Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 171 (2007) 3–10
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Membrane Computing and Brane Calculi (Some Personal Notes)

Gheorghe Păun

Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucureşti, Romania
and
Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
george.paun@imar.ro, gpaun@us.es

Abstract

This is a very short discussion of similarities and dissimilarities between membrane computing and brane calculi, insisting mainly on some recent ideas of bridging the two areas of research (from this last point of view, the bibliography tries to be as complete as possible) and proving that, in fact, there is no real dissimilarity...

Keywords: Membrane computing, P systems, Brane calculi, Computational biology

1 Introduction

"Membrane computing is a young[1] branch of natural computing, initiated in [23], aiming to abstract computing ideas, models, paradigms from the structure and functioning of the cell and from the cells organization in tissues, organs, and other higher order structures. The obtained devices, currently called P systems, are parallel distributed computing models, processing multisets in compartments defined by membranes. There were investigated many classes of P systems and most of them are computationally complete; when an exponential workspace can be created in a polynomial time, e.g., by membrane division, then polynomial (often, linear) solutions to computationally hard problems (typically, **NP**-complete problems) can be devised."

This paragraph can be found almost in this form in the introduction of many papers in the membrane computing area; very frequent was and still is the illus-

[1] Not anymore: eight years is a long time for a bio-inspired research area, and this is also proved by the data from [29] (bibliography, events, results, applications).

doi:10.1016/j.entcs.2007.05.003

tration of the notion of a membrane structure (and of the associated terminology) from Figure 1, a sort of logo of the domain.

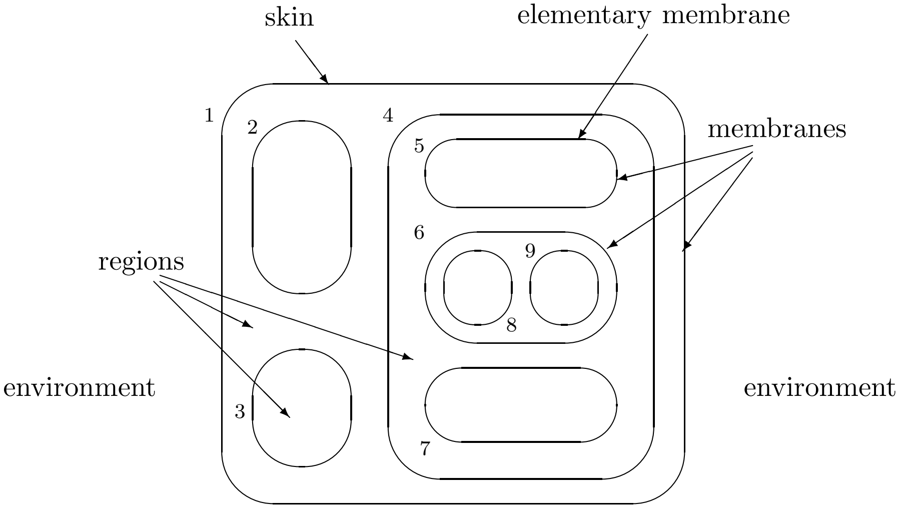

Fig. 1. A membrane structure

In a great extent, these phrases and this figure capture the essence of membrane computing. It should be added another slogan of the field, stating that "the rules are used in the non-deterministic maximally parallel way", and that recently membrane computing proved to be a very promising framework for devising models for biology (and other areas, such as economics and linguistics) – with surprising applications in areas never sought, such as approximate optimization, in the sense of evolutionary computing.

We already have here a list of features of membrane computing which are similar or different from corresponding features of brane calculi[2] .

The next section is devoted to explicitly listing some (no such list can be complete) of these common or different features of the two areas. In principle, each of them is an invitation to extend ingredients from one domain to another one, and in many cases such bridging investigations already started.

2 Things in Common, Things Different

Of course, no ordering is assumed below among the concepts invoked, as well as no pretention of completeness. Furthermore, no technical details are given, as the reader is assumed to be familiar with basic elements of both membrane computing and brane calculi.

[2] This is indeed a young direction of research, [7]. Actually, I am considering here only the two brane calculi from [7], based on *pino, exo, phago*, respectively, *mate, drip, bud* operations, although L. Cardelli and his collaborators have considered also other process algebra calculi before or after [7]: see, e.g., [27] and [12].

- *Objects in regions, objects on membranes:* This is a much mentioned difference, which, actually, is not totally valid. Even in the standard setup, of so-called transition P systems with symbol-objects, it is possible to assume that we have objects bound on the membrane, by simply assuming that the membrane is a bilayer with an interspace between layers. Actually, this was done already in [20], and was also briefly investigated in Section 9.3 of [24]. Sure, this trick with making a compartment from the membrane itself, raises a series of restrictions in handling the respective objects (and cannot be used for formalizing operations with membranes).
- Anyway, at the first sight, *the role of objects* placed on membranes is different in the two areas: in membrane computing the focus is on the evolution of objects themselves, while in brane calculi the objects ("proteins") mainly control the evolution of membranes. These protein-objects also evolve at the same time, and this is a very promising research topic, up to now probably the most fruitful and most investigated import of ideas from brane calculi to membrane computing: considering P systems with objects on membranes used (i) to control brane calculi operations with membranes, but with the multisets of protein-objects being the main data structure on which the computation works, or (ii) to control usual multiset processing rules for objects placed in compartments, maybe also evolving the objects bound on membranes. The first idea is followed in [9], [1], [16], [25] (in this last paper, the operations are not exactly those from brane calculi, in the sense that their semantics – the way of moving the contents of membranes – is different, that is why the operations were called *create* and *dissolve*), while the second one is investigated in [10], [21], [22], [17]. In [10] the "proteins" can be attached to and detached from membranes, while in the other papers they remain always bound on membranes.
- *The focus on operations with membranes* is again only at a superficial view a difference between the two areas. While the basic data structure of membrane computing is indeed the multiset of symbol-objects (but also sets of strings, and sporadically also sets of symbols or numerical variables), rules for handling membranes were always used, starting with [23], where the operation of membrane dissolving was introduced, for a while considered an "innocent" operation which however proved to be crucial in investigations related to complexity matters (see, e.g., [15]). Membrane creation and membrane division are basic ways for creating the exponential workspace for solving **NP**-complete (recently, also **PSPACE**-complete) problems in polynomial time using P systems. Merging, separating, gemmating, and other operations with membranes were also used in a series of papers. There are also papers (we refer to the bibliography from [29] for details) where the membrane structure itself is the goal of the computations, e.g., as a description of a tree, or a way to describe strings in a context-free language. A systematic investigation in this respect was started in [11], where a sort of AFL (Abstract Family of Languages) theory of membrane structures and their operations was proposed – still waiting for further research efforts.
- *Synchronous versus asynchronous computing:* This was many times invoked as

the main difference between membrane computing and brane calculi, and again the assertion is only partially valid. It is true that most papers in membrane computing deal with synchronized systems, having a general clock which marks the time for the whole system and in each time unit each compartment has to evolve in a maximal way. This assumption starts from the observation that biology "has a high level of parallelism", but, of course, it is mathematically grounded/useful: the synchronization (especially, the maximal parallelism, alone or together with the halting condition in defining successful computations) allows "appearance checking" (in the terminology of regulated rewriting) and "check for zero" (in the terminology of register/counter machines), hence this is a shortcut towards Turing universality[3]. There are however a large number of papers where the computations are sequential – see [13] and its references.

There also are variants/degrees of parallelism. O.H. Ibarra and his group has investigated P systems with a bounded parallelism. M. Cavaliere and D. Sburlan have considered time-free, clock-free, time-independent systems. G. Ciobanu and his group has considered both sequential and time-independent systems. Again, I refer to [29] for references.

- *Halting computations:* Starting with Turing machines, this is a standard way to define successful computations and it is introduced in membrane computing mainly to have a simple and powerful condition for defining the moment when the result of a computations "is ready", but for sure it is neither essential nor obligatory – although so frequently used. The end of a computation can also be signaled by means of *events* of various kinds, and this was done in several papers: P. Frisco uses frequently an acknowledging membrane (initially empty; the result of a computation is defined in the moment when any object enter this membrane), in general, in the case of P systems with string-objects one does not work with halting computations, events were considered recently by O.H. Ibarra, A. Păun and their collaborators, while in neural-like P systems and P automata in the sense of E. Csuhaj-Varjú and G. Vaszil one uses final states for concluding a computation.

  Then, of course, no idea of halting appears in applications of P systems, especially in biology or economics, where the goal is not the result of a computation but the computation itself, the evolution in time of populations of objects.

- We arrive now at an important point, also many times invoked: *brane calculi pay more attention to the faithfulness to the biological reality, membrane computing is mainly interested in computational issues.* Right in both parts of the assertion, with the mentioning that in the case of membrane computing this is true only for those investigations . . . interested in computational issues. When trying to build computing models, powerful or efficient, we need mathematical models, as elegant as possible, hence as restricted as possible. When trying to build models of biological processes, we stay as close as possible to biology. This is the case with the many recent applications, from oscillations in eco-systems to EGFR

[3] There fit with this intuition those results that show that brane calculi devices working synchronously are Turing complete/undecidable, but this is not the case for asynchronous devices – sse [3], [4].

robustness, from circadian rhythms to quorum sensing in bacteria [4] .

- It is perhaps the time to also mention a series of obvious *common features* of the two domains: both of them are directly inspired from the cell structure and functioning, are based on discrete mathematics, can handle small populations of agents and slow reactions, are implicitly or explicitly algorithmic, lead to scalable models, easy to understand by biologists. All these are important arguments in promoting them as modeling tools for biology (with a general question formulated in the next section about how far in biology we can go with these tools).

Conclusions? Maybe a malicious one: membrane computing is so diverse at this moment that any assertion about it (in particular, about the differences between it and brane calculi) can be illustrated, and, at the same time, any such assertion can be refuted... "Technically", both A and $\neg A$ can be "proved", whatever assertion A is (well, almost). Maybe the only real difference is the fact that, basically, brane calculi uses process algebra as its technical framework, while membrane computing uses techniques from languages, automata, complexity, dynamical systems. These conclusions do not make the discussion useless, but make many phrases from many introductions to papers to look superficial and pointless.

3 Are We on the Right Way?

Well [5] , this is a too dramatic and too general question, maybe not directly related to the subject of this discussion, maybe prematurely formulated. It concerns both local issues, such as the relevance of computability results for biology, and more general issues, such as the place in and contribution of membrane computing and brane calculi to systems biology.

Because membrane computing was developed very much as a branch of theoretical computer science, universality and non-universality results are of a central *internal* interest. Still, I claim that this could be (must be?) of interest also for biologists, at least because in a universal (hence non-decidable) framework we cannot expect algorithmic answers to (any) nontrivial problems related to the evolution of the systems (Rice theorem). What remains to do is to try ad-hoc techniques for specific questions and, mainly, empiric solutions, especially simulations on computer. Being aware of this intrinsic difficulty (the cell is indeed "a powerful computer") it is already an important result (the most applicable mathematical results are the impossibility ones...).

Then, this discussion can be placed in a larger context, of the question what biology can/should get from computer science or ask to computer science, and this comes to the question what systems biology should bring to biology. Without being so drastic as the author of [28], who already from the title asks whether this is a genuinely new area of research or is just a "reincarnation of systems theory applied

[4] I would like to point out here the fact that we can already count an application of membrane computing to a biology issue of the *pre-diction* type, contrasting the many *post-diction* applications, namely the one of a team of biologists and computer scientists from Milano in the study of regulation pathways in yeast – see [18].

[5] Actually, it depends where we want to arrive...

in biology”, I agree with him in the main conclusions of the paper, as soundly expressed in the last paragraph of [28], where one of the important names of classic systems theory, M. Mesarović, is invoked: “Mihajlo Mesarović wrote in 1968 that ‘in spite of the considerable interest and efforts, the application of systems theory in biology has not quite lived up to expectations. (...) One of the main reasons for the existing lag is that systems theory has not been directly concerned with some of the problems of vital importance in biology’. His advice for biologists was that progress could be made by more direct and stronger interactions with systems scientists. ‘The real advance in the application of systems theory to biology will come about only when the biologists start *asking questions* which are based on the system–theoretic concepts rather than using these concepts to represent in still another way the phenomena which are already explained in terms of biophysical or biochemical principles. (...) Then we will not have the applications of engineering principles to biology problems but rather a field of *systems biology* with its own identity and in its own right’, [19].”

The slow progresses made in modeling and simulating alive systems suggest, in terms of [2], that, in spite of the many sound achievements, “we might be missing something fundamental and currently unimagined in our models of biology”. The computers are good in crunching numbers, but not “at modeling living systems, at small or large scales.” The clear intuition is that life is more than biochemistry, but what else should be considered can be something *unimagined*, something “invisible to us right now”. “It is not completely impossible that we might discover some new properties of biomolecules or some new ingredient.” An example of such “new stuff” can be the quantum effects in the microtubules of nerve cells, which, according to Penrose, “might be the locus of consciousness at the level of individual cells, which combines in bigger wave functions at the organism level.” ([2], page 410)

Without becoming a vitalist, I cannot stop asking myself *what else* should we add to our models to model *living* systems...

4 Closing Remarks

These notes are only personal thought and only working ideas, catalyzed by the MeCBIC workshop, a quick proof that membrane computing and brane calculi are not at all as different as stated in many paper introductions. In some sense, only “mem” is missing to make one single domain from the artificially separated two. Of course, pointing differences is a challenge to diminish differences, hence this is worth doing. And, there are so many things to do related to the cell modeling that the best thing to do is to try to do those things, irrespective how we call the resulting theory.

References

[1] D. Besozzi, D. Busi, G. Franco, R. Freund, Gh. Păun: Two universality results for (mem)brane systems. *Proc. Fourth Brainstorming Weeek on Membrane Computing*, Sevilla Univ., 2006, Fenix Editora, Sevilla, 2006, vol. I, 49–62.

[2] R. Brooks: The relationship between matter and life. *Nature*, 409 (Jan. 2001), 409–411.

[3] N. Busi: On the computational power of the mate/bud/drip brane calculus: interleaving vs, maximal parallelism. *Proc. WMC6*, Vienna, July 2005, 235–252, and *Membrane Computing, International Workshop, WMC6, Vienna, Austria, 2005, Selected and Invited Papers* (R. Freund, Gh. Păun, G. Rozenberg, A. Salomaa, eds.), LNCS 3850, Springer, Berlin, 2006, 144–158.

[4] N. Busi, R. Gorrieri: On the computational power of brane calculi. *Third Workshop on Computational Methods in Systems Biology*, Edinburgh, 2005.

[5] N. Busi, M.A. Gutiérrez-Naranjo: A case study in (mem)brane computation: generating $\{n^2 \mid n \geq 1\}$. *Proc. Fourth Brainstorming Weeek on Membrane Computing*, Sevilla Univ., 2006, Fenix Editora, Sevilla, 2006, vol. I, 81–98.

[6] N. Busi, C. Zandron: Computing with genetic gates, proteins and membranes. *Seventh Workshop on Membrane Computing, WMC7*, Leiden, July 2006.

[7] L. Cardelli: Brane calculi. Interactions of biological membranes. In *Computational Methods in Systems Biology. International Conference CMSB 2004, Paris, France, May 2004, Revised Selected Papers* (V. Danos, V. Schachter, eds.), LNCS 3082, Springer, Berlin, 2005, 257–280.

[8] L. Cardelli, Abstract machines of systems biology, *Proc. Third Brainstorming Week on Membrane Computing*, Sevilla, 2005, RGNC Report 01/2005, Fénix Editora, Sevilla, 2005, 73–86.

[9] L. Cardelli, Gh. Păun: An universality result for a (mem)brane calculus based on mate/drip operations. *Cellular Computing. Complexity Aspects* (M.A. Gutierrez-Naranjo, Gh. Păun, M.J. Perez-Jimenez, eds.), Fenix Editora, Sevilla, 2005, 75–94, and *Intern. J. Found. Computer Sci.*, 17, 1 (2006), 49–68.

[10] M. Cavaliere, A. Riscos-Nunez, R. Brijder, G. Rozenberg: Membrane systems with marked membranes. Submitted, 2005.

[11] E. Csuhaj-Varjú, A. Di Nola, Gh. Păun, M.J. Pérez-Jiménez, G. Vaszil: Editing configurations of P systems. *Proc. Third Brainstorming Week on Membrane Computing*, Sevilla, 2005, RGNC Report 01/2005, 131–155.

[12] V. Danos, S. Pradalier: Projective brane calculus. In *Computational Methods in Systems Biology: International Conference CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers*, (V. Danos, V. Schachter, eds.), LNCS 3082, Springer-Verlag, Berlin, 2005, 134–148.

[13] R. Freund: Asynchronous P systems and P systems working in the sequential mode. *Membrane Computing. 5th International Workshop WMC5, Milano, Italy, 2004, Revised Selected and Invited Papers* (G. Mauri, Gh. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa, eds.), LNCS 3365, Springer, Berlin, 2005, 36–62.

[14] R. Freund, M. Oswald: Tissue P systems with mate and drip operations. *MeCBIC*, Venice, 2006.

[15] M.A. Gutiérrez-Naranjo, M.J. Pérez-Jiménez, A. Riscos-Núñez, F.J. Romero-Campero: Characterizing tractability with membrane creation. *Proc. SYNASC 05*, IEEE Press, 2005, 448–457.

[16] S.N. Krishna: Universality results for a brane calculus. *Theoretical Computer Sci.*, to appear.

[17] S.N. Krishna: Combining brane calculus and membrane computing. *Conf. Bio-Inspired Computing – Theory and Applications*, BIC-TA, Wuhan, China, 2006.

[18] E. Martegani, R. Tisi, F. Belotti, S. Colombo, C. Paiardi, J. Winderickx, P. Cazzaniga, D. Besozzi, G. Mauri: Identification of an intracellular signalling complex for ras/camp pathway in yeast: experimental evidences and modelling. *25th International Specialized Symposium on Yeasts, "Systems Biology of Yeasts"*, Hanassari, Espo, Finland, 2006.

[19] M.D. Mesarović: System theory and biology – view of a theoretician. In *System Theory and Biology* (M.D. Mesarović, ed.), Springer, New York, 1968, 59–87.

[20] T.Y. Nishida: Simulations of photosynthesis by a K-subset transforming system with membranes, *Fundamenta Informaticae*, 49, 1-3 (2002), 249–259.

[21] A. Păun, B. Popa: P systems with proteins on membranes. *Fundamenta Informaticae*, 72, 4 (2006), 467–483.

[22] A. Păun, B. Popa: P systems with proteins on membranes and membrane division. *Proc. DLT 2006*, Santa Barbara, CA, 2006.

[23] Gh. Păun: Computing with membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for CS-TUCS Report* No. 208, 1998.

[24] Gh. Păun: *Membrane Computing – An Introduction*. Springer, Berlin, 2002.

[25] Gh. Păun: One more universality result for P systems with objects on membranes. *Proc. Third Brainstorming Week on Membrane Computing*, Sevilla, 2005, RGNC Report 01/2005, 263–274, and *Intern. J. of Computers, Communication and Control*, 1, 1 (2006), 44–51.

[26] Gh. Păun: 2006 research problems in membrane computing. *Proc. Fourth Brainstorming Weeek on Membrane Computing*, Sevilla Univ., 2006, Fenix Editora, Sevilla, 2006, vol. II, 235–252.

[27] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, E. Shapiro: BioAmbients – An abstraction for biological compartments. *Theoretical Computer Sci.*, 325, 1 (2004), 141–167.

[28] O. Wolkenhauer: Systems biology: The reincarnation of systems theory applied in biology? *Briefings in Bioinformatics*, 2, 3 (2001), 258–270.

[29] The P Systems Web Page: http://psystems.disco.unimib.it.