RESEARCHING FRAMEWORK FOR SIMULATING/IMPLEMENTATING P SYSTEMS

Sandra Gómez, Luis Fernández, Iván García, Fernando Arroyo

Abstract: Researching simulation/implementation of membranes systems is very recent. Present literature gathers new publications frequently about software/hardware, data structures and algorithms for implementing P system evolution. In this context, this work presents a framework which goal is to make tasks of researchers of this field easier. Hence, it establishes the set of cooperating classes that form a reusable and flexible design for the customizable evaluation with new data structures and algorithms. Moreover, it includes customizable services for correcting, monitoring and logging the evolution and edition, recovering, automatic generating, persistence and visualizing P systems.

Keywords: P System, framework, simulation, implementation.

ACM Classification Keywords: D.1.m Miscellaneous – Natural Computing

Introduction

P systems are a new computational model based on the membrane structure of living cells. This model has become, during last years, a powerful framework for developing new ideas in theoretical computation. ”P systems with simple ingredients (number of membranes, forms and sizes of rules, controls of using the rules) are Turing complete” [Păun, 1999]. Moreover, P systems are a class of distributed, massively parallel and non-deterministic systems.

”As there do not exist, up to now, implementations in laboratories (neither in vitro or in vivo nor in any electronical medium), it seems natural to look for software tools that can be used as assistants that are able to simulate computations of P systems” [Ciobanu, 2006]. ”An overview of membrane computing software can be found in literature , or tentative for hardware implementations , or even in local networks is enough to understand how difficult is to implement membrane systems on digital devices” [Păun, 2005]. Moreover, he says: ”we avoid to plainly say that we have ’implementations’ of P systems, because of the inherent non-determinism and the massive parallelism of the basic model, features which cannot be implemented, at least in principle, on the usual electronic computer -but which can be implemented on a dedicated, reconfigurable, hardware [...] or on a local network” [Păun, 2005]. Thereby, there exists many simulators in bibliography but ”the next generation of simulators may be oriented to solve (at least partially) the problems of storage of information and massive parallelism by using parallel language programming or by using multiprocessor computers” [Ciobanu, 2006].

The goal of this work is to present a framework to make easier the tasks of researchers who develop simulators/implementations of P systems. It does not expect to be a new simulator/implementation. It presents a set of cooperating classes that form a reusable design for developing simulators/implementations of P systems. This framework provides an architectonical guide to divide the design in abstract classes and to define their responsibilities and collaborations. Researchers have to adapt the framework to a concrete simulator/implementation inheriting and compounding instances of framework classes.

This paper is structured as follows: next section presents related works then, they are presented the requisites and design guidelines for the framework. Finally, conclusions are presented.

Related Works

Membrane system implementation is a very recent investigation field. First approaches were simulators [Ciobanu, 2006] that demonstrated the functionality of the membrane systems. But, they lacked distributed and massively character.

First distributed implementations are presented in [Syropoulos, 2003] and [Ciobanu, 2004]. In their distributed implementations of P systems use Java Remote Method Invocation (RMI) and the Message Passing Interface (MPI) respectively, on a cluster of PC connected by Ethernet. These last authors do not carry out a detailed analysis of the importance of the time used during communication phase in the total time of P system evolution, although Ciobanu affirms that "the response time of the program has been acceptable. There are however executions that could take a rather long time due to unexpected network congestion" [Ciobanu, 2003]. In [Tejedor, 2007a] [Bravo, 2007a] [Bravo, 2007b], It is determined that the problem in implementing P systems is the time necessary in the communication of multisets among membranes allocated in different devices (PCs, PICs, or chips). This fact, forces to resign parallelism to the maximum to as much reach a parallelism degree dependent of the speed of the communications and the application of the evolution rules. Therefore, it is necessary to develop faster application algorithms that adapt so much to the sequential technologies as to the parallel ones.

On the other hand, [Fernández, 2007] determines the appropriate software architecture that is executed over a given evolution P system hardware architecture. So, it pretends to determine the set of process and their relationships that are appropriate to be executed over a set of connected processors. Considered possibilities are: evolution rules oriented software architecture, membranes oriented software architecture and processors oriented software architecture.

Works of investigation about sequential and/or parallel algorithms designed for the different phases of the evolution of a P system are very varied: for the utility of the evolution rules: [Frutos, 2007]; for the applicability of evolution rules: [Fernández, 2006a]; for the application of evolution rules: [Fernández, 2006b] [Fernández, 2006c] [Tejedor, 2007b] [Tejedor, 2007c] [Gil, 2007].

With respect to the storage of the information of a P System, [Fernández, 2005a] defines a universal vocabulary with XML technology and [Gutiérrez, 2007b] presents new data structures that compress multisets of objects information without penalizing the basic operations on these.

Finally, it is possible to indicate the works on different technologies whose objective is to implant the different architectures, algorithms and previous data structures. Thus, we found a line about circuits hardware in [Petreska, 2003] [Arroyo, 2004a] [Arroyo, 2004b] [Arroyo, 2004c] [Fernández, 2005b] [Martínez, 2006a] [Martínez, 2006b], the new opened line about microcontrollers in [Gutierrez, 2006] [Gutierrez, 2007a], and the traditional line about personal computers in [Syropoulos, 2003] [Ciobanu, 2004].

Requisites

In this context, pursued goals is to develop a highly reusable framework that is flexible enough for any researcher to be able of concentrating on developing the algorithm or data structure object of its investigation.

In this line, the framework provides to the researchers the following reusable modules:

1.1 Implementation of every standard data structure, agreed to the specification model, in order to equip framework with the total functionality of a simulator of membrane system. Hence, it is had the data structures for the symbols, multisets, evolution rules and membranes.

1.2 Implementation of every standard algorithm, agreed to the specification model, in order to equip framework with the total functionality of a simulator of membrane system. Hence, it is hadthe algorithms for utility phases, applicability, activity, application, communication and dissolution.

1.3 Process management and synchronization for the different software architectures: evolution rules oriented, membranes oriented and processor oriented.

1.4 Management of automatic detection of errors of any algorithm for functional tests.

1.5 Management of automatic monitoring (time, space, number of operations, …) of any algorithm for non functional tests.

1.6 Management of persistence, visualizing and logs for tracking the algorithms.

1.7 Management of P System for its edition, recovery, automatic generation of parameterized sets of tests

1.8 Management of configurations for P systems evolution.

1.9 Gestión de configuraciones para la evolución de P Systems.

On the other hand, the framework provides to the researchers the following flexibility for a given evolution of the P system:

2.1. Extension by inheritance of new data structures for the symbols, multisets, evolution rules and membranes.

2.2. Extension by inheritance of new algorithms for utility, applicability, activity, application, communication and dissolution phases.

2.3. Extension by inheritance of new functionalities over P systems (analyisis, compilation, …).

2.4. Architecture process, phases, algorithms and data structures configuration.

2.5. Evolution, visualization, monitorization, correction and logs configuration.

Figure 1 shows the use case diagram corresponding to the previous requisites.

Figure 1: Use Case Diagram.

Framework

Figure 2 shows a class diagram of the domain model that has the most important object classes according to P system specification.

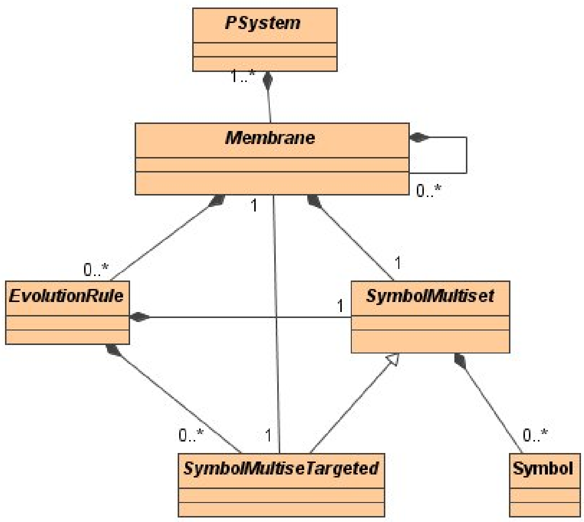

Figure 2: P system Domain Model Class Diagram.

Figure 3 shows the class diagram that was designed for covering requisites 1.1 and 2.1. This way, concrete classes in the third level of the inheritance hierarchy are contributed for every standard data structure. Also, it makes easy the incorporation of new data structures inheriting from the abstract classes of the second inheritance hierarchy level.

In particular, class *ElementFactory* is responsible of the configuration of the data structures for a given evolution of requisite 2.4.

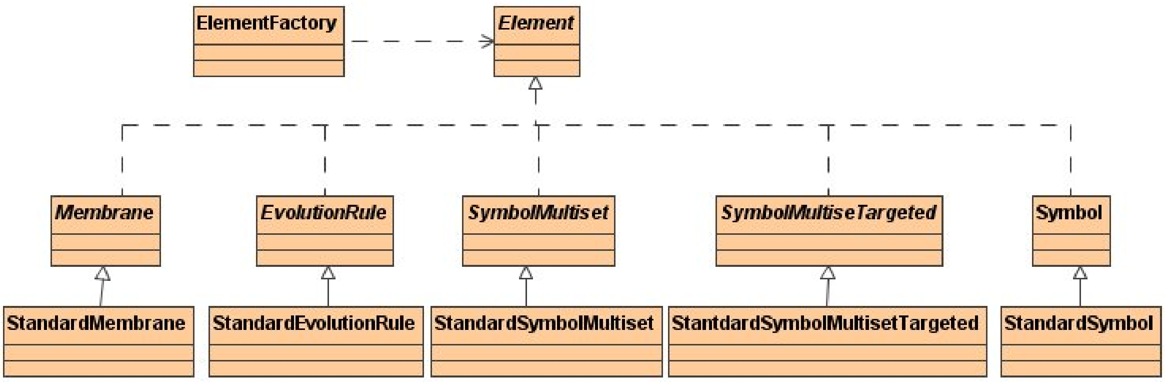

Figure 3: Data Structures Class Diagram.

Figure 4 shows the class diagram designed to cover requisites 1.2 and 2.2. This way, concrete classes in the fifth level of the inheritance hierarchy are contributed for every algorithm of evolution phases. Also, it makes easy the incorporation of new algorithms inheriting from forth level of inheritance hierarchy abstract classes.

In particular, class PhaseFactory is responsible of the configuration of the phases and algorithms for a concrete evolution of requisite 2.4.

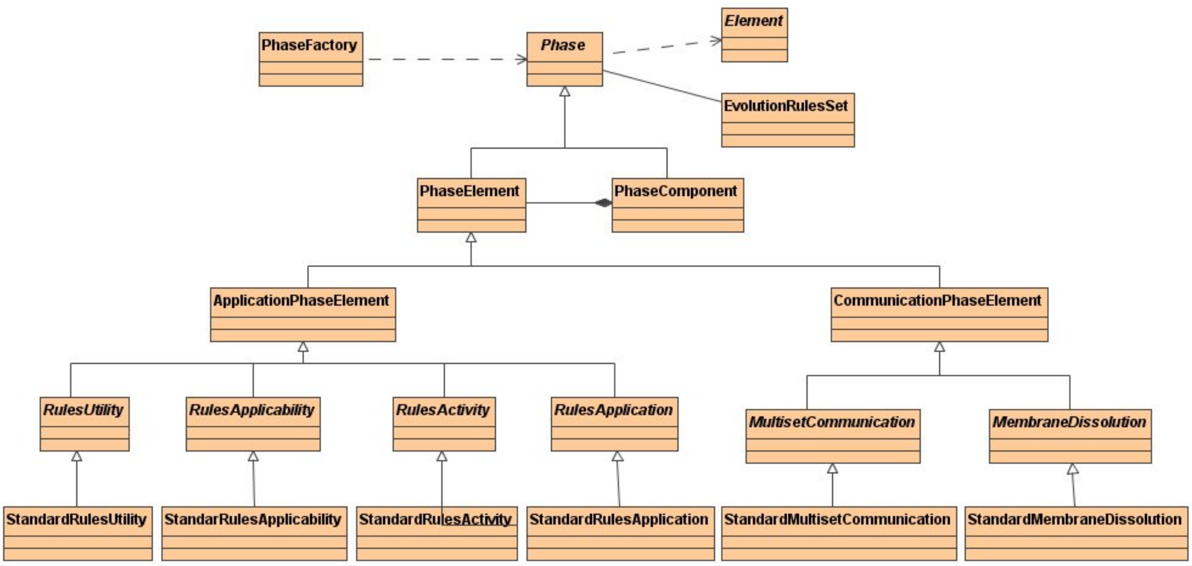

Figure 4: Algorithms Class Diagram.

Figure 5 shows the class diagram designed to cover the requisite 1.3. This way, concrete classes in the second level of the inheritance hierarchy are contributed for every process architectures together with the classes for the process synchronization.

In particular, class *ProcessFactory* is responsible of the configuration of the process architectures for a concrete evolution of requisite 2.4.

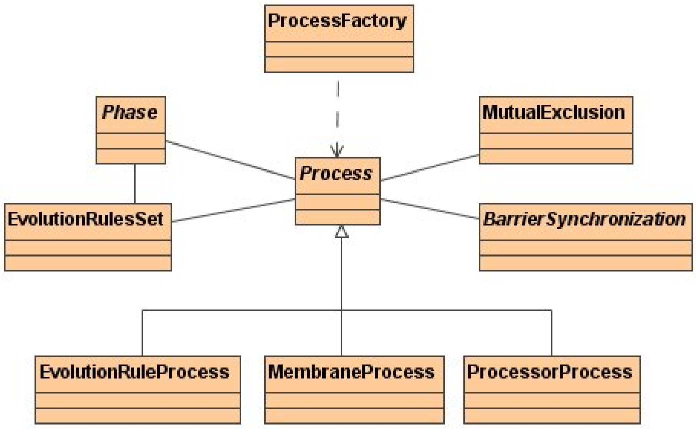

Figure 5: Software Architecture Class Diagram.

Figure 6 shows the class diagram designed to cover requisites 1.4, 1.5, 1.6 and 2.3. This way, concrete classes in the forth level of inheritance hierarchy are contributed for the detection of errors and automatic monitorization of functional and non functional sets of tests respectively, and for the persistence, log and visualization of the results of a given evolution. Moreover, new functionalities (P system analysis, compilation, …) can be developed inheriting from *VisitorElement*.

In particular, class *VisitorFactory* is responsible of the configuration of a given evolution of the requisite 2.5.

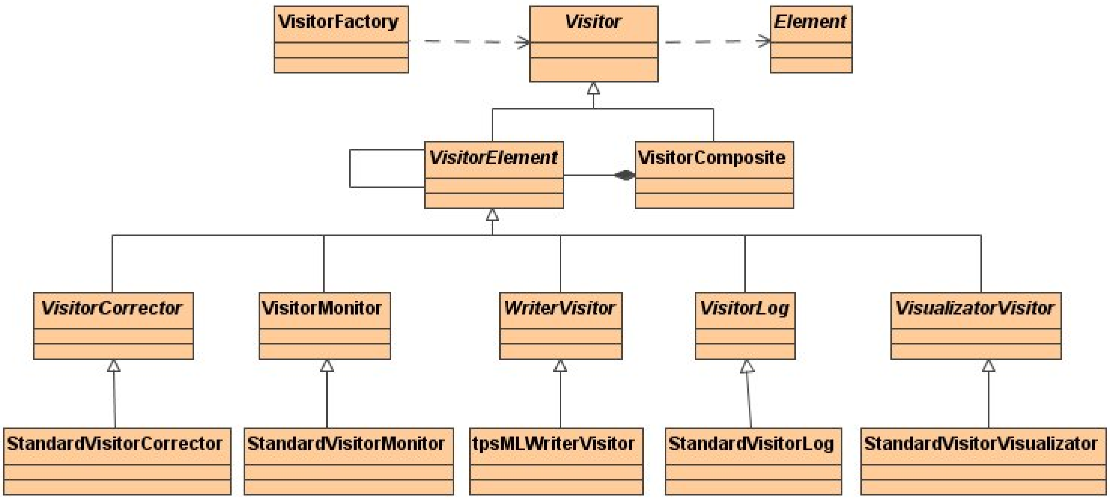

Figure 6: Visitors Class Diagram

Figure 7 shows the general class diagram that relates every class of previous class diagrams. In particular, class *PSystemFactory* is responsible of the edition, recovery and sets of parameterized tests automatic generation of requisite 1.7. Moreover, set of factory classes is responsible of managing the configurations for a given evolution of requisite 1.8.

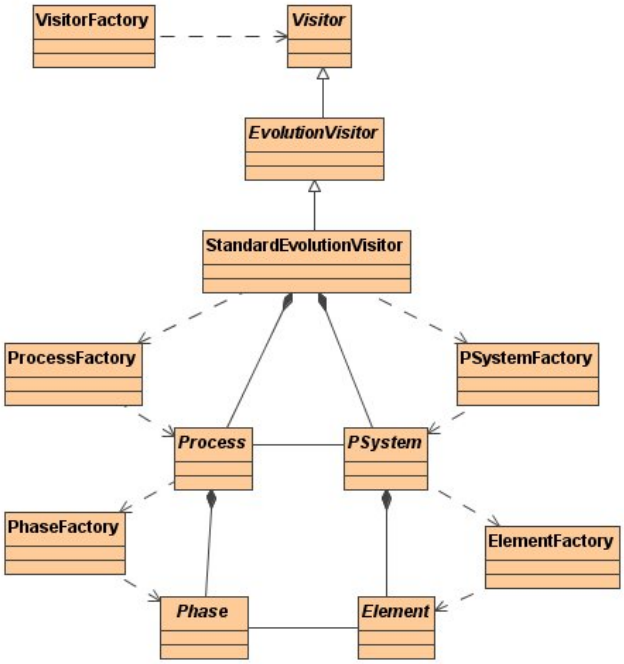

Figure 7: General Class Diagram.

Conclusion

This work contributes a framework that makes investigation of developing new simulators and implementations of membrane systems easier. Its goal is to provide enough reusability and flexibility to get the researcher is concentrated in the goals of his investigation. This way, it is possible to reuse standard data structures and algorithms of the P system model, processes and synchronization management, error detection, monitorization and log for tests phase and the edition, recovery, automatic generation, persistence and visualization of P systems. On the other hand, the simple inheritance mechanism provides flexibility for the incorporation of new data structures and algorithms.

Bibliography

[Alonso, 2007] S. Alonso, L. Fernández, F. Arroyo, J. Gil. *A Circuit Implementing Massive Parallelism in Transition P Systems*. Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Arroyo, 2004a] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. *A binary data structure for membrane processors: Connectivity Arrays*. Lecture Notes in Computer Science (A. Alhazov, C. Martin-Vide, G. Mauri, G. Paun, G.Rozenberg, A. Saloma, eds.) Springer Verlag (2933), 2004. 19-30.

[Arroyo, 2004b] F. Arroyo, C. Luengo, Castellanos, L.F. de Mingo. *Representing Multisets and Evolution Rules in Membrane Processors.* Preproceedings of the Fifth Workshop on Membrana Computing (WMC5). Milano (Italy) june, 2004. 126-137.

[Arroyo, 2004c] F. Arroyo, C. Luengo, L. Fernandez, L.F. de Mingo, J. Castellanos, *Simulating membrane systems in digital computers.* ITHEA-2004 International Journal Information Theories and Applications (vol.11 - 1) 2004. 29-34.

[Bravo, 2007a] G. Bravo, L. Fernández, F. Arroyo, J.A. Frutos. *A Hierarchical Architecture with parallel comunication for implementing P Systems.* Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Bravo, 2007b] G. Bravo, L. Fernández, F. Arroyo, J. Tejedor. *Master/Slave Parallel Architecture for Implementing P Systems.* The 8th WSEAS International Conference on Mathematics and Computers in Business and Economics (MCBE'07). Vancouver (Canada) june, 2007. (submitted).

[Ciobanu, 2004] G.Ciobanu, W.Guo . *P Systems Running on a Cluster of Computers*. Workshop on Membrane Computing (Gh. Păun, G. Rozenberg, A. Salomaa Eds.), LNCS 2933, Springer, 123-139.

[Ciobanu, 2006] G. Ciobanu, M. Pérez-Jiménez, Gh. Păun. *Applications of Membrana Computing*". Natural Computing Series, Springer Verlag, october, 2006.

[Fernández, 2005a] L. Fernández, F. Arroyo, J. Castellanos, V.J. Martinez, L.F. Mingo. *Software Tools/ P System Simulators Interoperability.* (R. Freund, G. Lojka, M. Oswald, Gh. Paun, eds.) Preproceedings of Sixth International Workshop on Membrane Computing (WMC6), Vienna, (Austria) june, 2005. 147-161.

[Fernández, 2005b] L. Fernandez, V.J. Martinez, F. Arroyo, L.F. Mingo. *A Hardware Circuit for Selecting Active Rules in Transition P Systems.* (G. Ciobanu, Gh. Paun, eds.) Preproceedings of First International Workshop on Theory and Application of P Systems, Timisoara (Romania), september, 2005. 45-48.

[Fernández, 2006a] L. Fernández, F. Arroyo, I. García, G. Bravo*. Decision Trees for Applicability of Evolution Rules in Transition P System*. ITHEA-2006 Interantional Journal Information Theories and Applications (vol.11 - 1) 2006. 29-34.

[Fernández, 2006b] L. Fernández, F. Arroyo, J.A. Tejedor, J. Castellanos*, Massively Parallel Algorithm for Evolution Rules Application in Transition P Systems*. Preproceedings of Membrane Computing, International Workshop (WMC7), Leiden (The Netherlands) july, 2006. 337-343.

[Fernández, 2006c] L. Fernández, F. Arroyo, J. Castellanos, J.A. Tejedor, I. García, *New Algorithms for Application of Evolution Rules based on Applicability Benchmarks*. International Conference on Bioinformatics and Computational Biology(BIOCOMP06), Las Vegas (EEUU), july, 2006.

[Fernández, 2007] L. Fernández, F. Arroyo, I. Garcia, A. Gutierrez. *Parallel software architectures analysis for implementing P systems.* (M. Sugisaka, H. Tanaka, eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics (AROB07), Beppu (Japan) january, 2007. 494-499.

[Frutos, 2007] J.A.Frutos, L. Fernández, F.Arroyo, G.Bravo. *Static Analysis of Usefulness States in Transition P Systems.* Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Gil, 2007] F.J. Gil, L. Fernández, F. Arroyo, J.A. Tejedor. *Delimited Massively Parallel Algorithm based on Rules Elimination for Application of Active Rules in Transition P Systems.* Fifth International Conference Information Research and Applications (i.TECH-2007). Varna (Bulgary) june, 2007. (submitted).

[Gutiérrez, 2006] A. Gutiérrez, L. Fernández, F. Arroyo, V. Martínez. *Design of a Hardware Architecture based on Microcontrollers for the Implementation of Membrane Systems*. Proceedings on 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC-2006). Timisoara (Romania) septiembre, 2006. 39-42.

[Gutiérrez, 2007a] A. Gutiérrez, L. Fernández, F. Arroyo, S. Alonso. *Hardware and Software Architecture for Implementing Membrane Systems: A case of study to Transition P Systems.* The International Meeting on DNA Computing (DNA13), Memphis (USA) June 3-8, 2007. (submitted)..

[Gutiérrez, 2007b] A. Gutiérrez, L. Fernández, F. Arroyo, G. Bravo. *Compression of Multisets and Evolution Rules Optimizing the Storage and Communication in Membrana System.* Eight Workshop on Membrane Computing (WMC8). Thessaloniki (Greece) june, 2007 (submitted).

[Martínez, 2006a] V. Martinez, L. Fernández, F. Arroyo, I. García, A. Gutierrez. *A HW circuit for the application of Active Rules in a Transition P System Region*. Proceedings on Fourth International Conference Information Research and Applications (i.TECH-2006). Varna (Bulgary) June, 2006. pp. 147-154. ISBN-10: 954-16-0036-0.

[Martínez, 2006b] V. Martínez, L. Fernández, F. Arroyo, A. Gutierrez. *HW Implementation of a Bounded Algorithm for Application of Rules in a Transition P-System*. Proceedings on 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC-2006). Timisoara (Romania) septiembre, 2006. pp. 32-38.

[Păun, 1999] Gh.Paun. *Computing with membranes*. An introduction. Bulletin of the EATCS, 67, 139-152, 1999.

[Păun, 2005] PGh. Păun. *Membrane computing. Basic ideas, results, applications.* Pre-Proceedings of First International Workshop on Theory and Application of P Systems. Timisoara (Romania), september , 2005. 1-8.

[Petreska, 2003] B. Petreska and C. Teuscher. *A hardware membrane system*. A. Alhazov, C. Martin-Vide, Gh. Paun (eds.):Pre-proceedings of the workshop on Membrane Computing Tarragona, July 17-22 2003, 343-355.

[Syropoulos, 2003] A. Syropoulos, E.G. Mamatas, P.C. Allilomes. *A distributed simulation of P systems.*(A. Alhazov, C. Martin-Vide and Gh. Păun, eds.) Preproceedings of the Workshop on Membrana Computing. Tarragona (Spain), july, 2003, 455-460.

[Tejedor, 2007a] A. Tejedor, L. Fernández, F. Arroyo, G. Bravo, *An architecture for attacking the bottleneck communication in P systems*. In: M. Sugisaka, H. Tanaka (eds.), Proceedings of the 12th Int. Symposium on Artificial Life and Robotics, Jan 25-27, 2007, Beppu, Oita, Japan, 500-505.

[Tejedor, 2007b] A. Tejedor, L. Fernández, F. Arroyo, A. Gutierrez. *Algorithm of Active Rule Elimination for Application of Evolution Rules.* The 8th WSEAS International Conference on Mathematics and Computers in Business and Economics (MCBE'07). Vancouver (Canada) june, 2007. (submitted).

[Tejedor, 2007c] A. Tejedor, L. Fernández, F. Arroyo, S. Gómez. *Application Algorithm based on Evolution Rules Competitivity.* Eight Workshop on Membrane Computing (WMC8). Thessaloniki (Greece) june, 2007 (submitted).

Authors' Information

Sandra Gómez Cannaval – Natural Computing Group. Dpto. Organización y Estructura de la Información de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: sgomez@eui.upm.es

Luis Fernández Muñoz – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: setillo@eui.upm.es

Iván García – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: igarcia@eui.upm.es

Fernando Arroyo – Natural Computing Group. Dpto. Lenguajes, Proyectos y Sistemas Informáticos de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, Ctra. de Valencia, km. 7, 28031 Madrid (Spain); e-mail: farroyo@eui.upm.es