

# P systems with dynamic channels transporting membrane vesicles

Rudolf FREUND      Marion OSWALD  
Faculty of Informatics  
Vienna University of Technology  
Vienna, Austria  
rudi@emcc.at      marion@emcc.at

## Abstract

Based on the biological model of cell-to-cell communication proposed by A. Rustom et al. in [5] we consider cells able to dynamically form connections (channels) between them according to specific constraints possibly relying on some attribute assigned to the membranes of the cells as well as of their contents. From a theoretical point of view, such P systems with dynamic channels transporting membrane vesicles are computationally complete even when using only some of the characteristic features of the general model. Also the efficiency of modelling specific processes in various application areas depends on the specific features of the chosen model of P systems with dynamic channels transporting membrane vesicles.

## 1 Introduction

Membrane systems were introduced in 1998 by Gh. Păun in [2] as a parallel distributed model of computation abstracted from cell functioning. In a membrane structure (that can be represented as a tree in P systems or as an arbitrary graph in the more general case of tissue P systems), multisets of objects can evolve according to given evolution rules. Many variants of membrane systems (P systems) have been considered so far, see [3] for a comprehensive overview and [4] for the actual status of research.

Like in the biological model of cell-to-cell communication proposed by A. Rustom et al. in [5], in this paper we consider cells (like in tissue P systems) able to dynamically form connections (channels or *nanotubes* in the sense of [5]) between them according to specific constraints possibly relying on some attributes assigned to the membranes of the cells as well as of their contents. In contrast to the P systems considered so far, through these nanotubes (channels) of P systems with dynamic channels transporting membrane vesicles,

multisets of elementary objects are transported in a membrane vesicle, which is the most important new feature of the model introduced in this paper. Attributes assigned to the cell membranes, for example, can be their electrical charges (polarizations) as this was formally already considered in P systems with active membranes. Moreover, the transport of a membrane vesicle through a nanotube between two cells may depend on the contents of the membrane vesicle itself as well as on the objects contained in the two cells connected by this nanotube and on the specific attributes assigned to the cell membranes and the membrane of the vesicle to be transported through the nanotube.

As it is often observed in the P systems area, processes may happen in parallel according to a universal clock; yet although in nature many processes are carried out in parallel, they are not synchronized. Therefore, to model these biological systems as described by A. Rustom et al., we consider P systems with dynamic channels transporting membrane vesicles also working in the asynchronous mode (an arbitrary number of rules that do not interfere with each other can be carried out in parallel in one derivation step) or in the sequential mode (exactly one rule is carried out in one derivation step); for an overview on P systems working in the asynchronous or in the sequential mode see [1]. On the other hand, processes in computers as for simulating mobile software agents can be simulated by P systems with dynamic channels transporting membrane vesicles working in the maximally parallel mode (which means that as many processes as possible are carried out in parallel).

In the second section, we now will introduce the general model of P systems with dynamic channels transporting membrane vesicles. From a theoretical point of view, P systems with dynamic channels transporting membrane vesicles are computationally complete, i.e., in the sense of Turing we can simulate the

actions of Turing machines; depending on the working mode (sequential, asynchronous or maximally parallel), different variants of rules and corresponding membrane attributes are needed to obtain this computational universality (we will prove universality for one specific variant working in the sequential mode in the third section). Also the efficiency of modelling specific processes in various application areas depends on the specific features of the chosen model of P systems with dynamic channels transporting membrane vesicles as we will discuss in the fourth section.

## 2 The General Model

In this section we describe the general model of P systems we are going to investigate in this paper. Intuitively, we consider cells enclosed by a membrane which allows for communication with the environment as well as for dynamically forming connections (channels or *nanotubes* in the sense of A. Rustom et al. in [5]) between cells that transport (multisets of) objects enclosed in a membrane (*vesicle*).

A *P system with dynamic channels transporting membrane vesicles* (in the following we shall use the notion *P system* only)  $\Pi$  is a construct

$$(O, O_T, O_\infty, C, Q, F, I, R)$$

where

- $O$  is the set of *objects*;
- $O_T \subseteq O$  is the set of *terminal* objects;
- $O_\infty \subseteq O$  is the set of objects occurring infinitely often in the environment of the cells;
- $C \subseteq O$  is the set of *catalysts*; a catalyst is never changed or moved by a rule;
- $Q$  is the set of *states* for the channels between cells;
- $F$  is the set of *attributes* assigned to a cell or a vesicle;
- $I$  specifies the initial contents of the environment (only those objects not occurring in an infinite number of copies, which were already specified by  $O_\infty$ ) as well as the cells and the vesicles and their initial contents the system starts with (observe that we assume no channels to exist at the beginning);
- $R$  is a set of *rules*.

The set of rules  $R$  itself consists of several sets of different types of rules:

- $R_{e,c}$  is a set of evolution rules for objects in a cell;
- $R_{e,v}$  is a set of evolution rules for objects in a vesicle;
- $R_{c,e}$  is a set of communication rules for exchanging (multisets of) objects between a cell and the environment;
- $R_{c,v}$  is a set of communication rules for exchanging (multisets of) objects between a vesicle in a cell and the surrounding cell;
- $R_{c,c,i}$  is a set of rules initializing a channel (nanotube) between two cells (or a cell and the environment);
- $R_{c,c,t}$  is a set of rules transporting a vesicle through a channel (nanotube) between two cells (or a cell and the environment);
- $R_{c,c,d}$  is a set of rules deleting a channel (nanotube) between two cells (or a cell and the environment; a vesicle sent out from a cell into the environment gets the status of a cell).

In some restricted variants, the sets  $R_{c,c,l}$ ,  $l \in \{i, t, d\}$ , may be combined in only one set  $R_{c,c}$  of rules which establish a channel between two cells for moving a vesicle from one cell to another one (or between a cell and the environment for expelling a vesicle into the environment) and immediately after having moved this vesicle deletes the channel again.

- $R_d$  is a set of rules eliminating a vesicle in a cell (in a biological sense, this deletion of a vesicle reflects a kind of phagocytosis);
- $R_g$  is a set of rules generating a new vesicle in a cell.
- $R_\delta$  is a set of rules eliminating the membrane of a vesicle and expelling its contents into the surrounding cell;
- $R_\mu$  is a set of rules generating a new vesicle containing the whole contents of a cell.

All the rules described above may depend on the features currently assigned to the involved cell(s) and/or the involved vesicle and possibly also change these features. In static variants of the general model, we omit the sets of rules  $R_d$ ,  $R_g$ ,  $R_\delta$ , and  $R_\mu$  and do not allow vesicles to be expelled into the environment.

Moreover we should like to point out that all the sets in the general model described above need not be finite, e.g., the set of objects may be the set of strings over a given alphabet, and the set of attributes may be infinite, too. If the sets of rules are infinite, we assume that given an instance of the P system, we are able to effectively list the rules applicable to this instance and to decide whether a set of rules chosen so far is maximal or not.

The P system  $\Pi$  may work in different derivation modes: In the maximally parallel mode, for each derivation step we choose such a subset of the rules in  $R$  that cannot be extended any more; in the asynchronous mode, an arbitrary number of rules is applied in parallel; in the sequential mode, exactly one rule is applied.

### 3 Theoretical Results

The general model of *P systems with dynamic channels transporting membrane vesicles* introduced in the previous section is very powerful from a theoretical point of view, i.e., not all possible features are needed to obtain computational completeness. Moreover, we may use these P systems for different tasks, e.g., as generating devices or as accepting devices (*P automata*), but also for modelling different processes carried out by other devices, yet needing only specific features of the general model for obtaining efficient simulation results. We here restrict ourselves to a model which works in the sequential mode and only uses communication rules (*antiport rules*) as well as, of course, rules for transporting vesicles from one cell to another one, and we show how in this restricted model we can simulate graph-controlled grammars (which are devices well known to be computationally complete (e.g., see [6])).

Now let us consider a finite set of elementary objects constituting  $O$  and a subset  $O_T$  to be the set of terminal objects. A *graph-controlled grammar*  $G_C$  then is a construct

$$(O, O_T, w, R, L_{in}, L_{fin})$$

where  $w$  is the initial multiset over  $O$ ,  $R$  is a finite set of rules  $r$  of the form  $(l(r) : p(l(r)), \sigma(l(r)), \varphi(l(r)))$ , with  $l(r) \in Lab(G_C)$ ,  $Lab(G_C)$  being a set of labels associated (in a one-to-one manner) with the rules  $r$  in  $R$ ,  $p(l(r))$  is a rewriting rule of the form  $a(r) \rightarrow u(r)$  with  $a(r) \in O$  and  $u(r)$  being a multiset over  $O$ ,  $\sigma(l(r)) \subseteq Lab(G_C)$  is the *success field* of the rule  $r$ , and  $\varphi(l(r)) \subseteq Lab(G_C)$  is the *failure field* of the

rule  $r$ ;  $L_{in} \subseteq Lab(G_C)$  is the set of initial labels, and  $L_{fin} \subseteq Lab(G_C)$  is the set of final labels. For  $r = (l(r) : p(l(r)), \sigma(l(r)), \varphi(l(r)))$  and  $v, u$  being two multisets over  $O$  we say that  $(u, k)$  is directly derivable from  $(v, l(r))$  if and only if

- **either**  $p(l(r))$  is applicable to  $v$ ,  $v$  is the result of the application of  $p(l(r))$  to  $u$ , and  $k \in \sigma(l(r))$ ,
- **or**  $p(l(r))$  is not applicable to  $v$ ,  $u = v$ , and  $k \in \varphi(l(r))$ .

The language generated by  $G_C$  is the set of all multisets  $u$  such that  $(u, k)$  can be derived in an arbitrary number of steps from  $(w, l)$  for some  $k \in L_{fin}$  and  $l \in L_{in}$ .

We now define a restricted variant of the general model of P systems with dynamic channels transporting membrane vesicles that allows for the simulation of graph-controlled grammars: Let

$$(O, O_T, w, R, L_{in}, L_{fin})$$

be a graph-controlled grammar (without loss of generality, we may assume  $L_{in}$  and even  $L_{fin}$  to contain only one label and, moreover,  $l(r) \notin \sigma(l(r))$  for all  $r$ ). Then we construct the P system with dynamic channels transporting membrane vesicles  $\Pi$  working in the sequential mode

$$(O, O_T, O_\infty, C, Q, F, I, R_\Pi)$$

as follows:

We do not use catalysts, and we do not need states for the channels to be created dynamically; therefore, we omit  $C$  and  $Q$ , as well as  $O_\infty$ , because we assume all objects from  $O$  to be available in the environment in an unbounded number, hence,  $\Pi$  is specified as a quintuple  $(O, O_T, F, I, R_\Pi)$  only.

As set of attributes  $F$  we take  $Lab(G_C) \cup \{-, 0, +\}$ , where we assume  $Lab(G_C) \cap \{-, 0, +\} = \emptyset$ . The initial configuration consists of  $card(Lab(G_C))$  cells having the labels from  $Lab(G_C)$  as attributes; moreover, the cell with attribute  $l(r)$  contains the multiset  $u(r)$ , and finally, the initial cell (having the initial label from  $L_{in}$  as attribute) also contains a vesicle with attribute 0 and the contents  $w$  (i.e., the initial multiset from  $G_C$ ). The rules in  $R_\Pi$  simulate the rules in  $R$  as follows:

- $(l(r), 0, a(r)/u(r), +) \in R_{c,v}$ : In the cell with attribute  $l(r)$ ,  $a(r)$  in a vesicle (with attribute 0 which by the application of this rule is changed to  $+$ ) contained in this cell is replaced by  $u(r)$  which is taken from the cell and regained by

- $(l(r), a(r)/u(r)) \in R_{c,e}$ ; in that way, one application of the rule  $r$  is simulated.
- $(l(r), 0, \neg a(r), \lambda/\lambda, -) \in R_{c,v}$ : Provided that the symbol  $a(r)$  is not present in a vesicle contained in the cell with attribute  $l(r)$ , the attribute of the vesicle is changed from 0 to  $-$  (without communicating symbols between the vesicle and the cell, which is indicated by the notion  $\lambda/\lambda$ , where  $\lambda$  denotes the empty word).
- $R_{c,c}$  consists of all rules of the form  $(i, k, j, 0)$  such that
  - $k = +$  and  $j \in \sigma(i)$  or
  - $k = -$  and  $j \in \varphi(i)$ .

The application of  $(i, k, j, 0)$  means establishing a channel between the cells with attributes  $i$  and  $j$  for transporting a vesicle with attribute  $k$  from cell  $i$  to cell  $j$  thereby changing the attribute of the vesicle from  $k$  to 0.

As results of a computation in  $\Pi$  we take the contents of any vesicle which appears in the final cell (i.e., the cell carrying the attribute from  $L_{fin}$ ) and contains only terminal symbols. In that way  $\Pi$  generates the same set of terminal multisets as  $G_C$ .

As is well-known (see the results cited in [1]), P systems with antiport rules are computationally complete in only one membrane when working in the maximally parallel mode, whereas in the sequential mode universality cannot be reached even with an arbitrary number of membranes. The proof sketched above takes advantage of the inherently new feature of P systems with dynamic channels transporting membrane vesicles allowing for transporting the complete multiset of symbols in a vesicle from one cell to another one.

## 4 Variants for Applications

The biologically motivated model of P systems with dynamic channels transporting membrane vesicles introduced in this paper can be used for modelling various mechanisms in different application areas. Again the main feature we have to take advantage of is the possibility to establish connections between cells and to transport whole multisets of objects from one cell to another one in a single step.

As the model of P systems with dynamic channels transporting membrane vesicles incorporates several other features well motivated from biology, it can be used for modelling various aspects of artificial life. The

cells themselves are organisms that may interact in an unguided manner via the environment or else, in a direct way, by establishing channels for exchanging complete sub-organisms (vesicles). By using suitable evolution rules, the organisms may develop also based on the development (evolution) of their components mostly represented by the vesicles contained in them, which usually happens in a parallel but unsynchronized way (and which corresponds with the asynchronous working mode of the P systems).

On the other hand, the vesicles can also be seen as packages of information transported from one computer (cell) to another one or as autonomous agents performing their tasks on different servers. In that case, these agents work in parallel, either in an unsynchronized way (which corresponds with the asynchronous mode of the P system) or even in a synchronized way (which corresponds with the maximally parallel mode of the P system); using sufficiently powerful operations (evolution rules) in the vesicles, we are able to describe the processes taking place in distributed systems.

## Acknowledgements

The work of Marion Oswald was supported by FWF-project T225-N04.

## References

- [1] R. Freund, Asynchronous P systems, Proceedings WMC5 (2004), 12-28
- [2] Gh. Păun, Computing with membranes. Journal of Computer and System Sciences 61 (1), 108-143
- [3] Gh. Păun, Membrane computing - an introduction, Springer-Verlag, Berlin, 2002
- [4] The P systems webpage:  
<http://psystems.disco.unimib.it>
- [5] A. Rustom, R. Saffrich, I. Markovic, P. Walther, HH. Gerdes, Nanotubular highways for intercellular organelle transport, Science 303 (2004), 1007-1010
- [6] A. Salomaa, G. Rozenberg (eds.), Handbook of Formal Languages, Springer-Verlag, Berlin, 1997