

# The $\pi$ -calculus and Model Checking for Molecular Systems

Gabriel Ciobanu<sup>1</sup> and Vlad Ciubotariu<sup>2</sup>

<sup>1</sup> Institute of Computer Science, Romanian Academy, Iași, Romania

<sup>2</sup> Department of Computer Science, University of Waterloo, Canada

**Abstract.** This paper describes the kinetics of the sodium-potassium exchange pump in terms of the  $\pi$ -calculus process algebra. The software tool called Mobility Workbench is used to check properties of finite state transition system of the pump. Some important properties of the model are proved by using this model checker, for instance the model of the pump is deadlock free. It is proved that a detailed description with a large number of states is equivalent with a simple model with a small number of states. This simpler model can become a part of a larger system, and in this way we get a scalable and compositional abstraction. Finally we use the stochastic  $\pi$ -calculus to describe a more realistic description of the pump with one miss per four execution cycle with the model, based on the assumption of the same speed of every actions.

## 1 Introduction

A new and active field of research in computer science (called “computational methods in system biology”) try to model and simulate various biological and biochemical networks (metabolic networks, molecular networks, gene networks) that are so complex that they require a formal framework for an accurate representation. Recent works done by Cardelli, Harel, Pnueli, Regev, Shapiro and some other known scientists suggested that process algebra, in particular the  $\pi$ -calculus and mobile ambients, may become valuable tools in modelling and simulation of the biological systems where the interaction and mobility are important features. The field may have a huge impact in understanding how the biological systems are working, giving in the same time a way to describe, manipulate and analyze them.

A real challenge for computer science is to understand what paradigm of computation the cell is using (if any), and what are the appropriate tools that could be used to study it. An important step is to find a good and appropriate abstraction for the biological systems. A good and appropriate abstraction should be able to highlight essential properties of a system, ignoring complicating and unimportant aspects. It should be relevant and understandable, providing a conceptual framework to express systems in a formal way, and then to prove their properties. The community working in molecular systems did not identify and adopt a unifying abstraction for describing dynamics, interaction, qualitative and quantitative reasoning, similar behaviours of two related systems, and scalable to systems of higher levels.

It is reasonable to expect that, in order to model the biological systems, we can adapt and apply the range of tools developed in concurrency theory from the formalisms

of process algebras to the accompanying model-checking techniques for temporal logics. This is what we do in this paper. We present a system of interacting entities as a system of computational interacting entities, and then use a process algebra to describe, simulate, and more important to check automatically properties of the molecular systems. We insist on the model checking aspects related to this representation of the biomolecular systems.

We focus on the sodium-potassium exchange pump. It is system where we have a lot of mobility; the main function of the pump is to move Na and K ions across a cell membrane, namely Na ions from inside to outside, and K ions from outside to inside. As the  $\pi$ -calculus [7] is a widely accepted formalism for mobility, we use the  $\pi$ -calculus to describe the changing configuration and movements of the pump. Having such a description, we emphasize on a powerful software tool that is able to check properties and to provide (proofs and) a certain confidence in the formal description.

## 2 Sodium-potassium Exchange Pump

Cell membranes are crucial to the life of the cell. A membrane defined the boundary of the first living cell nearby 4 billions years ago. Since then, membranes have acquired additional functions and modern membranes participate in many essential cell activities including barrier functions, transmembrane signaling and intercellular recognition. A very substantial fraction of the energy available for life processes is maintained in the form of gradients across biological membranes. Every organism spends a large part of its energy intake and production to generate a significant difference between the compositions of the cell interior and the cell exterior. We refer to the sodium-potassium exchange pump that is a membrane-bound protein that establishes and maintains the high internal  $K^+$  and low internal  $Na^+$  concentrations. By using the energy from the hydrolysis of one molecule of ATP, it transports three  $Na^+$  out in exchange for two  $K^+$  that are taken in. This exchange is critical in maintaining the osmotic balance of the cell, the resting membrane potential of most tissues, and the excitable properties of muscle and nerve cells.

The description given in Table 1 derives from what is known as the Albers-Post model. According to the Albers-Post cycle,  $Na^+$  and  $K^+$  transport occur as a ping-pong mechanism (meaning that the two ion species are transported sequentially), and Na-K pump essentially exists in two conformations,  $E_1$  and  $E_2$ , which may be phosphorylated or dephosphorylated. Ion transport is mediated by transitions between these conformations.  $E_1$  will bind  $Na^+$  to a high-affinity site available only from the inside (1). The binding of the sodium stimulates the enzyme to hydrolyze ATP (2), forming a phosphorylated enzyme intermediate.  $E_1$  changes to  $E_2$  (3):  $Na^+$  is exposed to the outside surface and  $Na^+$  binding is of a low-affinity type.  $Na^+$  is then released to the outside (4). Exposed also on the outside surface by the  $E_2$  phosphorylated enzyme is a potassium binding site. When potassium binds (5), the phosphoenzyme is hydrolyzed (6). This stimulates the enzyme to expose the potassium binding site to the inside surface of the membrane, changing its conformation (7).  $K^+$  binding becomes of low affinity and the release of the potassium to the inside follows (8). Inside and outside in this mechanism refer to the inside and the outside of the cell plasma membrane in

$E_1 + Na^I \Leftrightarrow Na \cdot E_1$	(1)
$Na \cdot E_1 + ATP \Leftrightarrow Na \cdot E_1 \sim P + ADP$	(2)
$Na \cdot E_1 \sim P \Leftrightarrow Na \cdot E_2 \sim P$	(3)
$Na \cdot E_2 \sim P \Leftrightarrow E_2 \sim P + Na^{II}$	(4)
$E_2 \sim P + K^{II} \Leftrightarrow K \cdot E_2 \sim P$	(5)
$K \cdot E_2 \sim P \Leftrightarrow K \cdot E_2 + P_i$	(6)
$K \cdot E_2 \Leftrightarrow K \cdot E_1$	(7)
$K \cdot E_1 \Leftrightarrow K^I + E_1$	(8)

**Table 1.** Albers-Post Model

which the  $Na^+/K^+$ -ATPase resides. The ATPase is now ready to bind sodium once more.

We emphasize the aspect of the relationship between the kinetic parameters of the transport process and the efficiency of the pump. The rate constants of competing steps (that would decrease the efficiency) are small. This ensures that the binding and the release of substrate occur at the proper point in the cycle. For example, the rate constant for the reaction  $E_1 + ATP \Leftrightarrow E_1 \sim P + ADP$  is much less than that of equations (1) and (2). As a consequence,  $E_1$  has enough time to bind sodium before undergoing the transition to  $E_2$ . Similar relationships among rate constants ensure that ions are released from the enzyme before they come back to the side at which they were initially bound. In other words, the slow rate constants "channel the enzyme" along a reaction path in which the hydrolysis of ATP is tightly coupled to the transport process.

### 3 The $\pi$ -calculus and Stochastic $\pi$ -calculus

We have many formalisms to describe mobile computation. Among them, the  $\pi$ -calculus [7] is a widely accepted model of interacting systems with dynamically evolving communication topology. Its mobility is expressed by the changing configuration and connectivity among processes. This mobility increases the expressive power of the  $\pi$ -calculus, and the  $\pi$ -calculus is a general model of computation which takes interaction as a primitive. The  $\pi$ -calculus models computing by reduction and interactive matching. Both the sender and receiver offer their availability for communication. Similar mechanisms work in computation and in biology [3, 4].

The  $\pi$ -calculus was introduced by Milner, Parrow, and Walker [8] as an attempt to describe mobile concurrent processes. A good reference is [7]. It is a very useful formalism for interacting systems with dynamically evolving communication topology. The  $\pi$ -calculus allows channels to be passed as data along other channels, and this fact provides a channel mobility. This mobility is an important feature and increases the expressive power. The  $\pi$ -calculus has a simple semantics and a tractable algebraic theory. The computational world of the  $\pi$ -calculus contains just processes (also called

agents) and channels (also called names or ports). The  $\pi$ -calculus models networks in which messages are sent from one site to another site, possibly containing links to active processes or to other sites. The  $\pi$ -calculus is a general model of computation which takes interaction as primitive. The  $\pi$ -calculus emphasizes the use of processes in systems with dynamical changing configurations and topology. The  $\pi$ -calculus allows dynamic reconfiguration among processes. The  $\pi$ -calculus is able to describe mobile systems, providing a conceptual framework and mathematical tools. There are two basic entities in  $\pi$ -calculus: the channels, named  $x, y, \dots$  and processes  $P, Q, \dots$  that communicate through them. Also, there are two types of atomic actions, called guards or prefixes: the input guard  $x(y)$  to receive a name for  $y$  along the channel  $x$ , and the output guard  $\bar{x}z$  to send the name  $z$  along the channel  $x$ .

Interaction is established by a nondeterministic matching which dynamically binds "senders" to eligible "receivers". Even though there are many pairs which can satisfy the matching condition, only a single receiver gets the commitment of the sender. Thus processes can interact by using names they share. A name received in one interaction can be used in another; by receiving a name, a process can interact with processes which are unknown to it, but now they share the same channel name. The  $\pi$ -calculus mobility is coming from its scoping of names and extrusion of names from their scopes. Starting with atomic actions and simpler processes, complex processes can be constructed in many ways. The process expressions are defined by guarded processes, parallel composition  $P|Q$ , nondeterministic choice  $P + Q$ , replication  $!P$  and a restriction operator  $(\nu x)P$  creating a local fresh channel  $x$  for a process  $P$ . Over the set of processes it is defined a structural congruence relation; this relation provides a static semantics of some formal constructions. The structural congruence deals with the aspects related to the structure of the processes. The evolution of a process is described in  $\pi$ -calculus by a reduction relation over processes called reaction. This reaction relation contains those transitions that can be inferred from a set of rules.

We present in this section the monadic version of the  $\pi$ -calculus: this means that a message consists of exactly one name. Let  $\mathcal{X} \subset N$  be a infinite countable set of *names*. The elements of  $\mathcal{X}$  are denoted by  $x, y, z, \dots$ . The terms of this formalism are called processes and processes are denoted by  $P, Q, R, \dots$ .

**Definition 1.** *The processes are defined over the set  $\mathcal{X}$  of names by using the prefixes*

$$\pi ::= \bar{x}z \mid x(y) \mid \tau \mid [x = y]\pi$$

*The processes are defined by the following grammar*

$$P ::= 0 \mid \pi.P \mid P + Q \mid P|Q \mid !P \mid (\nu x)P$$

Processes evolve by performing (inter)actions, and these (inter)actions are given by their prefixes  $\pi$ . The input and output prefixes represent sending and receiving a name along a channel. The output prefix  $\bar{x}z$  sends  $z$  along  $x$ ; an input prefix  $x(y)$  waits until a name is received along  $x$  and substitutes it for the bound variable  $y$ .  $\tau$  is an unobservable action;  $\tau$  can be thought of as expressing an internal action of a process. The match prefix  $[x = y]\pi.P$  can evolve as  $\pi.P$  if  $x$  and  $y$  are the same, and do nothing otherwise.  $0$  is the empty process.  $P + Q$  represents a nondeterministic choice

of  $P$  or  $Q$ .  $P \mid Q$  represents the parallel composition of  $P$  and  $Q$ . A replicated process  $!P$  denotes a process that allows to generate arbitrary instances of  $P$  in parallel. The replication  $!P$  can be expressed by recursive equations of parametric processes as well. The informal meaning of the restriction  $(\nu x)P$  is that  $x$  is a local fresh channel for  $P$ .

The parallel composition  $\bar{x}z.P \mid x(y).Q$  may synchronize on  $x$ . An interaction is actually defined by a "sender"  $\bar{x}z.P$  and a "receiver"  $x(y).Q$ , and it can be represented by the transition:

$$(\bar{x}z.P \mid x(y).Q) \rightarrow (P \mid Q\{z/y\})$$

This is a synchronous interaction, where the send operation is blocking: an output guard cannot be passed without the simultaneous occurrence of an input action. The prefix  $x(y)$  binds the name  $y$ , and  $(\nu x)$  binds the name  $x$ . We denote by  $fn(P)$  the set of the names with free occurrences in  $P$ . We denote by  $P\{v/u\}$  the result of simultaneous substitution in  $P$  of all free occurrences of the name  $u$  by the name  $v$ , using the  $\alpha$ -conversion wherever necessary to avoid the name capture.

Over the set of processes it is defined a structural congruence relation; this relation provides a static semantics of some formal constructions. We denote by  $=_\alpha$  the standard  $\alpha$ -conversion.

**Definition 2.** *The relation  $\equiv$  over processes is called structural congruence and it is defined as the smallest congruence over processes which satisfies*

- $[x = x]\pi.P \equiv \pi.P$
- $P \equiv Q$  if  $P =_\alpha Q$
- $P + 0 \equiv P$ ,  $P + Q \equiv Q + P$ ,  $(P + Q) + R \equiv P + (Q + R)$ ,
- $P \mid 0 \equiv P$ ,  $P \mid Q \equiv Q \mid P$ ,  $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$ ,
- $!P \equiv P \mid !P$
- $\nu x 0 \equiv 0$ ,  $\nu x \nu y P \equiv \nu y \nu x P$ ,
- $\nu x(P \mid Q) \equiv P \mid \nu x Q$  if  $x \notin fn(P)$ .

The structural congruence deals with the aspects related to the structure of the processes. The evolution of a process is described in  $\pi$ -calculus by a reduction relation over processes called reaction. This reaction relation contains those transitions which can be inferred from a set of rules.

**Definition 3.** *The reduction relation over processes is defined as the smallest relation  $\rightarrow$  satisfying the following rules:*

$$\begin{array}{ll}
 \text{pre: } \alpha.P \xrightarrow{\alpha} P & \text{sum: } \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \\
 \text{rep: } \frac{P \mid !P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'} & \text{par: } \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \\
 \text{com: } \frac{P \xrightarrow{\bar{a}y} P' \quad Q \xrightarrow{a(x)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q' \{y/x\}} & \text{match: } \frac{P \xrightarrow{\alpha} P'}{[a = a]P \xrightarrow{\alpha} P'} \\
 \text{res: } \frac{P \xrightarrow{\alpha} P'}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'} \quad x \notin n(\alpha) & \text{struct: } \frac{P \equiv P' \quad P \xrightarrow{\alpha} Q \quad Q' \equiv Q}{P' \xrightarrow{\alpha} Q'}
 \end{array}$$

The most studied forms of behavioural equivalence in process algebras are based on the notion of bisimulation. There have been given several definitions in the literature for bisimilarity of which one of them is called *open* bisimilarity. Its definition is given by using the labeled transition system defined by the reduction rules. We use here the so-called late-style transition system.

**Definition 4.** A relation  $S$  defined over processes is called an open simulation if for all  $P, Q$  whenever  $P S Q$  then for all substitutions  $\sigma$  the following holds

$$\text{if } P\sigma \xrightarrow{\alpha} P' \text{ there exists } Q' \text{ so that } Q\sigma \xrightarrow{\alpha} Q' \text{ and } P' S Q'.$$

$S$  is an open bisimulation if both  $S$  and  $S^{-1}$  are open simulations. Two processes  $P$  and  $Q$  are open bisimilar  $P \sim Q$  if there exists an open bisimulation  $S$  that relates them, i.e.  $P S Q$ .

Systems will be checked automatically by studying the bisimilarity between two processes, namely the model and its specification. More helpful in the verification process is another kind of bisimilarity called *weak open bisimilarity*. It allows the basic verification technique for proving properties about the mobile concurrent systems modeled in the  $\pi$ -calculus<sup>1</sup>. Let  $\Longrightarrow \stackrel{def}{=} \xrightarrow{\tau}^*$  be the transitive and reflexive closure of the  $\xrightarrow{\tau}$  relation and let  $\xrightarrow{\alpha} \stackrel{def}{=} \Longrightarrow$  whenever  $\alpha = \tau$  and  $\xrightarrow{\alpha} \xrightarrow{\alpha} \Longrightarrow$  if  $\alpha \neq \tau$ . The *weak open bisimulation* denoted by  $\approx$  is defined exactly as in the strong case replacing  $Q\sigma \xrightarrow{\alpha} Q'$  with  $Q\sigma \xrightarrow{\alpha} Q'$ .

A version of the  $\pi$ -calculus called stochastic  $\pi$ -calculus is useful in describing the efficiency of the Na-K pump. In the stochastic  $\pi$ -calculus, the prefix  $\pi.P$  is replaced by  $(\pi, d).P$ , where  $d$  is a probability distribution that characterizes the stochastic behaviour of the activity corresponding to the prefix  $\pi$ . Otherwise, the syntax remains the same as in  $\pi$ -calculus. This variant of the  $\pi$ -calculus allows us to describe the complexity of the molecular interactions involving the dynamic efficiency of the pump and other quantitative aspects (e.g. kinetics rates, energy).

## 4 Formal Description of the Sodium-potassium Pump

We present the computational model of the Na-K exchange pump. The equations of Albers-Post model are translated into an appropriate operational semantic which can describe both protein interactions (conformational transformations) and membrane transportation occurring in the pump mechanism. Together with the computational model, we provide an automated way of verification. In this way some important properties of the model can be checked by using a software (program) called Mobility Workbench.

### 4.1 The Computational Model

Generally speaking, the molecular components could be treated as computational processes where their individual domains correspond to communication channels. The

<sup>1</sup> Note however that the usual model checking techniques are still applicable where the state space of a process is finite.

complementary molecular domains that allow their interaction can be modeled as the ends of a channel (one end for input and another for output). In this way, molecular interaction coincides with communication and channel transmission.

The membrane transport system involves both information and energy. Consequently, we assume that the  $\pi$ -calculus can model and discipline formally the interactions along the sodium-potassium exchange. These interactions are defined syntactically and they have a clear operational semantics given by a reduction relation. It is possible to define rigorously and to study their behaviour. Finally the molecular conformational shapes are modified and the capabilities of the interacting components are dynamically changed.

In Table 2 we present the computational model of the Albers-Post mechanism.

$PhaseI = \underbrace{siteI(i).([i = K]PhaseI + [i = Na]atp.\tau.\overline{siteII}i.PhaseII)}_{E_1}$ $PhaseII = \underbrace{siteII(i).([i = Na]PhaseII + [i = K]\overline{atp}.\tau.\overline{siteI}i.PhaseI)}_{E_2}$ $ATPase = PhaseI;$ $ATP = \overline{atp}.ADP   atp$ $Inside = \overline{siteI}Na.Inside + siteI(i).Inside$ $Outside = siteII(i).Outside + \overline{siteII}K.Outside$ $System = Inside   Outside   ATPase   !ATP$
---

**Table 2.** The  $\pi$ -calculus model

In the following we describe the mapping between the operational semantics of the  $\pi$ -calculus and the steps found in the description of the Albers-Post model. First, the Na ion binds to the ATPase in conformation  $E_1$ . This ion is communicated by the agent *Inside* on channel *siteI*. According to the reduction rules and the transition system of the  $\pi$ -calculus, the system evolution is given by

- $Inside \xrightarrow{\overline{siteI}Na} Inside$ , inferred with rules *pre* and *sum*
- $ATPase = PhaseI \xrightarrow{siteI(i)} [i = K]PhaseI + [i = Na]atp.\tau.\overline{siteII}i.PhaseII$ , inferred with rule *pre*
- $Inside | Outside | ATPase | !ATP \xrightarrow{\tau} Inside | Outside | [Na = K]PhaseI + [Na = Na]atp.\tau.\overline{siteII}Na.PhaseII | !ATP$ , by rules *com*, *par* applied to the above transitions.

Now, the pump will need energy to proceed any further. This step corresponds to the second one in the scalar diagram.

$$Inside | Outside | [Na = K]PhaseI + [Na = Na]atp.\tau.\overline{siteII}Na.PhaseII | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | \overline{\tau.siteII} Na.PhaseII | ADP | atp | !ATP,$$

deducted in several steps using *pre*, *par*, *com*, *match*. Under the new conditions the pump will suffer a conformation change. It will follow that

$$Inside | Outside | \overline{\tau.siteII} Na.PhaseII | ADP | atp | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | \overline{siteII} Na.PhaseII | ADP | atp | !ATP,$$

due to rules *pre*, *par*. It is now the case that the Na ion should be released into the medium outside the membrane. This is translated into the  $\pi$ -calculus as following

$$Inside | Outside | \overline{siteII} Na.PhaseII | ADP | atp | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | PhaseII | ADP | atp | !ATP,$$

because the Outside medium accepts any ions pushed through the membrane and regenerates its state afterwards. We are in the second phase of the process. The pump has a high affinity for the K ions and as a consequence it will accept any K ion available from the outside of the cell.

$$Inside | Outside | PhaseII | ADP | atp | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | [K = Na]PhaseII + [K = K] \overline{atp}.\overline{\tau.siteI} K.PhaseI | ADP | atp | !ATP$$

which is similar to the inference of the first transition. As evolving toward the beginning conformation, the pump must release the P link. We modeled this behaviour by book-keeping the previous interaction with the *ATP* molecule, specifically allowing for an *atp* kind of residue:  $ATP = \overline{atp}.ADP | atp$ .

$$Inside | Outside | [K = Na]PhaseII + [K = K] \overline{atp}.\overline{\tau.siteI} K.PhaseI | ADP | atp | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | \overline{\tau.siteI} K.PhaseI | ADP | !ATP,$$

again, several rules have helped to infer the transition. The pump changes conformation. It does that spontaneously without further interaction. We expressed that as a  $\tau$  transition.

$$Inside | Outside | \overline{\tau.siteI} K.PhaseI | ADP | !ATP$$

$$\xrightarrow{\tau}$$

$$Inside | Outside | \overline{siteI} K.PhaseI | ADP | !ATP$$



also, rule `par` must be used. In the last step, before the cycle reiterates, the pump releases the K ion inside the cell.

$$\begin{array}{c}
 \text{Inside} \mid \text{Outside} \mid \overline{\text{siteI}} K . \text{PhaseI} \mid \text{ADP} \mid !\text{ATP} \\
 \xrightarrow{\tau} \\
 \text{Inside} \mid \text{Outside} \mid \text{PhaseI} \mid \text{ADP} \mid !\text{ATP}
 \end{array}$$

Our model by using the  $\pi$ -calculus has the advantage of an automated tool for checking bisimilarity called Mobility Workbench. This means that we have a software (program) that is able to verify various properties expressed in a temporal logic or by using the weak open bisimulation. The advantage is that we can replace some quite expensive lab experiments with some inexpensive software experiments. Once the unnecessary details have been abstracted away, the computational model could be checked for various properties. We treat two properties: faithfulness of the model and verification of a certain invariant, respectively.

$  \begin{aligned}  \text{INFNRGY} &= \overline{\text{atp}} . \text{atp} . \text{INFNRGY} \\  \text{PhaseISpec} &= \text{siteI}(i) . ([i = K] \text{PhaseISpec} + [i = Na] \overline{\text{siteII}} i . \text{PhaseIISpec}) \\  \text{PhaseIISpec} &= \text{siteII}(i) . ([i = Na] \text{PhaseIISpec} + [i = K] \overline{\text{siteI}} i . \text{PhaseI}) \\  \text{ATPaseSpec} &= \text{PhaseISpec} \\  \text{PUMPNRGY} &= (\nu \text{atp}) (\text{ATPase} \mid \text{INFNRGY}) \\  \text{System} &= \text{Inside} \mid \text{Outside} \mid \text{ATPase} \mid \text{INFNRGY}  \end{aligned}  $
---

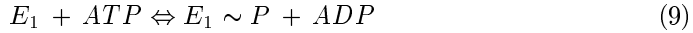
**Table 3.** The  $\pi$ -terms for verification

- $\neg \text{deadlocks}(\text{System})$ , in other words, the model is *deadlock* free
- $\text{PUMPNRGY} \approx \text{ATPaseSpec}$  - a pump with an infinite source of energy behaves externally, with regard to the flux of transported ions through the membrane as an agent whose carrier behaviour is desirable, simple and reflects the understanding of how the pump works. This is also a *liveness* property.

This automated verification is an important feature of modelling by using process algebras like CCS and  $\pi$ -calculus. The authors think that biology will largely benefit from the formal description of the molecular processes by using various process algebras, both for abstraction and for their verification tools.

## 4.2 Stochastic Approach

A more realistic approach in modeling the pump is to allow for pump misses. For instance if the reaction



is allowed to occur before the Na ion binds to the pump then a certain notion of nondeterminism is introduced. We'd like to impose the condition that the reaction (9) occurs very rarely compared to the usual work of the pump. We could do that using the stochastic  $\pi$ -calculus which enriches the usual prefixes of the calculus with probability distributions. This new calculus allows us to model the quantitative behaviour of the system. This is done by adding some probability distributions to the  $\pi$ -calculus prefixes. For instance, in Table 3 we add the same probability distribution to all the prefixes in order to keep the same quantitative information regarding the relative speeds of the actions performed. Then we can add a new branch to the term describing *PhaseI*, such that *PhaseI* becomes

$$\begin{aligned} & (siteI(i), \mathcal{F}_1).([i = K]PhaseI + [i = Na](atp, \mathcal{F}_1).(\tau, \mathcal{F}_1).(\overline{siteII}, \mathcal{F}_1) i.PhaseII) \\ & + (atp, \mathcal{F}_2).(\tau, \mathcal{F}_1).(\overline{siteII}, \mathcal{F}_1) i.PhaseII \end{aligned}$$

The probability of the transition

$$PhaseI \xrightarrow{(atp, \mathcal{F}_2)} (\tau, \mathcal{F}_1).(\overline{siteII}, \mathcal{F}_1) i.PhaseII$$

is  $\int_0^\infty \mathcal{F}_2'(t) \cdot (1 - \mathcal{F}_1(t)) dt$ . Moreover, considering a numerical example suggested in [9] wherein  $\mathcal{F}_1$  is an exponential distribution with parameter 2 and  $\mathcal{F}_2 = 1 - 2te^{-2t} - e^{-2t}$  is a two-level Erlang distribution, then the probability of the out-of-order transition is

$$\int_0^\infty 4te^{-2t}[e^{-2t}]dt = \int_0^\infty 4te^{-4t}dt = [(-\frac{1}{4} - t)e^{-4t}]_0^\infty = 0.25,$$

which means that in average there's one miss of the pump out of 4 execution cycles. Of course some more faithful parameters could have been used as well.

## 5 Conclusions

In recent years, various approaches from mathematics and computer science have been adapted for the representation of molecular processes. The use of the  $\pi$ -calculus or other process algebras to model the molecular interaction is quite new. The  $\pi$ -calculus is a widely accepted model of interacting systems with dynamically evolving communication topology. An important feature of the  $\pi$ -calculus is its mobility expressed by the changing configuration and connectivity among processes. This mobility increases the expressive power enabling the description of many high-level concurrent features. The  $\pi$ -calculus has a well-defined semantics and an appropriate algebraic theory. We think the  $\pi$ -calculus might be an adequate formalism for describing the biomolecular processes. In this way the knowledge for describing and analyzing behaviours of complex concurrent processes should be adapted and applicable to biological processes.

As far as we know, the first papers using the  $\pi$ -calculus in describing molecular processes were [1] and [2], followed by the successful papers [11, 9, 10] that represent and simulate biomolecular processes, describing the set of steps taken to adapt, extend and implement a core language to conform to the unique requirements of biochemical systems, first on a qualitative and then on a quantitative, stochastic scale. In [1], the  $\pi$ -calculus is used to describe the DNA methylation. In [2] are defined the so-called molecular structures and it is proved that they have the same expressive power as the  $\pi$ -calculus (which have the same computational power as Turing machines). A more detailed approach is presented in [4].

In this paper we motivate the use of the  $\pi$ -calculus as an adequate formalism for molecular processes by describing the dynamics of the sodium-potassium exchange pump, an important physiologic process present in all animal cells. This molecular process have to concern with phenomena related to distribution, cooperation, but with mobility and adaptability as well. Using the stochastic  $\pi$ -calculus, we describe the molecular interactions and conformational transformations in an explicit way. We manipulate formally the changing conformations and describe the corresponding dynamic systems using discrete mathematics instead of the usual partial differential equations. The transfer mechanisms are described in more details, step by step. Moreover, we can use some automated tools of verification developed for the  $\pi$ -calculus. This means that it would be possible to verify properties of the described systems by using a rather sophisticated software tool.

## Acknowledgments

Many thanks to Bogdan Tanasă for the biological information on the sodium pump. This work was supported by the academic research grant *Molecular Modeling* (R-252-000-120-112) from National University of Singapore.

## References

1. G. Ciobanu. Formal Description of the Molecular Processes. In: C. Martin-Vide, Gh. Păun (eds.): *Recent Topics in Mathematical and Computational Linguistics*, Editura Academiei, Bucharest, (2000) 82-96
2. G. Ciobanu. Molecular Structures. In: C. Martin-Vide, V. Mitrană (eds.): *Words, Sequences, Languages*, Kluwer, (2001) 299-317
3. G. Ciobanu, M. Rotaru. A  $\pi$ -calculus Machine. *Journal of Universal Computer Science*, 6(1), (2000)
4. G. Ciobanu, M. Rotaru. Molecular Interaction. *Journal of Theoretical Computer Science*, 289(1):801-827, (2002)
5. G. Ciobanu, B. Tanasă. Gene Expression by Software Mechanisms, *Fundamenta Informaticae*, 49:67-80, (2002)
6. G. Ciobanu, V. Ciubotariu, B. Tanasa. A pi-calculus Model of the Na Pump, *Genome Informatics*, Universal Academy Press, Tokyo, (2002) 469-472
7. R. Milner. *Communicating and Mobile Systems: the  $\pi$ -calculus*. Cambridge University Press, (1999)

8. R. Milner, J. Parrow, D. Walker. A calculus of mobile processes. *Journal of Information and Computation*, 100:1-77, (1992)
9. C. Priami, A. Regev, E. Shapiro, W. Silverman. Application of Stochastic Process Algebras to Bioinformatics of Molecular Processes. *Information Processing Letters*, 80:25-31, (2001)
10. A. Regev, E. Shapiro. Cellular Abstractions. *Nature*, 419:343, (2002)
11. A. Regev, W. Silverman, E. Shapiro. Representation and Simulation of Biochemical Processes Using the pi-calculus Process Algebra. *Proceedings Pacific Symposium on Biocomputing*, Vol.6, World Scientific Press, (2001) 459-470
12. B. Victor, F. Moeller. The Mobility Workbench: a Tool for the  $\pi$ -calculus, *Proceedings Computer-Aided Verification (CAV)*, Springer LNCS Vol.818, (1994) 428-440