CoBiC: Context-dependent Bioambient Calculus

Luca Bortolussi [1,3]

Dipartimento di Matematica ed Informatica
University of Trieste, Trieste, Italy

Maria G. Vigliotti [4,2]

Department of Computing, Imperial College,
180 Queen's Gate, London SW7 2BZ, UK

Abstract

In biological phenomena like osmosis, the rate of flow of water molecules in or out of biological compartments depends on the solute concentration and on other forces, like hydrostatic pressure. A similar example is the passive transport of ions in and out the cell membrane. In this paper, we address the problem of faithfully modelling these kind of phenomena with an adequate process calculus. We enhance the ambient calculus stochastic semantics with functional rates, which are calculated by taking into account the volume of ambients and the surrounding environment. A model of osmosis in plant cells will be used as an example to show the new features of our calculus.

Keywords: Stochastic Process algebras, Markov Processes, Systems Biology, Functional rates.

1 Introduction

Since the article by Regev and Shapiro [10], modelling biological systems using process algebras has became very popular. In [10] it was suggested that biological molecules can be represented as processes and signalling as communication. Thus, the behaviour of a biological system can be naturally derived by the (possibly stochastic) semantics of the process calculi. This method proved successful for modelling signalling, biochemical and genetic networks. [2,3,4,8]. Roughly speaking one can see that process calculi like stochastic π-calculus, PEPA, sCCP, and
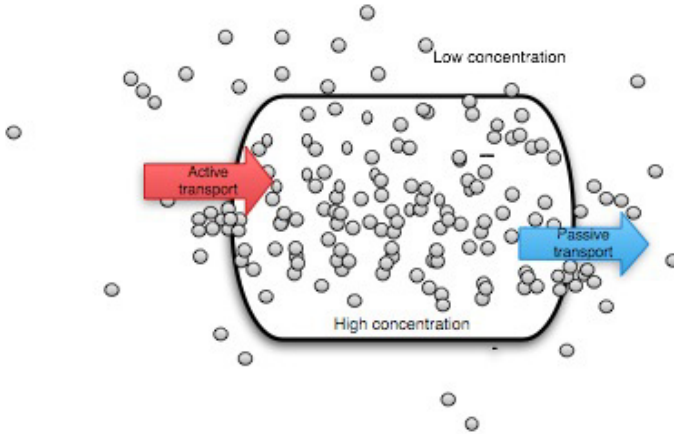
doi:10.1016/j.entcs.2009.10.012

Fig. 1. Graphical representation of active and passive transport

BioAmbients, are mostly suitable to model chemical reactions or systems whose behaviour is naturally modelled as exchange of signals among components.

In this work we focus on modelling transport of molecules in and out of membranes via process calculi. The transport of molecules can be divided in two types: *passive transport* and *active transport*.

Passive transport refers to passage of molecules that does not require any energy. A typical example is *osmosis*. Osmosis is the process of water passing by diffusion from a region of higher concentration to a region of lower concentration. Note that this refers to the concentration of water, not the concentration of any solutes present in the water. We will see in Section 4 how the solute affects osmosis.

Active transport is the pumping of molecules through a membrane against their concentration gradient i.e. from a compartment with lower concentration to a compartment with higher concentration. Active transport requires proteins called *transporters* (usually located on the membrane) and consumption of *energy*. A graphical representation of the active and passive transport can be found in Figure 1.

To model passive or active transport using process calculi, we need the following components:

- A notion of membrane or compartment that separates what is inside from what is outside;
- A function that yields internal concentration of the compartments;
- A function that yields external concentration of the compartments.

In this paper, we have modified BioAmbients [10] by introducing volumes an by allowing rates to be context-dependent. We shall concentrate on modelling passive transport, however our work can be easily adapted to model active transport.

Other people have considered either volumes or functional rates. Versari et al. [11] present a stochastic version of polyadic π-calculus, with volume dependent rates. Their notion of volume turns out to be rather similar to ours, yet their calculus does not have context-dependent rates. Regarding functional rates, they

have been introduced in some stochastic process algebras dealing with biological system: we recall BioPEPA [5], sCCP [3], and a version of π-calculus [8]. However, all these calculi do not have a notion of dynamic compartmentalisation.

The rest of the paper is organised as follows: Section 2 introduces the key ideas behind the definition of the language, Sections 3.1 and 3.2 introduce the syntax and the operational semantics, Section 4 presents the model of osmosis in detail, and Section 5 draws final conclusions and suggests future work.

2 Rationale behind the choice of the language

CoBic, the language that we present hereafter, is essentially Bioambients with some rather important modifications. Prior to introducing the language we explain the rationale beyond our choices.

Concentration To model either passive or active transport correctly, we need to calculate the rate at which molecules enter or exit the cell/membrane. This depends on the concentration and pressure; these quantities are, in turn, determined by the concentration of different substances inside and outside the membrane. To faithfully model this, we need to introduce *context-dependent rates*. This means that our rates are calculated via functions that consider the whole context.

Volume In consideration of defining a calculus for membranes/cells, *volume* is an important parameter as it can influence rates of reactions. It is known that cells modify their volumes, for instance in cell cycle or in during the intake or loss of water provided by the environment. In cells there is a physical limit to volume variation.

3 CoBic: language with context dependent semantics

In this section we introduce the syntax and the semantics of CoBic.

3.1 Syntax

We assume the existence of a set $\mathcal{N}$ of *channel names* and a set $\mathcal{N}_{amb}$ of *ambient names*, which are meant to identify the type of each ambient.

We let the Greek letters $\alpha, \beta, \ldots$ range over ambient names to distinguish those from channel names. The notion of environment we have in mind will be relative to an ambient, as ambients provide a clear notion of border. In order to define the boundaries of the whole system, we will wrap it up in an impenetrable ambient. To this purpose, we introduce a special ambient name system $\notin \mathcal{N}_{amb}$, and let $\overline{\mathcal{N}_{amb}} = \mathcal{N}_{amb} \cup \{\text{system}\}$. We impose that $\mathcal{N} \cap \overline{\mathcal{N}_{amb}} = \emptyset$

We can now introduce the syntax for the extension of Bioambients.

Definition 3.1 The set of processes of CoBic (denoted by CoBic) is given by the

following syntax:

$$P, Q ::= \mathbf{0} \mid \sum_{i \in I} M_i.P_i \mid (\mathsf{new}\, n)\, P \mid \mathsf{A}\langle \widetilde{x} \rangle \mid \alpha\, [P]^V \mid P \mid Q$$
$$M ::= \mathbf{enter}\, n \mid \mathbf{exit}\, n \mid \mathbf{accept}\, n \mid \mathbf{expel}\, n \mid \tau_a \mid \$n(x) \mid \$\overline{n}\langle m \rangle$$
$$\$ ::= \mathbf{s2s} \mid \mathbf{local} \mid \mathbf{p2c} \mid \mathbf{c2p}$$

We now explain the informal meaning of the syntax. The different primitives for communication and enabling ambients are inherited from BioAmbients.

Communication Communication happens on a channel n by sending on a name -or channel- m; $\$n(x)$ stands for the input, and $\$\overline{n}\langle m \rangle$ stands for output. There are three ways of communicating: channels in the same ambient perform *local communication*, $\mathbf{local}\, n(y)$ for the input on channel n and $\mathbf{local}\, \overline{n}\langle m \rangle$ for the output of m on channel n. Inputs and outputs located in sibling ambients respectively perform *sibling communication*; $\mathbf{s2s}\, n(y)$ stands for such input and $\mathbf{s2s}\, \overline{n}\langle m \rangle$ stands for output. Finally, parent to child communication happens when inputs $\mathbf{p2c}\, n(y)$ and outputs $\mathbf{p2c}\, \overline{n}\langle m \rangle$ are located in parent-child ambients respectively (or vice-versa for $\mathbf{c2p}\, n(y)$ and $\mathbf{c2p}\, \overline{n}\langle m \rangle$).

Ambient capabilities The capabilities such as $\mathbf{exit}\, n$ or $\mathbf{enter}\, n$ give the ambient the power to become active; $\mathbf{enter}\, n/\mathbf{accept}\, n$ allow an ambient to move into a sibling, $\mathbf{exit}\, n/\mathbf{expel}\, n$ allow a child ambient to leave the parent. Note that we have excluded the capability of merging two ambients, which is present in the original paper.

In general, inputs are binding operators on the arguments. This means that in the process $\mathbf{local}\, n(y).P$ the name y is bound in P, and not accessible from outside P. A similar argument applies to the other inputs in the communication primitives. The definition of *free names* of channels of P, written $\mathsf{fn}(P)$, (and *bound names* of P written $\mathsf{bn}(P)$) is standard, taking into account that the only binding operators are inputs and restriction. We write $P\{y/m\}$ to mean the substitution of every occurrence of the name y by m in P. Similarly we write $P\{A/Q\}$ to mean the substitution of every occurrence of the process A by Q in P.

$\mathbf{0}$ *Nil* represents the inactive process.

$\sum_{i \in I} M_i.P_i$ *Local sum* represents the standard choice. Given a set of indexes I and a permutation p on it, we write $\sum_{p(i) \in I} M_{p(i)}.P_{p(i)}$ to represent a reordering of the terms of the summation. We reserve the letters G, C to represent summation as in $\sum_{i \in I} M_i.P_{p(i)} = M_j.P_j + G$ where $G = \sum_{i \in I\ i \neq j} M_i.P_i$.

The process $\tau_a.P$ represents an exponentially distributed time delay. Its subscript $a \in \mathcal{N}$ is used to attach to each τ a specific rate function (cf. below).

$(\mathsf{new}\, a)\, P$ *Restriction* of the name a makes that name private and unique to P: the name a becomes bound in P. Restriction does not apply to ambient names.

$\mathsf{A}\langle \widetilde{x} \rangle$ *Recursion* models infinite behaviour by assuming the existence of a set of equations of the form $A(\widetilde{x}) \stackrel{\mathrm{df}}{=} P$ such that $\{\widetilde{x}\} \subset \mathsf{fn}(P)$.

$\alpha\, [P]^V$ *Ambient* represents a compartment with name α that contains a process P.

Each ambient has a basic or minimal volume $V \in \mathbb{R}^+$ associated to it. Its actual volume is calculated according to Definition 3.2. In our setting, ambients can represent either physical or abstract objects. In the latter case, the volume of the ambient is 0 and it can be omitted i.e. $\alpha\,[S]$.

$P|Q$ *Parallel composition* means that P and Q are running in parallel.

We introduce now some definitions to be used in the following.

Definition 3.2 The volume V_α of a CoBic process is calculated as $V_\alpha \stackrel{\text{df}}{=} f_V(\alpha\,[P]^V)$, where the function $f_V : \mathsf{CoBic} \to \mathbb{R}^+$ is defined as follows:

$$f_V(\mathbf{0}) \stackrel{\text{df}}{=} 0$$
$$f_V(M.P) \stackrel{\text{df}}{=} 0$$
$$f_V(\textstyle\sum_{i \in I} M_i.P_i) \stackrel{\text{df}}{=} \sum_{i \in I} f_V(M_i.P_i) = 0$$
$$f_V((\mathsf{new}\,n)\,P) = f_V(P)$$
$$f_V(\mathsf{A}\langle \widetilde{x} \rangle) \stackrel{\text{df}}{=} f_V(P) \quad \text{if}\,\mathsf{A}\langle \widetilde{x} \rangle = P$$
$$f_V(\beta\,[P]^V) \stackrel{\text{df}}{=} V + f_V(P)$$
$$f_V(P \mid Q) \stackrel{\text{df}}{=} f_V(P) + f_V(Q)$$

Definition 3.3 The number of ambients with name $\alpha \in \mathcal{N}_{amb}$ inside another ambient is calculated by the function $\epsilon_\alpha : \mathsf{CoBic} \to \mathbb{N}$

$$\epsilon_\alpha(P) = \begin{cases} \epsilon_\alpha^\sharp(S) & \text{if} \quad P = \beta\,[S]^V \\ \text{undefined} & \text{otherwise} \end{cases}$$

where $\epsilon_\alpha^\sharp$ is defined as follows:

$$\epsilon_\alpha^\sharp(\mathbf{0}) = 0$$
$$\epsilon_\alpha^\sharp(M.P) = 0$$
$$\epsilon_\alpha^\sharp(\textstyle\sum_{i \in I} M_i.P_i) \stackrel{\text{df}}{=} 0$$
$$\epsilon_\alpha^\sharp((\mathsf{new}\,n)\,P) = 0$$
$$\epsilon_\alpha^\sharp(\mathsf{A}\langle \widetilde{x} \rangle) \stackrel{\text{df}}{=} \epsilon_a^\sharp(P) \quad \text{if}\,\mathsf{A}\langle \widetilde{x} \rangle = P$$
$$\epsilon_\alpha^\sharp(\beta\,[P]^V) = \begin{cases} 1 \text{ if } \beta = \alpha \\ 0 \text{ otherwise} \end{cases}$$
$$\epsilon_\alpha^\sharp(P \mid Q) = \epsilon_a^\sharp(P) + \epsilon_a^\sharp(Q)$$

3.2 Semantics

The semantics of CoBic is defined by a set of reduction rules in the style of Chemical Abstract Machine [1], but it is different from standard stochastic semantics [12,6]. The main difference is the fact that rates of basic actions in CoBic are not real numbers, but rather functions depending on the context of the executing processes. These functions, in any case, depend only on the global configuration of the system,

hence in each state of a model they are evaluated to (positive) real numbers, so that from the labeled transition system a Continuous Time Markov (CTMC) Chain [9] can still be derived.

First of all, we need to define the notion of environment of a process. For convenience, we parameterize such definition w.r.t. a *finite* subset $A \subseteq \mathcal{N}_{amb}$, counting how many occurrences of ambients $\alpha \in A$ are present in parallel.

Definition 3.4 The *inner environment* of an ambient $\alpha \, [S]^V$, w.r.t. $A \subseteq \mathcal{N}_{amb}$ is the vector $\nu_A(\alpha \, [S]^V)$ in $\mathbb{R}^{|A|}$, defined by

$$\nu_A(\alpha \, [S]^V) = \left(\epsilon_a(\alpha \, [S]^V) \right)_{a \in A}$$

We now introduce the set $\mathcal{F}_\mathcal{R}$ of functions that will be used in defining the rates of actions. Each element of $\mathcal{F}_\mathcal{R}$ is a function $f_A : \mathbb{R}^{h_f} \to \mathbb{R}^+$, where $A \in \mathcal{N}_{amb}$ is the set of ambients on which f_A depends and h_f is the number of arguments of f_A, depending on the type of action of which f_A is the rate (cf. below). In general, the functions in $\mathcal{F}_\mathcal{R}$ will depend on the inner environment and volume of the ambient in which the corresponding action happens, and possibly also on the inner environments and the volumes of the ambients involved in the action, if any.

We can now proceed to associate a rate function to each different channel name. In general, channel names can be used to perform different kinds of actions (enter, exit, and so on), whose meaning is different. Hence, functions assigning rates to enter actions **enter** n need to be different from functions assigning rates of exit actions **exit** n (a similar argument is valid for other action's types). Thus, we have chosen to associate a function to each channel name *and* action type.

Definition 3.5 Let $\mathcal{A} = \{enter, exit, local, s2s, p2c, \tau\}$ be the set of action types. The function $\rho : \mathcal{N} \times \mathcal{A} \to \mathcal{F}_\mathcal{R}$ associates a function from $\mathcal{F}_\mathcal{R}$ to each pair of channel names and action types.

The reduction rules of the operational semantics are defined in Figure 3. Before commenting them, we need to define some notational conventions that simplify their writing. If $\alpha \, [S]^V$ is an ambient appearing in the left hand side of a rule, with $\nu_{A,\alpha}$ we indicate the $|A|$-vector $\nu_{A,\alpha} = \nu_A(\alpha \, [S]^V)$ (here and in the following $A \subseteq \mathcal{N}_{amb}$) and with $V_\alpha = f_V(\alpha \, [S]^V)$ its volume. Furthermore, we assume that $\boldsymbol{x}_A$ is always a vector of non-instantiated variables of size $|A|$, while y is a single non-instantiated variable. Finally, with σ we indicate either a real number r or a function $f_A(\boldsymbol{z})$ with non-instantiated variables $\boldsymbol{z}$.

Each action has a rate calculated according to a rate function $\rho(n, type)$, depending on the channel and on the action type. The information needed to compute such functions is not always available at the descriptive level of each rule: in fact, some rates may depend on the context, which is not known by the agents performing the action. In order to solve this problem, we introduce a simple mechanism: some variables in the function giving the rate of the rule are left non-instantiated, and will be instantiated afterwards, when the context will be "closed", i.e. when all the agents in parallel with the communicating ones will be wrapped up in an ambient. As a

matter of fact, this is the effect of rule (AMB2), which computes the inner environment and the volume of the ambient γ, substituting them into the non-instantiated variables of the rate function. Note that the fact that CoBic models are always surrounded by the special system ambient is crucial in order to guarantee that all functions appearing in the derivation rules will have all their variables sooner or later instantiated. In fact, the rule (AMB2) can be applied also to this special ambient (as $\gamma \in \overline{\mathcal{N}_{amb}}$ in the rules of Figure 3) [5]. There is another rule which deals with ambients, namely (AMB1), which is needed to deal with the case in which all variables of rate functions are already instantiated, hence the function has already been evaluated to a real number r.

We now discuss the dependence of rate function on action types in more detail.

enter: we assume that the rate of entrance of an ambient α into an ambient β may depend on the inner environment and the volume of their parent ambient. It may also depend on the inner environment of the accepting ambient β, and on volumes of α and β. The parent ambient is however not specified in the left hand side of the rule for enter/accept in Figure 3, hence we leave some variables non-instantiated in its rate function, namely those for the state of the environment ($\boldsymbol{x}$) and for the volume of the parent ambient (y). According to the previous discussion, these values will be set by rule (AMB2), when applied to the parent ambient of α and β.

exit: we assume that the rate of an exit action can depend on the inner environment of the expelling ambient, on its volume and on the volume of the expelled ambient.

local: the rate of a local communication can depend on the volume of the ambient in which it happens, and also on its inner environment.

s2s: similarly to a local communication, a sibling to sibling one will have a rate dependent on the volume and the inner environment of the parent ambient of the two siblings.

p2c–c2p: the rate for parent to child communication may depend on the inner environments and the volumes of the parent and the child ambients.

τ**-actions:** silent actions are indexed by names, hence ρ assigns a rate function also to them, depending on the inner environment and the volume of the containing ambient.

The rules dealing with parallel composition, restriction, structural congruence are similar to the ones in Bioambients [12].

In the previous discussion, we did not provide a specific functional form for the rates. Rate functions can range from constant functions to very complex functions of the environment. They will be specified by the programmer and suggested by the model under study. This gives a great deal of flexibility to the language. In the next section we will exemplify this by dealing with osmosis.

Remark 3.6 The rate functions assigned by ρ do not need to depend on all the

---

$$P \mid \mathbf{0} \equiv P \qquad\qquad P \mid Q \equiv Q \mid P$$
$$(P \mid Q) \mid R \ \equiv\ P \mid (Q \mid R) \qquad\qquad \sum_{i \in I} M_i.P_i \equiv \sum_{p(i) \in I} M_{p(i)}.P_{p(i)}$$
$$\text{system}\,[P] \mid \text{system}\,[P] \ \equiv\ \text{system}\,[P|Q] \qquad (\mathsf{new}\,n)\,\alpha\,[P]^V \equiv \alpha\,[(\mathsf{new}\,n)\,P]^V$$
$$(\mathsf{new}\,n)\,\mathbf{0} \ \equiv\ \mathbf{0} \qquad\qquad (\mathsf{new}\,m)\,(\mathsf{new}\,n)\,P \equiv (\mathsf{new}\,n)\,(\mathsf{new}\,m)\,P$$
$$(\mathsf{new}\,n)\,(P \mid Q) \ \equiv\ P \mid (\mathsf{new}\,n)\,Q \qquad \text{if } n \notin \mathsf{fn}(P)$$
$$\mathsf{A}\langle \widetilde{m} \rangle \ \equiv\ P\{\widetilde{m}/\widetilde{x}\} \qquad\qquad \text{if } A(\widetilde{x}) = P$$

Fig. 2. Structural congruence

$$G + \tau_a.P \xrightarrow{f_A(\boldsymbol{x}_A, y)} P, \text{ with } \rho(a, \tau) = f_A$$

$$\alpha\,[(G + \mathbf{enter}\,n.P) \mid Q]^{V_1} \mid \beta\,[(G' + \mathbf{accept}\,n.R) \mid S]^{V_2} \xrightarrow{\sigma} \beta\,[\alpha\,[P \mid Q]^{V_1} \mid R \mid S]^{V_2}$$
$$\text{with } \sigma = f_A(\nu_{A,\beta}, \boldsymbol{x}_A, V_\alpha, V_\beta, y), \ f_A = \rho(n, enter)$$

$$\alpha\,[\beta\,[(G + \mathbf{exit}\,n.P) \mid Q]^{V_1} \mid (G' + \mathbf{expel}\,n.R) \mid S]^{V_2} \xrightarrow{\sigma} \alpha\,[P \mid Q]^{V_1} \mid \beta\,[R \mid S]^{V_2}$$
$$\text{with } \sigma = f_A(\nu_{A,\alpha}, V_\alpha, V_\beta), \ f_A = \rho(n, exit)$$

$$(C + \mathbf{local}\,n(y).P) \mid (C' + \mathbf{local}\,\overline{n}\langle m \rangle.Q) \xrightarrow{\sigma} P\{y/m\} \mid Q$$
$$\text{with } \sigma = f_A(\boldsymbol{x}_A, y), \ f_A = \rho(n, local)$$

$$\alpha\,[(C + \mathbf{s2s}\,n(y).P)]^{V_1} \mid \beta\,[(C' + \mathbf{s2s}\,\overline{n}\langle m \rangle.Q)]^{V_2} \xrightarrow{\sigma} \alpha\,[P\{y/m\}]^{V_1} \mid \beta\,[Q]^{V_2}$$
$$\text{with } \sigma = f_A(\boldsymbol{x}_A, y), \ f_A = \rho(n, s2s)$$

$$\gamma\,[\beta\,[(C + \mathbf{c2p}\,n(y).P) \mid R]^{V_1} \mid (C' + \mathbf{p2c}\,\overline{n}\langle m \rangle.Q) \mid S]^{V_2} \xrightarrow{\sigma} \gamma\,[\beta\,[P\{y/m\} \mid R]^{V_1} \mid Q \mid S]^{V_2}$$
$$\text{with } \sigma = f_A(\nu_{A,\gamma}, \nu_{A,\beta}, V_\gamma, V_\beta), \ f_A = \rho(n, s2s)$$

$$\gamma\,[\beta\,[(C + \mathbf{c2p}\,\overline{n}\langle m \rangle.Q) \mid R]^{V_1} \mid (C' + \mathbf{p2c}\,n(y).P) \mid S]^{V_2} \xrightarrow{\sigma} \gamma\,[\beta\,[Q \mid R]^{V_1} \mid P\{y/m\} \mid S]^{V_2}$$
$$\text{with } \sigma = f_A(\nu_{A,\gamma}, \nu_{A,\beta}, V_\gamma, V_\beta), \ f_A = \rho(n, c2p)$$

$$(\text{PAR}) \quad \frac{P \xrightarrow{\sigma} P'}{P \mid R \xrightarrow{\sigma} P' \mid R} \qquad\qquad (\text{NEW}) \quad \frac{P \xrightarrow{\sigma} P'}{(\mathsf{new}\,n)\,P \xrightarrow{\sigma} (\mathsf{new}\,n)\,P'}$$

$$(\text{AMB1}) \quad \frac{P \xrightarrow{r} P'}{\gamma\,[P]^V \xrightarrow{r} \gamma\,[P']^V} \qquad (\text{AMB2}) \quad \frac{P \xrightarrow{f_A(\boldsymbol{x}_A, y)} P'}{\gamma\,[P]^V \xrightarrow{f_A(\nu_{A,\gamma}, V_\gamma)} \gamma\,[P']^V}$$

$$(\text{CONG}) \quad \frac{P \equiv P' \xrightarrow{\sigma} Q' \equiv Q}{P \xrightarrow{\sigma} Q} \qquad\qquad \text{with } \alpha, \beta \in \mathcal{N}_{amb}, \gamma \in \overline{\mathcal{N}_{amb}}$$

Fig. 3. Reduction Relation

values described in the rules. In the simplest case, they can be taken as constant functions. If all the rate functions are constant, we obtain the BioAmbients calculus as a subset of CoBic.

Definition 3.7 Let P be a process. A CoBic *model is written as* system $[P]^V$.

Models enjoy the nice property that the rate of each transition is fully resolved to a real number.

Proposition 3.8 *Let M, M' be two* CoBic *models, such that* $M \xrightarrow{\sigma} M'$. *Then* $\sigma \in \mathbb{R}$.

The semantics as CTMC is derived in the standard way [12,6]. Moreover, since our rates depend on the state of the system only, our CTMC is time-homogenous,

which implies that our system can be simulated, for instance, via direct method [13].

4   Osmosis

In this section we provide a detailed example of the osmosis to to show how CoBic can be put to use.

4.1   The physical process

Osmosis is the physical process regulating the quantity of water within a cell. It is a passive process: the cell consumes no energy, as water movement is driven by physical forces arising due to different energy levels between the cell and its surrounding environment. Water tends to flow from regions of low solute concentration to regions of higher concentration (Figure 1). The cell membrane is selectively permeable: just small, non-charged molecules can pass through freely, other molecules (charged ions and large molecules, like proteins) can be incorporated only by an active cellular process or through protein channels. Therefore, most of the substances dissolved in water cannot cross the cell membrane passively, but water can. Brownian motion of water molecules makes them bump into the cell membrane, so that some of them can cross it, inwards or outwards. The presence of a solute in water reduces the freedom of movement of water molecules (some of them are tied to the solute by electrostatic forces), hence water molecules in a concentrated solution tend to cross the membrane less frequently. Consequently, the net balance of water flow is from regions with low concentration (more freedom of movement) to regions of high concentration (less freedom of movement).

A flow of water into the cell will increase its volume; a flow of water outside the cell will reduce it. If the volume variation becomes significant, the cell can collapse (plasmolysis) or burst. In plants, in particular, the cell membrane is surrounded by a semi-rigid structure called the cell wall. Thus, an increase in water quantity (hence volume) induces an increase of the pressure exerted by the cell wall. Pressure is another important regulator of osmosis: high pressure increases the frequency of pumping of water molecules into the cell membrane, hence the flow of water outwards of the cell. The gradients of water flow induced by concentration and pressure are important mechanisms for plants, as they also keep cells turgid in non-ligneous parts of the plants, like stems.

There are also other forces that take part in the osmotic process, like gravity and surface adherence between water and other materials. However, they are not relevant to our work. We refer the reader to [7] for further details.

In order to quantify the rate of water flow, we need to delve into some physical notion. The rate of osmosis depends on the difference in *water potential*, a measure of Gibbs free energy associated to the water status. The water potential, historically denoted by ψ_w, is the sum of two terms, $\psi_w = \psi_s + \psi_p$:

• The *solute potential* ψ_s, depends on the concentration of solutes dissolved in

water. For a dilute solution of non-dissociating substances[6], like sucrose, there is a simple expression for such potential, the *van Hoff's equation*: $\psi_s = -\mathsf{RT}c_s$, where R is the gas constant, T is the absolute temperature, and c_s is the solute concentration. An important issue is that this formula is independent from the type of solute.

- The *pressure potential* ψ_p, depends -in cells- mainly on the elastic properties of their membrane and of the cell wall. For simplicity, in this work we assume ψ_p to be linearly depended on the cell volume.

Once we have computed the water potentials of the cell and its environment, we can give an expression of the rate of flow of water molecules into and from the cell. According to [7], the total flow is

$$J = L_p \frac{S}{V} \Delta\psi_w,$$

where L_p is the *hydraulic conductivity constant* (depending on semipermeability properties of the cell membrane), S is the cell's surface, V is the cell's volume and $\Delta\psi_w = \psi_{w(ext)} - \psi_{w(int)}$ is the difference in water potential between the external water potential and internal water potential. The sign of ψ_w determines the direction of the flow (positive if directed towards the cell, negative otherwise).

4.2   Model of Osmosis in CoBic

In CoBic modelling osmosis is quite natural because of the functional definition of rates. The model is shown in Table 1. Water is free to get in and out the cell, while the membrane is impermeable to the solute. The flow of ambients into the cell is modeled with an enter/accept pair, similarly the exit of water ambients towards the environment is modeled by an exit/expel pair. As we can see, only one accept/expel agent is active in the cell, and it represents the membrane semipermeability.

The model in Table 1 is not complete: we still have to specify rates and volumes. In order to do this, we first recall that the concentration of the solute in a water solution is defined as the ratio between the number of molecules (usually expressed in moles) of the solute and the volume of the solution. The solution volume is different from the volume of the entire cell, and the latter contains some parts that are not dissolved in water (like organelles). As a first approximation, ignoring effects of temperature, the volume of a solution is given by the sum of the volume of water and the volume of the solute. For example, in looking at the $CELL$ agent, we can write:

$$V_{solution}^{CELL} = N_{water}^{CELL} V_{water} + N_{solute}^{CELL} V_{solute} \qquad (1)$$

where $N_{water}^{CELL} = \epsilon_{water}(CELL)$ and $N_{solute}^{CELL} = \epsilon_{solute}(CELL)$ is the number of water (resp. solute) molecules and V_{water} (V_{solute}) is the volume of one single molecule

---

$$P_{in} = \mathbf{enter}\, n.P_{in}$$
$$P_{out} = \mathbf{exit}\, m.P_{out}$$
$$Q_{in} = \mathbf{accept}\, n.Q_{in}$$
$$Q_{out} = \mathbf{expel}\, m.Q_{out}$$
$$W = water[P_{in} \mid P_{out}]^{V_{water}}$$
$$S = solute[]^{V_{solute}}$$
$$CELL = cell[Q_{in} \mid Q_{out} \mid \underbrace{W \mid \ldots \mid W}_{k_{in}} \mid \underbrace{S \mid \cdots \mid S}_{h_{in}}]^{V_0^{CELL}}$$
$$SYSTEM = \mathrm{system}[\underbrace{W \mid \ldots \mid W}_{k_{out}} \mid \underbrace{S \mid \cdots \mid S}_{h_{out}} \mid CELL]^{V_z}$$

Table 1
The CoBic code for the osmosis. The set of ambient names in the SYSTEM is
$\eta(SYSTEM) = \{water, solute, cell\}$, plus the special ambient system, which wraps up the whole model.
The rate functions $\rho(n, enter)$ and $\rho(m, exit)$, and volume functions are specified in the text.

of water (solute). In real terms, V_{water} roughly equals $V_{water} = 3 \cdot 10^{-26}$, while for sucrose, a typical solute inside cells, approximatively $V_{sucrose} = 3.6 \cdot 10^{-25}$.

Since the volume of the cell does not coincide with the volume of the water solution inside: we need to add another term representing the volume of elements not diluted in water. Hence,

$$V_{cell} = V_0^{CELL} + V_{solution}^{CELL},$$

where $V_{solution}^{CELL}$ is defined in (1). This defines the volume functions. As for rates, we start by defining them only taking into account the effect of the concentration gradient. According to the previous discussion, we may define rates as $L_p \frac{S}{V} \psi_s$, with $\psi_s = -\mathsf{RT} c_{solute}$ the solute potential. Now, the rate of water molecules into the cell will depend on the water potential outside the cell, while the rate of exit of water from the cell will depend on the water potential inside the cell. The previous definition, however, will not work, as the ψ_s are negative! We cannot simply take their absolute value, as this would reverse the direction of osmosis. However, observe that the lower the solute concentration, the higher the water concentration, defined as the ratio between the number of water molecules and the solution's volume. Hence, we can simply replace solute concentration with water concentration, obtaining the following solute potential:

$$\psi'_s = \mathsf{RT} \frac{N_{water}}{N_A V_{solution}},$$

with N_A the Avogadro number.

In our case we can easily calculate each water potential for $CELL$ and for the concentration outside the cell in $SYSTEM$:

$$\psi'_{s(cell)} = \mathsf{RT} \frac{N_{water}^{CELL}}{N_A V_{solution}^{CELL}}, \qquad \psi'_{s(System)} = \mathsf{RT} \frac{N_{water}^{SYSTEM}}{N_A V_{solution}^{SYSTEM}}, \tag{2}$$

hence $\psi'_{w(cell)} = \psi'_{s(cell)}$ and $\psi'_{w(System)} = \psi'_{s(System)}$. The total flow is:

$$J = J_{in} - J_{out} = L_p \frac{S}{V}(\psi'_{w(System)} - \psi'_{w(cell)}) \tag{3}$$

To obtain the rate functions for enter and exit actions, we still need to define the surface to volume ratio. If we approximate the cell shape with a sphere, the ration $\frac{S}{V}$ equals $\frac{3}{r}$, with $r = \sqrt[3]{\frac{3}{4\pi}V_{cell}}$. Other parameters are set according to [7]: $L_p = 10^{-6}$ and $\mathsf{RT} = 2.5$ (corresponding to $\mathsf{T} = 30°C$). J_{in} and J_{out} (cf. 3) are the total flows of the system, which can be seen as the speed of movement of all water molecules. Hence, if we want to assign a rate to each single water molecule, we need to divide J_{in} and J_{out} by the number of water molecules outside (resp. inside) the cell. Hence, the rate for enter and exit actions is the following:

$$3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\psi'_{w(system)}}{N_{water}^{SYSTEM}} = 3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\mathsf{RT}}{N_A V_{solution}^{SYSTEM}} \tag{4}$$

$$3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\psi'_{w(cell)}}{N_{water}^{CELL}} = 3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\mathsf{RT}}{N_A V_{solution}^{CELL}} \tag{5}$$

Remark 4.1 By following the rule for exit in Figure 3 we can calculate the expression (5), which essentially depends on three variables: $N_{water}^{CELL}, N_{solute}^{CELL}, V_{cell}$, since L_p and $\mathsf{T}, \mathsf{R}, N_A$ are fixed parameters.

$$\rho(m, exit) = f_A(\nu_{A,cell}, V_{cell}, V_{water}), \quad A = \{cell, solute\}$$

where $\nu_{A,cell} = (N_{water}^{CELL}, N_{solute}^{CELL})$. We observe that in f there are no variables, in order to derive the final rate we apply the rule (AMB1) from Figure 3. Similarly, the function for entrance of water molecules in cell, $\rho(n, enter)$, depends on the volume of the cell, and on the number of water and solute molecules in the system. Unlike the exit, these values are not at disposal at the moment of application of the rule, hence they must be left non-instantiated. The function is thus $\rho(n, enter) = g_A(\nu_{A,cell}, x_{water}, x_{solute}, V_{cell}, y)$ where

$$g_A(\nu_{A,cell}, x_{water}, x_{solute}, V_{cell}, y) = 3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\mathsf{RT}x_{water}}{N_A V_{solution}^{SYSTEM}}\frac{1}{x_{water}}$$
$$= 3L_p\left(\frac{3}{4\pi}V_{cell}\right)^{-\frac{1}{3}}\frac{\mathsf{RT}}{N_A V_{solution}^{SYSTEM}}$$

When looking at $SYSTEM$, in order to derive the final rate we need to apply the rule (AMB2), we obtain $g = g_A(\nu_{A,system}, V_{system})$, which is the rate in Equation (4), as $\nu_{A,system} = (N_{water}^{SYSTEM}, N_{solute}^{SYSTEM})$.

In Figure 4, we present the result of a simulation of the model of Table 1 (dotted lines), showing the temporal evolution of the solute concentration inside and outside the cell [7]. As we can see, the net effect is that the solute concentration inside the

---

[7] We decided to depict the solute concentration instead of the number of water molecules or the water concentration, as this is the most common measure.
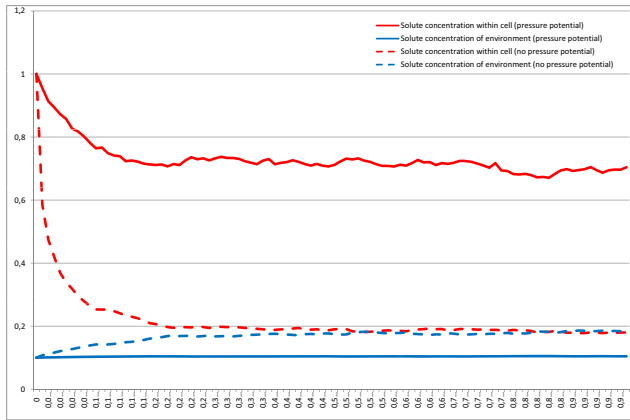
Fig. 4. Solute concentration ignoring the effects of pressure (dotted lines) and considering it (solid lines) The initial solute concentration in SYSTEM is equal to 0.1, while the initial concentration in CELL is 1. This corresponds to the following number of ambients inside and outside: set by $k_{in} = 1960$, $k_{out} = 24460$, $h_{in} = h_{out} = 4500$. Each water ambient represents not one, but a set of water molecules (this is to speed up the simulation), namely 10^7. Similarly, each solute ambient represents 10^5 molecules of sucrose, taken here as the prototype solute.

cell, initially equal to 1, decreases, while the concentration outside the cell increases, until they reach their equilibrium value, close to 0.2. This corresponds to a net flow of water within the cell, which increases its volume. Looking at Figure 5, we can see that the cell volume increases 7 times [8], an event that would probably kill the cell. This can happen for cells having no protection against influx of water from outside. In plant cells, instead, the presence of the cell wall counterbalances the effect of the concentration gradient by exerting a strong pressure inside the cell.

In order to take into account the effects of pressure, we need to introduce a model of the pressure potential. We assume the following simple model [7]:

$$\psi_p(V) = \begin{cases} 0 & \text{if } V \le \alpha V_0 \\ \frac{1}{(1-\alpha)^2}\left(\frac{V}{V_0} - \alpha\right)^2 & \text{for } V \ge \alpha V_0 \end{cases}$$

with $V_0 = 10^{-15}$ and $\alpha = 0.85$. The pressure potential outside the cell is assumed to be zero. Now, we have new water potentials, namely $\psi''_{w(cell)} = \psi'_{s(cell)} + \psi_p$ and $\psi''_{w(system)} = \psi'_{s(system)}$. Rates for enter and exit actions are like those in equations (4) and (5), with ψ''_w in place of ψ'_w.

In Figure 4, we also show the result of simulating the system with both concentration and pressure potential. As we can see, the solute concentration converges to the value of 0.7, a much higher level than the one reached under concentration effects alone, while the system concentration remains almost constant. Inspecting again Figure 5, we can see that now the volume increases by a factor of 1.3, which is an acceptable parameter.

---

[8] We assumed an initial cell volume of 10^{-15} liters, $\frac{3}{4}$ of which due to the water solution.
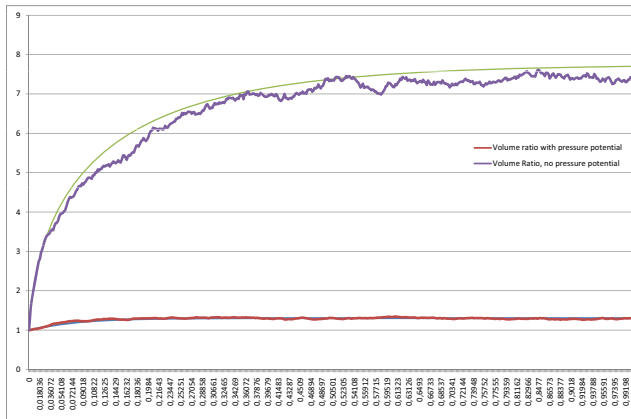
Fig. 5. Increase of volume of cell. We plot the ratio w.r.t. the initial volume, both for the system with rates depending only on concentration and for the system with rates depending also on pressure (cf. Figure 4).

5   Conclusions and future work

In this paper we have shown how to model passive transport in cells using a variant of BioAmbients we called CoBic as the rates of the transition depend on the environment of the process. We also provided a detailed example of the osmosis.

The general functional form of rates endows CoBic with the possibility of integrating analytic information in a process-based calculus. This has been exploited in osmosis to provide a more realistic model of water flow. Other straightforward applications are in the modeling of ion transportation in and out a cell. Indeed, this feature can be used also to *abstract* some aspects of the model, describing them in the functional rates. For instance, the kinetics of a simple enzymatic reaction can be abstracted (neglecting the formation of the substrate-enzyme complex) using a Michaelis-Menten form for the rate.

References

[1] G. Berry and G. Boudol. Chemical Abstract Machine. In *Proceedings of the 17th ACM Symposium on Principles of Programming Languages*, pages 81–94. ACM, 1990.

[2] R. Blossey, L. Cardelli, and A. Phillips. A compositional approach to the stochastic dynamics of gene networks. *Transactions on Computational Systems Biology IV*, 3939:99–122, 2006.

[3] L. Bortolussi and A. Policriti. Modeling biological systems in concurrent constraint programming. *Constraints*, 13(1), 2008.

[4] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In *Proceedings of the BioConcur Workshop on Concurrent Models in Molecular Biology*, London, England, August 2004.

[5] F. Ciocchetta and J. Hillston. Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. In *Proceeding of FBTC 2007*, 2007. Workshop of CONCUR 2007.

[6] J. Hillston. *A Compositional Approach to Perfomance Modelling*. PhD thesis, Department of Computer Science, Edinburgh, 1994.

[7] l. Tainz and E. Zeiger. *Plant Physiology*. Sinauer Associated Inc., 2006.

[8] P. Lecca and C. Priami. Cell cycle control in eukaryotes: A biospi model. *Electron. Notes Theor. Comput. Sci.*, 180(3):51–63, 2007.

[9] J. R. Norris. *Markov Chains*. Cambridge University Press, 1997.

[10] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E.Y. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[11] C. Versari and N. Busi. Efficient Stochastic Simulation of Biological Systems with Multiple Variable Volumes. In *Proceeding of FBTC 2007*, 2007. Workshop of CONCUR 2007.

[12] M.G. Vigliotti and P.G. Harrison. Stochastic mobile ambients. In *Proceeding of QAPL 2006*, volume 164 (issue 3) of *E N T C S*, pages 169–186. Elsevier, 2006.

[13] D.J. Wilkinson. *Stochastic Modelling for System Biology*. Chapman & Hall, 2006.