graphical notation for Beta-binders, we omit the parallel composition operation $\|$ on bio-processes.

$[B]$. That is, the two boxes $[A]$ and $[B]$ would evolve into a complex of two boxes, instead of into two separate boxes.

8 sCCP

Our last example is a model in sCCP. This is shown in Figure 17. There are five processes: one for each reaction, with stream variables representing the species. Each process has the form ask (check that there is sufficient of a species) followed by the parallel composition of all the possible effects of the reaction (i.e. production or consumption) expressed by tell. The state space of the model is shown in Figure 18. Unsurprisingly this includes the state space which has been retrieved from the other models, such as the reagent-centric PEPA models (shown in the shaded area in the diagram). However note that this model also permits the unbounded growth of the population of E (as in the π-calculus model), leading to an infinite state space unless an explicit guard is inserted which disables reaction r_5 when the population of E reaches a given size.

This model bears some similarity to the state based PRISM model given in [CVOG06], where species are represented by state variables. This is not surprising, since the PRISM modelling language is essentially the language of reactive modules [AH90]. However, in [CVOG06], there is still explicit synchronisation and commands are grouped by species, rather than by reaction. The reactions-as-processes models of sCCP can therefore be considered to be *reaction-centric* and in that they are similar to other rule-based formalisms such as the κ-calculus [VFF$^+$07] and BIOCHAM [CRCD$^+$04].

$$\mathsf{reaction}(\mathsf{r}_1, [\mathsf{A}, \mathsf{B}], [\mathsf{C}]) :-$$
$$\mathsf{ask}(\mathsf{A} > 0 \wedge \mathsf{B} > 0).\, (\mathsf{tell}(\mathsf{A}\ \$\!= \mathsf{A} - 1) \parallel \mathsf{tell}(\mathsf{B}\ \$\!= \mathsf{B} - 1) \parallel \mathsf{tell}(\mathsf{C}\ \$\!= \mathsf{C} + 1))$$

$$\mathsf{reaction}(\mathsf{r}_2, [\mathsf{C}], [\mathsf{A}, \mathsf{B}]) :-$$
$$\mathsf{ask}(\mathsf{C} > 0).\, (\mathsf{tell}(\mathsf{C}\ \$\!= \mathsf{C} - 1) \parallel \mathsf{tell}(\mathsf{A}\ \$\!= \mathsf{A} + 1) \parallel \mathsf{tell}(\mathsf{B}\ \$\!= \mathsf{B} + 1))$$

$$\mathsf{reaction}(\mathsf{r}_3, [\mathsf{B}], [\mathsf{D}]) :-$$
$$\mathsf{ask}(\mathsf{B} > 0).\, (\mathsf{tell}(\mathsf{B}\ \$\!= \mathsf{B} - 1) \parallel \mathsf{tell}(\mathsf{D}\ \$\!= \mathsf{D} + 1))$$

$$\mathsf{reaction}(\mathsf{r}_4, [\mathsf{D}, \mathsf{E}], [\mathsf{B}]) :-$$
$$\mathsf{ask}(\mathsf{D} > 0 \wedge \mathsf{E} > 0).\, (\mathsf{tell}(\mathsf{D}\ \$\!= \mathsf{D} - 1) \parallel \mathsf{tell}(\mathsf{E}\ \$\!= \mathsf{E} - 1) \parallel \mathsf{tell}(\mathsf{B}\ \$\!= \mathsf{B} + 1))$$

$$\mathsf{reaction}(\mathsf{r}_5, [], [\mathsf{E}]) :-$$
$$(\mathsf{tell}(\mathsf{E}\ \$\!= \mathsf{E} + 1))$$

$$\mathsf{5\_reaction\_system} :-$$
$$\mathsf{reaction}(\mathsf{r}_1, [\mathsf{A}, \mathsf{B}], [\mathsf{C}]) \parallel \mathsf{reaction}(\mathsf{r}_2, [\mathsf{C}], [\mathsf{A}, \mathsf{B}]) \parallel \mathsf{reaction}(\mathsf{r}_3, [\mathsf{B}], [\mathsf{D}])$$
$$\parallel \mathsf{reaction}(\mathsf{r}_4, [\mathsf{D}, \mathsf{E}], [\mathsf{B}]) \parallel \mathsf{reaction}(\mathsf{r}_5, [], [\mathsf{E}])$$

Fig. 17. Example pathway: sCCP model

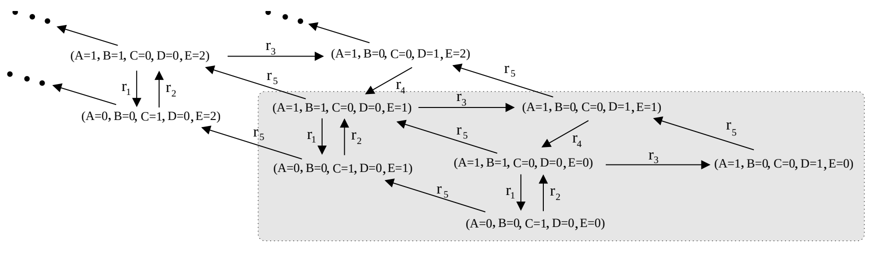

(A=1, B=1, C=0, D=0, E=2)
(A=1, B=0, C=0, D=1, E=2)
(A=0, B=0, C=1, D=0, E=2)
(A=1, B=1, C=0, D=0, E=1)
(A=1, B=0, C=0, D=1, E=1)
(A=0, B=0, C=1, D=0, E=1)
(A=1, B=1, C=0, D=0, E=0)
(A=1, B=0, C=0, D=1, E=0)
(A=0, B=0, C=1, D=0, E=0)
r1
r2
r3
r4
r5

Fig. 18. State space of the sCCP model of the example

9 Discussion

The three main abstractions for mapping chemical equations to process algebras are *molecule-as-process*, *species-as-process*, and *reaction-as-process*. We have further defined four styles: *reagent-centric*, *pathway-centric*, *reactant-centric*, and *reaction-centric*. We have presented reactant-centric π-calculus and Beta-binders models, and (individuals-based) reagent-centric PEPA models as examples of the the molecule-as-process abstraction, (population-based) pathway-centric PEPA and (population-based) reagent-centric Bio-PEPA models as examples of the species-as-process abstraction, and a reaction-centric sCCP model as an example of the reaction-as-process abstraction.

The styles of modelling supported by a process algebra is strongly influenced by the form of synchronisation available. Whilst languages with multiway synchronisation are capable of representing models in reagent-centric, reactant-centric or pathway-centric style, the same is not true for languages with conjugate actions and binary synchronisation. These languages cannot generally represent reactions in the reagent-centric style. Only first degree, or unary, reactions could be modelled in this style in these languages.

In the example considered we have only considered reactions with degree one and two — indeed there are thermodynamic arguments for restricting consideration to such reactions if we wish to be faithful to biochemistry. However, abstractions which lead to higher degree reactions are often applied by biologists for a variety of reasons. For example, consider the enzyme-enabled association of two smaller molecules (A and B) into a complex C. In terms of elementary reactions this might proceed as follows:

$$A + B + E \ {}_{r_2}{\longleftrightarrow}^{r_1} \ A + B{:}E \longrightarrow^{r_3} C + E$$

where E is the enzyme and $B{:}E$ is a complex formed from B and the enzyme. This could abstracted as $A + B \overset{\text{E}}{\longrightarrow}{}^{r} C$. The abstraction has the advantage that the number of reagents considered in the transformation is reduced, and that the number of reaction rates which have to be measured, estimated or fitted is cut from three to one. Moreover this is typically more consistent with what can be observed in the lab as $r_1, r_2 \gg r_3$. It may not even be known whether the enzyme binds with A or B, leaving uncertainty about how to model the reaction without the abstraction. However, representing this in even the reactant-centric style requires three-way synchronisation, and four-way synchronisation in the reagent-centric style, assuming that the enzyme is modelled as both a reactant and a product in the abstracted reaction. In other words, it is not possible to support modelling such biological abstractions using strictly binary synchronisations.

As with reagent-centric style, reaction-centric style seems to implicitly assume a multi-way synchronisation. However note that in the way that this style is captured in sCCP, the only process algebra that currently supports reaction-centric modelling to the best of our knowledge, the requirement is not so strong. What is needed is atomic multi-way composition of updates to the constraint store, but this is not necessarily a synchronisation. Whilst sCCP is the only process algebra supporting reaction-centric, or reaction-as-process, modelling, conversely it is difficult to see sCCP being used to construct models in any of the other styles or abstractions.

In process algebras with conjugate actions, each partner in an action/reaction must be assigned an input/output role. In general this will be rather arbitrary and somewhat artificial from the perspective of the biochemistry. Consider the reaction r_1 in our example. When A and B form the complex C there does not appear to be a natural way to choose which of A and B should receive input and which provide output. Furthermore, reactions of degree one, such as r_5 in the example, must be represented as a τ action. This means that the textual representation of the model does not clearly articulate the biologists' notion of the system. This problem becomes even worse at the level of the state space where all transitions are labelled τ and information about the reactions that gave rise to them is lost.

If we consider the contrast between population-based and individuals-based modelling we can observe that population-based modelling is more compact both from the point of view of the textual model expression and the underlying state space. This means that for such models it can be feasible to use explicit state space representations and the analysis techniques associated with them such as model checking, equivalence checking and numerical analysis of the continuous time Markov chain. Of course, such techniques reply on the state space being finite. In contrast individuals-based modelling has a clear association with stochastic simulation as proposed by Gillespie [GP06]. These models can be used in association with explicit state space techniques, such as those listed above, but only for very small systems or in combination with abstractions such as the assumption of single molecules, as discussed in Section 4.1.

In the PEPA and Bio-PEPA models, as a consequence of the two level grammar used to define these languages as compositions of sequential components, the number of system components is constant, regardless of whether individuals-based or population-based. This matches the species-as-process abstraction since the *possible* species of the pathway will be known and fixed and is particularly natural in the population-based modelling where the state of the system is a count for each species. In contrast, in the π-calculus and Beta-binder models, which are without the syntactic restriction, the number of system components fluctuates throughout system evolution. This is in keeping with the molecules-as-processes abstraction since we would expect the visible molecules within a system to change as complexes are formed and dissociated etc. In sCCP, based on the reaction-as-process abstraction, the number of species is fixed as the variables in the variable store remain fixed. Here as in the PEPA/Bio-PEPA population-based modelling the state of the system is captured in terms of the number of each species so each species must always be present, even if to record that its current count is zero.

The conservative nature of the PEPA/Bio-PEPA models (in terms of number of components, and fixed number of levels) also means that the state space underlying such models is necessarily finite. This is not the case in the other process algebras as we have seen. It can be argued that if we consider the example as presented there is the potential for unbounded numbers of E via reaction r_5 and π-calculus, beta binder and sCCP correctly capture this. But on the other hand, in a biological system unbounded growth like this will lead to cell death, and when we introduced the example we explained that this reaction would be used as an abstraction of some more complex, but bounded, situation. The beta binders and sCCP formalisms do offer language mechanisms which allow the number of E to remain bounded by introducing guards on the reaction, but there is no such possibility in the π-calculus.

In this paper we have focussed on the standard discrete state spaces. However analysis based on these state spaces is rarely feasible. Therefore for all the languages there are alternative semantics given by ordinary differential equations (population-based) and/or Gillespie simulations (individuals-based). The discrete state space does of course form the basis of the Gillespie simulation but it is never considered explicitly and the offered semantics avoid the construction. Additionally, PEPA and Bio-PEPA support an alternative representation, which is based on an explicit discrete state space but seeks to avoid the state space explosion. Rather than states representing the count of molecules of each species, the states represent the current *level* of concentration for each species. In other words, the range of possible concentration values is discretised into intervals, and these intervals constitute the states of the CTMC. In such models the stochastic element of Gillespie's approach is retained but the resulting CTMCs can be considerably smaller. Keeping the state space manageable means that the CTMCs can be solved explicitly and the repeated runs necessitated by stochastic simulation are avoided. Further, in addition to quantitative analysis on the CTMC, analysis by model checking of stochastic properties is possible, as illustrated in [CVOG06] or [HKNT06].

10 Conclusions

As highlighted by Regev and Shapiro computational abstractions have already brought considerable benefit to the study of biological phenomena [RS01]. For example the *DNA-as-string* abstraction has been hugely successful and allowed significant leaps forward. In the context of biomolecular processes the potential benefit seems equally large. However further work is needed to assess the abstractions that are on offer, and their suitability to the systems under study. Research in this direction has been enthusiastically taken up by theoretical computer scientists as witnessed by the plethora of formal languages currently proposed for modelling such systems. In this paper we have aimed to extract the general paradigms of expression which underlie process algebras which aim to model biomolecular processes. We have discovered that there are genuine differences in the form of expression used, and this can impact on the form of analysis that is readily applied.

In the long term all research on formal description techniques for biomolecular systems has the objective of attracting biological users, and contributing to the growing body of knowledge on how cells function. However in the medium term we need to develop closer links with biologists, not only as users of our formal description techniques, but also in the important work of evaluating them.

References

[AH90] R. Alur and T. A. Henzinger. Reactive modules. *Formal methods in System Design*, 15(1):7–48, 1990.

[BMMT06] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo and A. Troina. A Calculus of Looping Sequences for Modelling Microbiological Systems *Fundamenta Informaticae* 72:1-3, pages 21–35, 2006

[BJG96] L. Brim, J-M Jacquet, and D. Gilbert. A process algebra for synchronous concurrent programming. *Proceedings of ALP96: 5th International conference on Algebraic and Logic Programming*, pages 165–178, 1996.

[Bor06] L. Bortolussi. Stochastic concurrent constraint programming. *Proceedings of QAPL2006: 4th International workshop on quantitative aspects of programming languages*, 164:65–80, 2006.

[BP08] L. Bortolussi and A. Policriti. Modelling biological systems in stochastic constraint programming. *Constraints*, 13:66–90, June 2008.

[CGH06] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Trans. on Computat. Syst. Biol. VII*, 4230:1–23, 2006.

[Car04] L. Cardelli. Brane Calculus. in *Proceedings of the 2nd International Workshop on Computational Methods in Systems Biology* Paris, France, April 2004.

[Car08] L. Cardelli. On process rate semantics. *Theoretical Computer Science*, 391(1):190–215, 2008.

[CPR$^+$04] L. Cardelli, E. M. Panina, A. Regev, E. Shapiro and W. Silverman. BioAmbients: An Abstraction for Biological Compartments. *Theoretical Computer Science*, volume 325, number 1, pages 141–167, Elsevier, 2004.

[CG09] F. Ciocchetta and M.L. Guerriero. Modelling Biological Compartments in Bio-PEPA. *ENTCS*, 227:77–95, 2009.

[CH08] F. Ciochetta and J. Hillston. Bio-PEPA: a framework for modelling and analysis of biological systems. *Theoretical Computer Science, to appear.*

[CRCD$^+$04] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.

[CVOG06] M. Calder, V. Vyshemirsky, R. Orton, and D. Gilbert. Analysis of signalling pathways using Continuous Time Markov Chains. *Trans. on Computat. Syst. Biol. VI*, 4220:44–67, 2006.

[VFF$^+$07] V.Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule based modelling of cellular signaling. Proc. CONCUR '07, *Lecture Notes in Computer Science*, 4703:17-41, 2007.

[DPPQ06] P. Degano, D. Prandi, C. Priami, and P.Quaglia. Beta-binders for biological quantitative experiments. *Electronic Notes in Computer Science*, 164:101–117, 2006.

[GP06] D. Gillespie and L. Petzold. *System Modelling in Cellular Biology*, chapter Numerical Simulation for Biochemical Kinetics. MIT Press, 2006.

[HKNT06] J. Heath, M. Kwiatkowska, G. Norman, D. Parker and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. in the Proceedings of 4th International Workshop on *Computational Methods in Systems Biology 2006*. Trento, Italy, 18-19th October 2006.

[Hil96] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[Mil80] R. Milner. *A Calculus for Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1980.

[Mil99] R. Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.

[Pri95] C. Priami. Stochastic π-calculus. *The Computer Journal*, 38:578–589, 1995.

[RS01] A. Regev and E. Shapiro. Cellular abstractions: cells as computation. *Nature*, 419:343, 2001.

[RSS01] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using π-calculus process algebra. *Pacific Symposium on Biocomputing 2001 (PSB 2001)*, pages 459–470, 2001.

[TK08] O. Tymchyshyn and M. Kwiatkowska. Combining intra- and inter-cellular dynamics to investigate intestinal homeostasis. In *Proc. Formal Methods in Systems Biology*, volume 5054 of *Lecture Notes in Computer Science*. Springer, 2008.

[TDG09] M. Tribastone, A. Duguid, and S. Gilmore. The PEPA Eclipse Plug-in. *Performance Evaluation Review*, volume 36, number 4, pages 28–33, 2009.

[V07] C. Versari. A Core Calculus for a Comparative Analysis of Bio-inspired Calculi In *Proc. of ESOP*, volume 4421 of *LNCS*, pages 411–425.Springer 2007.