Mutual Mobile Membranes Systems with Surface Objects

Bogdan Aman, Gabriel Ciobanu

[1] Romanian Academy, Institute of Computer Science
[2] A.I.Cuza University of Iaşi, Romania
baman@iit.tuiasi.ro, gabriel@info.uaic.ro

Summary. In this paper we introduce mutual mobile membranes with surface objects, systems which have biological motivation. In P systems with mobile membranes with surface objects, a membrane may enter or exit another membrane. The second membrane just undergoes the action, meaning that it has no control on when the movement takes place. This kind of movement illustrates the lack of an agreement (synchronization) similar to an asynchronous evolution. In mutual mobile membranes with surface objects this aspect is adjusted: any movement takes place only if both participants agree by synchronizing their evolution. In membranes two kinds of competition can occur: resource competition and location competition. Resource competition refers to rules which request the same resources, and the available resources can only be allocated to some of the rules. Location competition refers to the movement of a membrane in the hierarchical structure of the membrane systems under the request of some conflict rules. We use the two variants of membrane systems in order to describe and explain these kinds of competition, and introduce synchronizing objects in mutual mobile membranes which will help to solve the resource and location competitions.

1 Introduction

Two recent computational models have been inspired from the structure and the functioning of the living cell: membrane systems [16, 17] and brane calculus [8]. Although the models start from the same observation, they are build having in mind different goals: membrane systems investigate formally the computational nature and power of various features of membranes, while the brane calculus is intended to give a faithful and intuitive representation of the biological reality. In [9] the initiators of these two formalisms describe the goals they had in mind: "While membrane computing is a branch of natural computing which tries to abstract computing models, in the Turing sense, from the structure and the functioning of the cell, making use especially of automata, language, and complexity theoretic tools, brane calculi pay more attention to the fidelity to the biological reality, have as a primary target systems biology, and use especially the framework of process algebra."

A *membrane system* consists of a hierarchy of membranes which do not intersect, with a distinguishable membrane called *skin* surrounding all of them. A membrane without any other membranes inside is *elementary*, while a non-elementary membrane is a *composite* membrane. The membranes define demarcations between *regions*; for each membrane there is a unique associated region. Since we have a one-to-one correspondence, we sometimes use membrane instead of region, and vice-versa. The space outside the skin membrane is called the environment. Regions contain multisets of *objects*, *evolution rules* and possibly other membranes. Only rules in a region delimited by a membrane act on the objects in that region. More details about membrane systems can be found in [17].

Exocytosis is the movement of materials out of a cell via membranous vesicles. These processes allow patches of membrane to flow from compartment to compartment, and require us to think of a cell as a dynamic, rather than static, structure. *Endocytosis* is a general term for a group of processes that bring macromolecules, large particles, small molecules, and even small cells into the eukaryotic cell. There are three types of endocytosis: *pinocytosis*, *phagocytosis* and *receptor-mediated endocytosis*. In all three, the plasma membrane folds inward around materials from the environment, forming a small pocket. The pocket deepens, forming a vesicle. This vesicle separates from the plasma membrane and migrates with its contents to the cell interior.

In brane calculus we have a membrane structure, in which the membranes represent the sites of activity. Opposite to the initial classes of membrane systems in which a computation took place inside the membranes, in brane calculi a computation happens on the membrane. The operations of the two basic brane calculi are directly inspired by biologic processes such as endocytosis, exocytosis and mitosis. The calculus formed using *pino, exo, phago* operations is more expressive then the calculus formed by *mate, drip, bud*, because we can simulate the latter operations using the former ones. Another difference regarding the semantics is expressed in [6]: "whereas brane calculi are usually equipped with an interleaving, sequential semantics (each computational step consists of the execution of a single instruction), the usual semantics in membrane computing is based on maximal parallelism (a computational step is composed of a maximal set of independent interactions)."

Some work was done trying to relate these two models [6, 7, 10, 11]. Inspired by brane calculus, a model of the membrane system having objects attached to the membranes has been introduced in [9]. In [5], a class of membrane systems containing both free floating objects and objects attached to membranes have been proposed, while in [20] a simulation of a bounded symport antiport membrane system using brane calculus is proposed. In [2] we have related membrane computing with brane calculi, namely a translation of the PEP class into a special class of membrane systems, while in this paper we introduce a membrane system having objects and co-objects attached to the membrane.

The structure of the paper is as follows. In Section 2 we define the class of membrane systems with surface objects and co-objects together with the biological

motivation for the rules used. In Section 3 we present the notions of competitions in membrane systems. Conclusions and references end the paper.

2 Mutual Membrane Systems with Surface Objects

The phospholipid bilayer serves as a lipid "lake" in which some proteins "float" (see Figure 1).

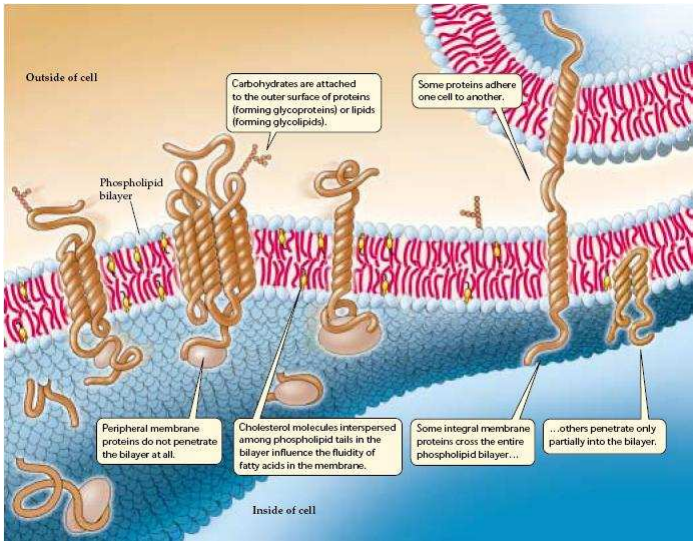

Fig. 1. The Fluid Mosaic Model: The general molecular structure of biological membranes is a continuous phospholipid bilayer in which proteins are embedded.

2.1 Endocytosis and Exocytosis in Biology

Endocytosis is a general term for a group of processes that bring macromolecules, large particles, small molecules, and even small cells into another cell. There are three types of endocytosis: phagocytosis, pinocytosis, and receptor-mediated endocytosis. In all three, the membrane invaginates (folds inward) around materials from the environment, forming a small pocket. The pocket deepens, forming a vesicle. This vesicle separates from the membrane and migrates with its contents to the cell's interior.

In phagocytosis ("cellular eating"), part of the membrane engulfs large particles or even entire cells. Phagocytosis is used as a cellular feeding process by

unicellular protists and by some white blood cells that defend the body by engulfing foreign cells and substances. In pinocytosis ("cellular drinking"), vesicles also form. However, these vesicles are smaller, and the process operates to bring small dissolved substances or fluids into the cell. Like phagocytosis, pinocytosis is relatively nonspecific as to what it brings into the cell.

Receptor-mediated endocytosis (Figure 2) is used by animal cells to capture specific macromolecules from the cell's environment. This process depends on receptor proteins, integral membrane proteins that can bind to a specific molecule in the cell's environment. The uptake process is similar to nonspecific endocytosis, as already described. However, in receptor-mediated endocytosis, receptor proteins at particular sites on the extracellular surface of the plasma membrane bind to specific substances. These sites are called coated pits because they form a slight depression in the plasma membrane. The cytoplasmic surface of a coated pit is coated by proteins, such as clathrin.

When a receptor protein binds to its specific macromolecule outside the cell, its coated pit invaginates and forms a coated vesicle around the bound macromolecule. Strengthened and stabilized by clathrin molecules, this vesicle carries the macromolecule into the cell.

Since only the receptor-mediated endocytosis uses receptors and co-receptors we are interested only in modeling this process.

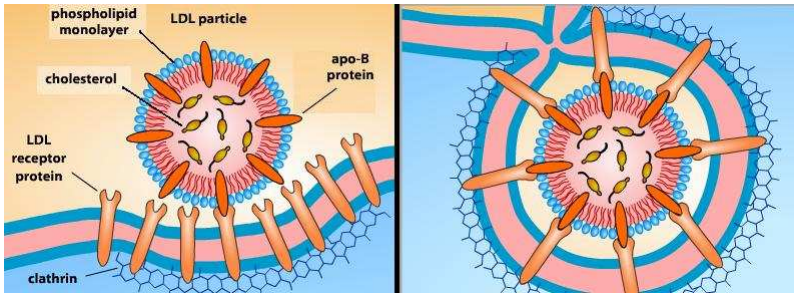

Fig. 2. Receptor-mediated endocytosis

Exocytosis (Figure 3) is the movement of materials out of a cell via membranous vesicles. These processes allow patches of membrane to flow from compartment to compartment, and require us to think of a cell as a dynamic, rather than static, structure. SNARES (Soluble NSF (N-ethylmaleimide Sensitive Factor) Attachment Protein Receptor) located on the vesicles (v-SNARES) and on the target membranes (t-SNARES) interact to form a stable complex that holds the vesicle very close to the target membrane.

The B lymphocyte cell searches for antigen matching its receptors. If it finds such antigen, it connects to it, and inside the B cell a triggering signal is set off. In order to become fully activated a B cell needs proteins produced by helper T cells. This is simulated using mutual contextual evolution (Figure 4).

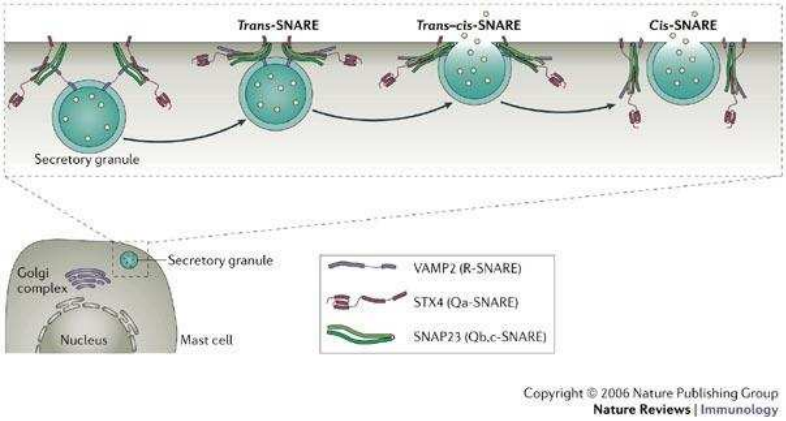

Fig. 3. SNARE-mediated exocytosis

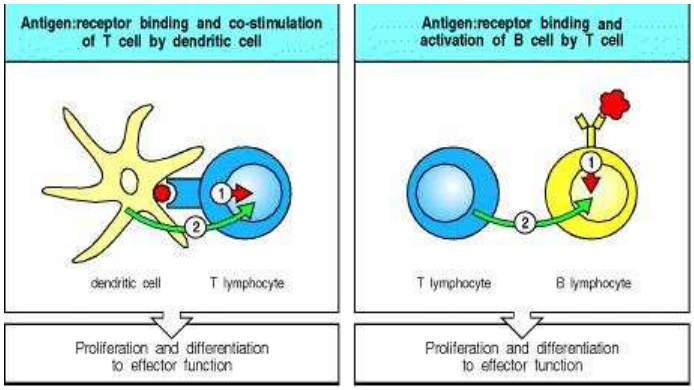

Fig. 4. Contextual evolution

This provides a biological motivation of using objects and co-objects for the exo, endo and contextual evolution rules.

We define now the membrane systems with surface objects and co-objects. Let $\mathbb{N}$ be a set of positive integers, and consider a finite alphabet Γ of symbols. A multiset over Γ is a mapping $u : \Gamma \to \mathbb{N}$. The empty multiset is represented by λ. For any $a \in \Gamma$, the value $u(a)$ denotes the multiplicity of a in u (i.e., the number of occurrences of symbol a in u). Given two multisets u, v over Γ, for any $a \in \Gamma$, we have $(u \uplus v)(a) = u(a) + v(a)$ as the multiset union, and $(u \backslash v)(a) = max\{0, u(a) - v(a)\}$ as the multiset difference. We use the string representation of multisets used in the membrane systems. An example of such a representation $u = aabca$, where $u(a) = 3$, $u(b) = 1$, $u(c) = 1$. Using such a representation, the operations over multisets are defined as operations over strings.

2.2 Definition

Definition 1. *A* mutual mobile membrane system with surface objects *is a construct*

$$\Pi = (P, \mu, w_1, \ldots, w_n, R, i_O)$$

where:

1. $n \geq 1$ *(the initial* degree *of the system);*
2. P *is an alphabet of proteins (its elements are called* objects*);*
3. i_O *is the output membrane;*
4. μ *is a* membrane structure*, consisting of* n *membranes. A membrane structure is a hierarchically arranged set of membranes, where we distinguish the external membrane (usually called the "skin" membrane) and several internal membranes; a membrane without any other membrane inside it is said to be elementary;*
5. $w_1, \ldots, w_n$ *are strings over* V*, describing the initial markings of the* n *membranes of* μ*;*
6. R *is a finite set of* developmental rules*, of the following forms:*
   a) $[\ ]_{uv}[\ ]_{\overline{u}v'} \rightarrow [[\ ]_w]_{w'}$ *for* $u, \overline{u}, w, w' \in P^+; v, v' \in P^*$

   mutual endocytosis

   An elementary membrane containing the multiset uv *on the membrane enters the adjacent membrane containing the multiset* $\overline{u}v'$ *on the membrane; the multisets* uv *and* $\overline{u}v'$ *are transformed into multisets* w *and* w' *during the evolution;*
   b) $[[\ ]_{uv}]_{\overline{u}v'} \rightarrow [\ ]_{uw}[\ ]_{\overline{u}w'}$*,* $u, \overline{u}, w, w' \in P^+; v, v' \in P^*$

   mutual exocytosis

   An elementary membrane containing the multiset uv *on the membrane exits the adjacent membrane containing the multiset* $\overline{u}v'$ *on the membrane; the multisets* uv *and* $\overline{u}v'$ *are transformed into multisets* w *and* w' *during the evolution;*
   c) $[\ ]_{uv}[\ ]_{\overline{u}v'} \rightarrow [\ ]_w[\ ]_{w'}$*,* $u, \overline{u}, w, w' \in P^+; v, v' \in P^*$

   mutual contextual evolution

   The multisets of objects uv *and* $\overline{u}v'$ *placed on two sibling membranes are transformed into the multisets* w *and* w'*.*

The rules are applied according to the following principles:

1. All rules are applied in parallel, non-deterministically choosing the rules, the membranes, and the objects, but in such a way that the parallelism is maximal; this means that in each step we apply a set of rules such that no further rule can be added to the set.
2. The membrane containing the multiset u on it from the rules of type $(a) - (b)$ is said to be active, while the membrane containing the multiset $\overline{u}$ on it is said to be passive. In any step of a computation, any object and any active membrane can be involved in at most one rule, but the passive membranes

are not considered involved in the use of the rules (hence they can be used by several rules at the same time as passive membranes).
3. When a membrane is moved across another membrane, by endocytosis or exocytosis, its whole contents (its objects) are moved.
4. If a membrane exits the system (by exocytosis), then its evolution stops.
5. All objects and membranes which do not evolve at a given step (for a given choice of rules which is maximal) are passed unchanged to the next configuration of the system.

3 Competitions

We start with an example from [19] which motivates biologically the study of competitions.

Example 1. Bacteriophage λ is a temperate phage, meaning that it can undergo either a lytic or a lysogenic cycle (see Figure 5). When there is a rich medium available and its host bacterium is growing rapidly, the prophage takes advantage of its favorable cellular environment and remains lysogenic. When the host bacteria are not as healthy, the prophage senses this and, as a survival mechanism, leaves the host chromosome and becomes lytic.

The phage makes this decision by means of a "genetic switch": Two regulatory viral proteins, labeled cI and Cro, compete for two operator/promoter sites on phage DNA. The two operator/promoter sites control the transcription of the viral genes involved in the lytic and the lysogenic cycles, respectively, and the two regulatory proteins have opposite effects on the two operators (Figure 2). Phage infection is essentially a "race" between these two regulatory proteins. In a healthy E. coli host cell, Cro synthesis is low, so cI "wins" and the phage enters a lysogenic cycle. If the host cell is damaged by mutagens or other stress, Cro synthesis is high, promoters for phage DNA and viral coat proteins are activated, and bacterial lysis ensues.

Remark 1. Rarely, two viruses infect a cell at the same time. This is an unusual event, as once an infection cycle is under way, there is usually not enough time for an additional infection. In addition, an early protein prevents further infections in some cases. In this case the protein is the one who prevents further races inside the infected cell.

In concurrency, a competition occurs when more than one rule are engaged for the same resources. Here is an example:

Example 2. Suppose at a given moment of computation we have a membrane containing the multiset of objects aa and two rules: $aa \to b$ and $aa \to c$. We observe that the rules have similar left objects. The application of one of the rules blocks the application of the other rule.

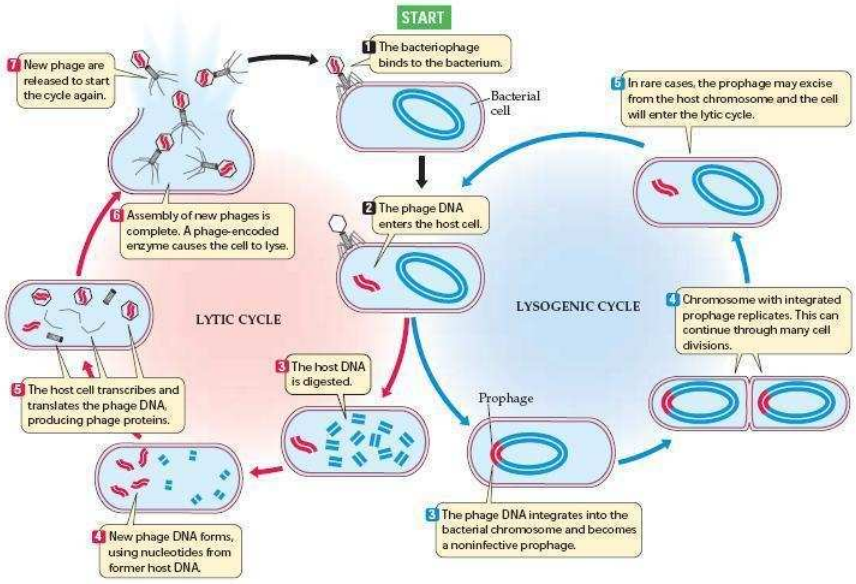

Fig. 5. The Lytic and Lysogenic Cycles of Bacteriophage: Infection by viral DNA leads to the multiplication of the virus and lysis of the host bacterial cell. In the lysogenic cycle, an inactive prophage is replicated as part of the host's chromosome.

This kind of competitions are called *resource competitions*. Such a competition occurs when two rules try to use at least one common resource. Resource competitions are desirable in membrane computing for simulating non-deterministic behaviors. In mutual mobile membranes we also find a different form of competition, namely the *location competition*. Consider the following example:

Example 3. Suppose at a given moment of the computation we have the following membrane configuration: $[\,][\,]_a[\,]_b$, and two rules: $[\,][\,]_a \to [[\,]_c]$ and $[\,][\,]_b \to [[\,]_d]$. We observe that membrane containing a is included in the left part of both rules. We have two possible evolutions:

- If we first apply the rule $[\,][\,]_a \to [[\,]_c]$, we obtain the membrane structure $[[\,]_c][\,]_b$ and the other rule cannot be applied anymore.
- If we first apply the rule $[\,][\,]_b \to [[\,]_d]$ we obtain the membrane structure $[\,][[\,]_d]_a$ and the other rule cannot be applied since now membrane n is not elementary. We refer to the rules of mobile membrane systems in which only elementary membranes can pass through other membranes [12].

The choice of which rule is applied first makes the other rule useless, so the system should be described initially with only one of these rules, depending on the expected evolution.

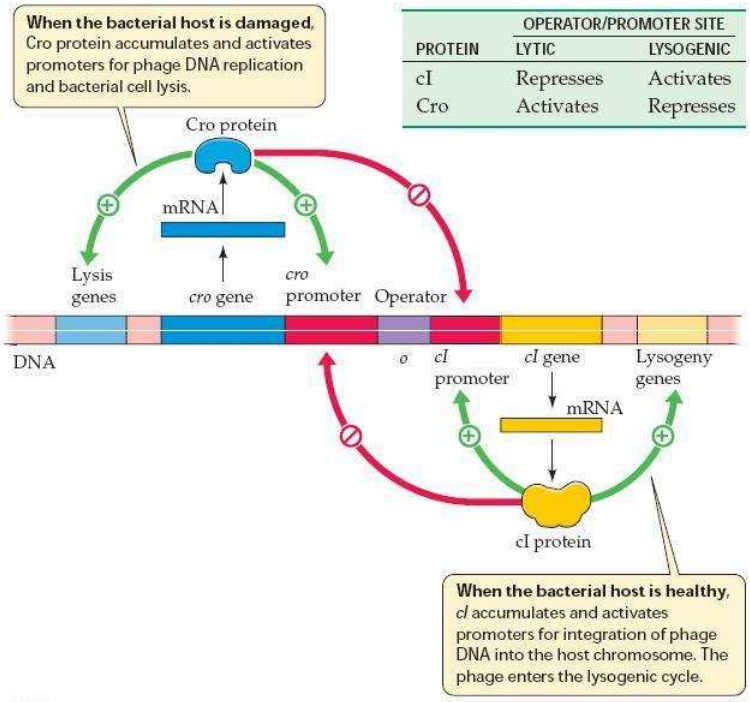

Fig. 6. Control of Phage λ Lysis and Lysogeny: *Cro* and *cI* compete for the operator/promoter sites controlling the gene transcription for viral lysis and lysogeny.

The movement in mutual mobile membrane systems is *local*, namely a membrane can only interact with neighboring membranes. Locality of movement implies that a membrane written to accomplish a certain task in a given membrane, it shall not work correctly in another membrane.

The location competition raises some problem:

1. it is difficult to describe membrane systems which behave as expected in all contexts;
2. it is difficult to prove behavioral properties of membrane systems.

The rules of mobile membranes allow a membrane to enter, or to exit another membrane. The second membrane just undergoes the action, meaning that it has no control on when the movement takes place. As a consequence, it is hard to control the resources inside a given membrane. By defining mutual mobile membranes this is rectified: any movement takes place only if both participants agree. This is achieved by using objects and co-objects to control the movement. The inspiration comes from biology where we have receptors and co-receptors which

control the interaction between membranes. Location competitions also exist in mutual mobile membranes, but they are easier to detect.

4 Conclusion

In the area of membrane computing the authors usually consider that a system is synchronous if the rules are applied in a maximally parallel manner, otherwise it is asynchronous. On the other hand, in process algebra the authors consider that two processes are synchronized if they interact by using actions and co-actions. We adapt the second approach to membrane systems, by replacing the actions and co-actions with objects and co-objects.

As related work we can mention [13] and [14] where the authors study the plain and grave interferences which appear in mobile ambients, and try to remove them by defining safe mobile ambients and an appropriate type system. The work presented in this paper corresponds to the first step described in their work, more exactly we define the competitions in mutual mobile membranes. Further work will include the use of a type system for mutual mobile membranes in order to limit the competitions which appear during the evolution of a membrane system.

Regarding the asynchronous aspects, we define in [4] a compositional asynchronous membrane system based on a handshake mechanism implemented by using antiport rules and promoters. Such a system is used to evaluate arithmetical expressions starting from simple membranes for addition, subtraction, multiplication and division.

References

1. B. Aman, G. Ciobanu: Structural properties and observability in membrane systems. *SYNASC*, IEEE Computer Society, 2007, 74–84.
2. B. Aman, G. Ciobanu: Membrane systems with surface objects. *Proceedings of the International Workshop on Computing with Biomolecules (CBM 2008)*, 2008, 17–29.
3. C. Bodei, A. Bracciali, D. Chiarugi: Control flow analysis for brane calculi. *Electronic Notes in Theoretical Computer Science*, 227 (2009), 59–75.
4. C. Bonchis, C. Izbasa, G. Ciobanu: Compositional asynchronous membrane systems. *Progress in Natural Science*, 17, 4 (2007), 411–416.
5. R. Brijder, M. Cavaliere, A. Riscos-Núñez, G. Rozenberg, D. Sburlan: Membrane systems with marked membranes. *Electronic Notes in Theoretical Computer Science,* 171, 2 (2007), 25–36.
6. N. Busi: On the computational power of the mate/bud/drip brane calculus: Interleaving vs. maximal parallelism. *Workshop on Membrane Computing*, LNCS 3850, Springer, 2006, 144–158.
7. N. Busi, R. Gorrieri: On the computational power of Brane calculi. *Third Workshop on Computational Methods in Systems Biology*, 2005, 106–117.
8. L. Cardelli: Brane calculi. Interactions of biological membranes. *Lecture Notes in BioInformatics*, 3082, Springer, 2004, 257–278.

9. L.Cardelli, Gh. Păun: An universality result for a (mem)brane calculus based on mate/drip operations. *ESF Exploratory Workshop on Cellular Computing (Complexity Aspects)*, Sevilla, 2005, 75–94.
10. M. Cavaliere, S. Sedwards: Membrane systems with peripherial proteins: Transport and evolution. *Electronic Notes in Theoretical Computer Science*, 171, 2 (2007), 37–53.
11. S.N. Krishna: Universality results for P systems based on brane calculi operations. *Theoretical Computer Science*, 371 (2007), 83–105.
12. S.N. Krishna, Gh. Păun: P systems with mobile membranes. *Natural Computing*, 4, 3 (2005), 255–274.
13. F. Levi, D. Sangiorgi: Controlling interference in ambients. *Principles of Programming Languages*, 2000, 352–364.
14. F. Levi, D. Sangiorgi: Mobile safe ambients. *Transactions on Programming Languages and Systems*, 25, 1 (2003), 1–69,.
15. R. Milner: *Communicating and Mobile Systems: The pi-calculus*. Cambridge University Press, 1999.
16. Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143.
17. Gh. Păun: *Membrane Computing. An Introduction*. Springer, 2002.
18. Gh. Păun: Membrane computing and brane calculi (Some personal notes). *Electronic Notes in Theoretical Computer Science*, 171 (2007), 3–10.
19. W.K. Purves, D. Sadava, G.H. Orians, H.C. Heller: *Life: The Science of Biology*, 7th Edition, W.H. Freeman & Co, 2004.
20. A. Vitale, G. Mauri, C. Zandron: Simulation of a bounded symport antiport P system with brane calculi. *BioSytems*, 91, 3 (2008), 558–571.