A uniform framework for modeling based on P systems

M. Angels Colomer [*1], M.A. Martínez-del-Amor [#2], Ignacio Pérez-Hurtado [#3], Mario J. Pérez-Jiménez [#4], A. Riscos-Núñ

[*] *Department of Mathematics, University of Lleida*
Avda. Alcalde Rovira Roure, 191, 25198 Lleida, Spain
[1] colomer@matematica.udl.es

[#] Dpt. of Computer Science and Artificial Intelligence, University of Seville
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
[2] mdelamor@us.es, [3] perezh@us.es, [4] marper@us.es, [5] ariscosn@us.es

***Abstract*—In this paper, a P systems based general framework for modeling the dynamics of a population biology is presented. Multienvironment probabilistic functional P systems with active membranes provide the syntactical specification, and the semantics is captured by using stochastic or probabilistic strategies implemented through simulation algorithms.**

I. Introduction

Modeling is real world problem solving and so the development of a model is a hard process prone to failure where one has to reconsider many times the assumptions, simplifications, etc. made at different points. The modeling process is a semiformal set of rules that guides us to produce a model, formulate it in a formal language, implement it on a computer and derive properties of the system under research.

The use of models is necessary because the subject of investigation is often too complex to work wit, and it is intrinsic to any scientific activity. Scientists regularly use abstractions of the some part of the universe with the aim to describe and understand the part they are analyzing.

Generally, important steps in the modeling process are the selection of relevant features and laws governing the behavior of the phenomenon under investigaction. Our mental picture and understanding of the phenomenon is first suggested by experimental data, then we improve our hypotheses, theories and models of the phenomenon by carrying out experiments that produce data.

Although biologists are familiar with modeling, quantitative computational mathematical models have lain outside the mainstream due to the lack of techniques from both experimental and theoretical/computational sides. Nonetheless, at the end of the last century extraordinary advances were achieved in both computer science and biology. The progress in molecular cell biology and population biology along with the development of computer science has reached the point where each one can benefit from the other one, integrating computer science and biology.

A model is an abstraction of a part of a real-world onto a mathematical/computational domain that highlights some key properties while ignoring other that are assumed to be irrelevant. A good model must show how a phenomenon work and then predict what may follows. In a formal model it is desirable to have at least four properties: relevance, understandability, extensibility and computability [22].

One of the main objectives of any model is in capability to predict, that is, the possibility to make conjecture of plausible hypothesis related to the dynamics of the observed phenomenon in different scenarios. It is important to notice that different models might be able to create similar spatio-temporal behavior but they are distinguished by the different experiments they suggest.

Usually cellular systems and population biology often depend on many variables of the observed behaviors. Since they define the dynamics of the system, variables must satisfy some conditions, which can be refered as the invariants of the behavior. Some of these expressions can be expressed by rules and can be obtained realizing experiments, nonetheless, others can not be measured in the lab or are very expensive to estimate. Therefore before simulations can be performed in order to make predictions, we need to *calibrate* our model by obtaining estimates for missing variables and validate it against the expected behaviour of the system. This requires comparing trial results or simulations obtained using the model with real data coming from the lab or with data generated using *trustable* methods. This stage consist of an iterative process in which a candidate set of parameters is proposed, some simulations are generated and on the basis of some metric of closeness to the desired behaviour a new set of parameters is tested. In some cases, the design od the model must be reconsidered.

In this context it is necessary to develop software applications in order to capture the semantics of the model and to be able to reproduce the behavior of the phenomenon observed in different initial conditions suggested by experts, by using electronic computers

The paper is structured as follows. Next, we present some relevant features related to the computational paradigm of Membrane Computing. In Section 3 we introduce a P system based modeling approach to population biology fulfilling the requirements of a *good* modeling framework. Section 4 is devoted to present a simulation algorithm that captures the

semantics of probabilistic P systems, and a case study is briefly described in order to illustrate the modeling principles presented. Section 5 concludes the paper with some possible directions for further research.

II. P SYSTEMS BASED MODELS

Membrane Computing is a young branch of Natural Computing providing distributed parallel computing models whose computational devices are called P systems. This discipline was initiated by Gh. Păn at the end of 1998 [17] and it is inspired by some basic biological features, by the structure and functioning of living cells, as well as from the interactions in tissues or higher order biological structures. Specifically, it starts from the observation that membranes play a fundamental role in the functioning of living cells because are involved in many reactions taking place inside the various compartments of a cell, and they act as selective channels of communication between different compartments as well as between the cell and the environment. P systems process multisets of objects in the compartments of a cell–like hierarchical arrangement of membranes (a rooted tree), or in a tisse–like structure consisting of cell placed in the nodes of a directed graph.

Phenomena under investigation are described by means of multienvironment P systems consisting of a finite number of environments, each of them having a specific P system with active membranes. Each rule has associated a computable function which depends on the left-hand side of the rule and the run time. P systems provide a high level computational modeling framework which integrates the structural and dynamical aspects of ecosystems in a compressive and relevant way. The inherent randomness and uncertainty in ecosystems is captured by using probabilistic strategies. Rather than being an alternative to more classical modeling frameworks, like ODEs, P systems constitute a complementary approach to be used when the classical modeling approaches fail to specify and simulate certain populations biology correctly. The inherent stochasticity, external noise and uncertainty in population biology is captured by using stochastic and probabilistic strategies.

We must stress that P systems, according to the original motivation, were not intended to provide a comprehensive and accurate model of the living cell, rather, to explore the computational nature of various feature of biological membranes. Indeed, most variants of membrane systems have been proved to be *computationally complete*, that is, their are equivalent in power to Turing machines, and *computationally efficient*, that is, able to solve computationally hard problems in polynomial time by trading time with space (by making use of an exponential workspace constructed in a natural way). Although most research in P systems concentrates on computational powers, recently P systems have been used to model biological phenomena within the framework of cellular systems and population biology presenting models of oscillatory systems [5], signal transduction [18], [4], gene regulation control [23], quorum sensing [24], metapopulations [19], and real ecosystems [1], [2].

A. Syntactics Specification

First, we define a P systems based framework (probabilistic P systems) where additional features, such as 3 electrical charges which describe specific properties in a better way, are used.

Definition 1: A skeleton of an extended P system with active membranes of degree $q \geq 1$ is a tuple $\Pi = (\Gamma, \mu, R)$ where:

- Γ is an alphabet (the working alphabet);
- μ is a membrane structure (that is, a rooted tree) consisting of q membranes, with the membranes injectively labeled with $0, 1, \ldots, q-1$. The skin membrane is labeled with 0. We also associate electrical charges with membranes from the set $\{0, +, -\}$;
- R is a finite set of evolution rules of the form $r : u[\ v\ ]_i^{\alpha} \rightarrow u'[\ v'\ ]_i^{\alpha'}$ where $u, v, u', v' \in M(\Gamma)$, $i \in \{0, 1, \ldots, q-1\}$, and $\alpha, \alpha' \in \{0, +, -\}$;

A skeleton of an extended P system with active membranes of degree $q \geq 1$, $\Pi = (\Gamma, \mu, R)$, can be viewed as a set of (polarizated) membranes hierarchized by structure μ labeled by $0, 1, \ldots, q-1$. All membranes in μ are supposed to be (initially) neutral and they have associated a finite set R of evolution rules that can modify their polarization.

Definition 2: A functional P system with active membranes of degree $q \geq 1$ taking T time units, $T \geq 1$, is a tuple

$$\Pi = (\Gamma, \mu, R, T, \{f_r : \ r \in R\}, \mathcal{M}_0, \ldots, \mathcal{M}_{q-1})$$

where:

- (Γ, μ, R) is the skeleton of an extended P system with active membranes of degree q
- T is a natural number;
- For each $r \in R$, f_r is a computable function such that $dom(f_r) \subseteq \{1, \ldots, T\}$
- $\mathcal{M}_0, \ldots, \mathcal{M}_{q-1}$ are strings over Γ, describing the multisets of objects initially placed in the q regions of μ.

A functional P system with active membranes of degree $q \geq 1$ taking T time units, $\Pi = (\Gamma, \mu, R, T, \{f_r : \ r \in R\}, \mathcal{M}_0, \ldots, \mathcal{M}_{q-1})$, can be viewed as a set of membranes hierarchized by the structure μ labeled by $0, 1, \ldots, q-1$. T is a natural number that represents the simulation time of the system. For each $r \in R$ and t, $1 \leq t \leq T$, $f_r(t)$ represents a constant associated with the rule r at moment t. In a generic way, we denote $r : u[v]_i^{\alpha} \xrightarrow{f_r(t)} u'[v']_i^{\alpha'}$. If $f_r(t) = 1$, then we briefly denote $r : u[v]_i^{\alpha} \longrightarrow u'[v']_i^{\alpha'}$.

The q–tuple of multisets of objects present at any moment in the q *regions* of the system, and their polarizations, constitutes the *configuration* of the system at that moment. The q–tuple $(\mathcal{M}_0, \ldots, \mathcal{M}_{q-1})$ with neutral polarizations in all membranes, is the initial configuration of the system Π.

The P system can pass from one configuration to another by using the rules from R as follows:

- A rule $u[v]_i^\alpha \rightarrow u'[v']_i^{\alpha'}$ is applicable to a membrane labeled by i, and with α as electrical charge if multiset u is contained in the father of membrane i, and multiset v is contained in the membrane labeled by i having α as electrical charge. When that rule is applied, multiset u (resp. v) in the father of membrane i (resp. in membrane i) is removed from that membrane, and the multiset u' (resp. v') is produced in that membrane, changing its label to α'.
- The family $\{f_r : r \in R\}$ of computable functions associates with the rules allow us to define the dynamics of the system, according to two basic orientations: stochastic and probabilistic.

Definition 3: A multienvironment functional P system with active membranes of degree (q, m, n) with $q \geq 1,\ m \geq 1,\ n \geq 1$, taking T time units, $T \geq 1$, is a tuple

$$(G, \Gamma, \Sigma, R_\Pi, R_E, \{f_{r,k} : r \in R_\Pi, 1 \leq k \leq n\},$$

$$\{M_{ik} : 0 \leq i \leq q-1,\ 1 \leq k \leq n\})$$

where:

- $G = (V, S)$ is a directed graph such that $(x, x) \in S$, for each $x \in V$. The elements of the set $V = \{e_1, \ldots e_m\}$ are called environments;
- Γ is the working alphabet and $\Sigma \subsetneqq \Gamma$ is an alphabet representing the objects that can be present in the environments;
- $\Pi = (\Gamma, \mu, R_\Pi)$ is the skeleton of an extended P system with active membranes of degree q;
- For each $r \in R_\Pi$ and for each j, $1 \leq j \leq m$, $f_{r,j}$ is a computable function whose domain is $\{1, 2, \ldots, T\}$ and its range is contained in $[0, 1]$;
- For each k, $1 \leq k \leq n$,

  $$\Pi_k = (\Gamma, \mu, R_\Pi, T, \{f_{r,k} : r \in R_\Pi\}, \mathcal{M}_{0k}, \ldots, \mathcal{M}_{q-1,k})$$

  is a functional extended P system with active membranes of degree $q \geq 1$ taking T time units.
- R_E is a finite set of communication rules of the form

  $$(x)_{e_j} \xrightarrow{p_{(x,j,j')}} (y)_{e_{j'}} \qquad \text{and} \qquad (\Pi_k)_{e_j} \xrightarrow{p_{(k,j,j')}} (\Pi_k)_{e_{j'}}$$

  where $x, y \in \Sigma$, $(e_j, e_{j'}) \in S$, $1 \leq k \leq n$, $p_{(x,j,j')}$ and $p_{(k,j,j')}$ are computable real functions whose domain is $\{1, \ldots, T\}$.

A multienvironment functional extended P system with active membranes of degree (q, m, n) taking T time units

$$(\Gamma, \Sigma, G, R_E, \Pi, \{f_{r,j} : r \in R_\Pi, 1 \leq j \leq m\},$$

$$\{M_{ij} : 0 \leq i \leq q-1,\ 1 \leq j \leq m\})$$

can be viewed as a set of m environments $e_1, \ldots e_m$ linked between them by the arcs from the directed graph G in which there are n P systems eventually different but with the same skeleton (and initially distributed according to the orientation of the system).

When a communication rule between environments

$$(x)_{e_j} \xrightarrow{p_{(x,j,j')}} (y)_{e_{j'}}$$

is applied, object x pass from e_j to $e_{j'}$ possibly modified into an object y. When a communication rule between environments

$$(\Pi_k)_{e_j} \xrightarrow{p_{(k,j,j')}} (\Pi_k)_{e_{j'}}$$

is applied, the P system Π_k pass from e_j to $e_{j'}$.

We assume that a global clock exists, marking the time for the whole system (for its compartments), that is, all membranes and the application of all rules are synchronized.

The tuple of multisets of objects present at any moment in the m environments and at each of the regions of the P systems located at the said environment, and the polarizations of every membrane in each P system, constitutes a *configuration* of the system at that moment. At the initial configuration of the system we assume that each environment is empty and each membrane has a neutral polarization.

The P system can pass from one configuration to another by using the rules from $R = R_E \cup \bigcup_{j=1}^m R_{\Pi_j}$ as follows: at each transition step, the rules to be applied are selected according to the probabilities assigned to them, and all applicable rules are simultaneously applied and all occurrences of the left–hand side of the rules are consumed, as usual.

III. Stochastic P systems based models

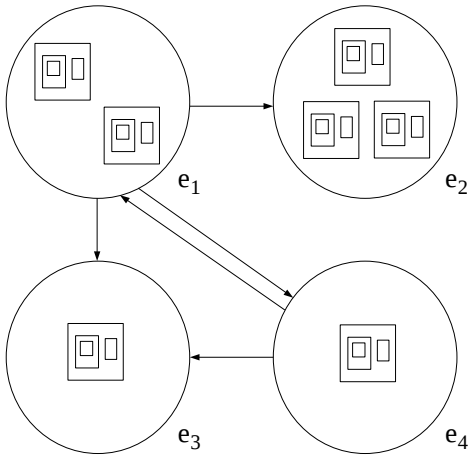
A multienvironment functional P system with active membranes of degree (q, m, n) with $q \geq 1,\ m \geq 1,\ n \geq 1$, taking T time units, $T \geq 1$, is said to be stochastic if:

- The functions $p_{(x,j,j')}$ and $p_{(k,j,j')}$ are constant functions and all the P systems Π_k are equal. Usually, they are *mesoscopic constants*, propensities, associated with the rules (computed from the stochastic constant which can be computed using the reaction rate constant used in conventional deterministic chemical kinetics like ODEs).
- At the initial moment, the n systems Π_k are ramdomness distributed between the m environments; for instance, as appear in the picture.
  When a communication rule between environments $(x)_{e_j} \xrightarrow{c} (y)_{e_{j'}}$ is applied, object x pass from e_j to $e_{j'}$ possibly modified into an object y. When a communication rule between environments $(\Pi_k)_{e_j} \xrightarrow{c'} (\Pi_k)_{e_{j'}}$ is applied, the P system Π_k pass from e_j to $e_{j'}$. The role played by the constants c and c' is given by the simulation algorithm that captures the semantics of the model.

A. A simulation algorithm for stochastic P systems

In what follows we present an adaptation of the Gillespie algorithm that can be applied in the different regions defined by the hierarchical and compartmentalized structure of a P system model. This will be referred to as *Multi-compartmental Gillespie algorithm*.

The starting point consists of dealing each region, delimited by a membrane, as a well mixed and fixed volume where



the classical Gillespie algorithm is applied. Thus in each compartment/membrane the next rule to be applied and the waiting time for this application is computed using a *local* Gillespie algorithm that only takes into account the number of molecules, rules and volume of the compartment. Given the state of a compartment i, $M_i = (l_i, w_i, s_i)$, with $l_i \in L$, the label of this membrane, $w_i \in O^*$ a finite multiset of objects and s_i a finite set of strings over O, the next rule to be applied and its waiting time is computed as follows:

- For each rule in $r_j \in R_{l_i}$, compute its propensity, p_j, from the stochastic constant c_j.
- Compute the sum of all propensities: $a_0 = \sum p_j$, for all $r_j \in R_{l_i}$;
- Generate two random numbers r_1 and r_2 uniformly distributed over the unit interval $(0,1)$;
- calculate the waiting time for the next reaction as $\tau_i = \frac{1}{a_0} ln(\frac{1}{r_1})$;
- take the index j such that $\sum_{k=1}^{j-1} p_k < r_2 a_0 \leq \sum_{k=1}^{j} p_k$;
- return the triple (τ_i, j, i).

By the *Multi-compartmental Gillespie Algorithm*, the compartments defined by membranes are ordered in a priority queue according to when the rules are scheduled to be applied. The first rule to be applied in the whole system occurs in the compartment on top of the priority queue. Depending on the type of rule that has been applied the state of a single compartment or of two compartments is changed. Therefore the waiting time and rule to be applied in these compartments must be recalculated. The algorithm halts when a prefixed simulation time is reached.

That is, the pseudocode of the multi-compartmental Gillespie algorithm is the following:

- **Initialisation**
  - set time of the simulation $t = 0$;
  - for each membrane i compute a triple (τ_i, j_i, i) by using the procedure described above; construct a list containing all such triples;
  - sort this list of triples (τ_i, j_i, i) in increasing order according to τ_i;
- **Iteration**
  - extract the first triple, $(\tau_{i_0}, j_{i_0}, i_0)$ from the list;
  - set time of the simulation $t = t + \tau_{i_0}$;
  - update the waiting time for the rest of the triples in the list by subtracting τ_{i_0};
  - apply the rule $r_{j_{i_0}}$ in membrane i only once changing the number of objects and sites in the membranes affected by the application of the rule;
  - for each membrane i' affected by the application of the rule remove the corresponding triple $(\tau_{i'}, j_{i'}, i')$ from the list;
  - for each membrane i' affected by the application of the rule $r_{j_{i_0}}$ re-run the Gillespie algorithm for the new context in i' to obtain $(\tau'_{i'}, j'_{i'}, i')$, the next rule $r_{j'_{i'}}$, to be used inside membrane i' and its waiting time $\tau'_{i'}$;
  - add the new triples $(\tau'_{i'}, j'_{i'}, i')$ in the list and sort this list according to each waiting time and iterate the process.
- **Termination**
  - Terminate simulation when time of the simulation t reaches or exceeds a preset maximal time of simulation.

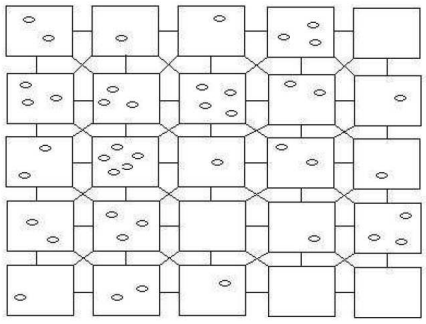
It is worth noting that this is a local algorithm in the sense that all computations only consider the number of objects and rules of a single compartment. The only remaining global computation is the location of the index of the smallest waiting time, which could be improved by keeping all reaction times in an *indexed priority queue*. The advantage of having local computations is that the algorithm is easily implemented in an event-driven object-oriented programming style, such an implementation could be multithreaded on a hyper-threading machine and would also lend itself to full message-passing implementation on a parallel computing cluster.

B. Quorum sensing in Vibrio Fischeri: a case study

Bacteria are generally considered to be independent unicellular organisms. However it has been observed that certain bacteria, like the marine bacterium *Vibrio fischeri*, exhibit coordinated behaviour which allows an entire population of bacteria to regulate the expression of certain or specific genes in a coordinated way depending on the size of the population. This cell density dependent gene regulation system is referred to as *quorum sensing* [25].

In [24] a model of the quorum sensing system in Vibrio fischeri using a multienvironment stochastic P system has been presented. The behaviour of a population of n bacteria placed inside a multienvironment with 25 different environments has been studied. Each bacteria is represented by a P system Π with only one membrane, and in the initial configuration of the model, m copies of Π will be distributed randomly over

the 25 different environments. Then, the system will evolve applying the rules from R_E in the environments and the rules from R_Π in the bacteria according to the multi-compartmental Gillespie Algorithm. One of the possible initial configurations for a population of 44 bacteria can be seen in the next figure.



The results of the model show that on the one hand bacteria remain dark at low cell densities and on the other hand in big size populations bacteria are able to sense the number of individuals and the population starts to emit light in a coordinated way. These results agree well with in vitro observations.

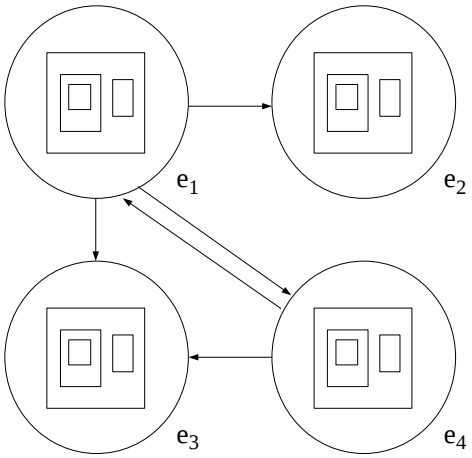
IV. Probabilistic P systems based models

A multienvironment functional P system with active membranes of degree (q, m, n) with $q \geq 1,\ m \geq 1,\ n \geq 1$, taking T time units, $T \geq 1$, is said to be probabilistic if:

- $m = n$.
- The functions $p_{(k,j,j')}$ associated with the rules $(\Pi_k)_{e_j} \xrightarrow{p_{(k,j,j')}} (\Pi_k)_{e_{j'}}$ are constant functions equal to zero; that is, we can assume that there is no communication rules between environmentes of the previous form.
- The functions $p_{(x,j,j')}$ associated with the rules $(x)_{e_j} \xrightarrow{p_{(x,j,j')}} (y)_{e_{j'}}$ are computable functions whose domain is $\{1, \ldots, T\}$ and its range is contained in $[0, 1]$. These rules verify the following: for each e_j, if $\{e_{j_1}, \ldots e_{j_z}\}$ is the set of reachable nodes from e_j, then $\sum_{i=1}^{z} p_{(x,j,j_i)}(t) = 1$, for each $x \in \Sigma$ and $t = 1 \leq t \leq T$;
- For each $r \in R_\Pi$ and for each k, $1 \leq k \leq m$, $f_{r,k}$ is a computable function whose domain is $\{1, 2, \ldots, T\}$ and its range is contained in $[0, 1]$ verifying the following:

  For each $u, v \in M(\Gamma)$, $i \in \{0, \ldots, q-1\}$ and $\alpha \in \{0, +, -\}$, if $r_1, \ldots, r_z$ are the rules from R whose left–hand side is $u[v]_i^\alpha$, then $\sum_{j=1}^{z} f_{r_j}(t) = 1$, for each $t,\ 1 \leq t \leq T$.

  That is, tor each $r \in R$ and $t,\ 1 \leq t \leq T,\ f_r(t)$ represents a probabilistic constant associated with the rule r at moment t.
- At the initial moment, each environment of the system contains exactly a functional P system Π_k all of them with the same skeleton, and $\mathcal{M}_{0k}, \ldots, \mathcal{M}_{q-1,k}$ represent the initial multisets of Π_k, as appear in the next picture.



When a communication rule between environments $(x)_{e_j} \xrightarrow{c} (y)_{e_{j'}}$ is applied, object x pass from e_j to $e_{j'}$ possibly modified into an object y. When a communication rule between environments $(\Pi_k)_{e_j} \xrightarrow{c'} (\Pi_k)_{e_{j'}}$ is applied, the P system Π_k pass from e_j to $e_{j'}$. The role played by the constants c and c' is given by the simulation algorithm that captures the semantics of the model.

A. A simulation algorithm for probabilistic P systems

Nezt, we present a simulation algorithm dealing to capture the seamntics of a computaional model based on probabilistic P systems.

(a) Rules are classified into sets so that all the rules belonging to a class have the same left--hand side.

(b) Let $\{r_1, \ldots, r_t\}$ be one of the said sets of rules. Let us suppose that the common left-hand side is $u\ [v]_i^\alpha$ and their respective probabilistic constants are $c_{r_1}, \ldots, c_{r_t}$.

(c)) Let $F(N, p)$ be a function that returns a discrete random number within the binomial distribution $B(N, p)$

- It is computed the highest number N so that u^N appears in the father membrane of i and v^N appears in membrane i.
- let $d = 1$
- For each k $(1 \leq k \leq t-1)$ do
  - let c_{r_k} be $\frac{c_{r_k}}{d}$
  - let n_k be $F(N, c_{r_k})$
  - let N be $N - n_k$
  - let q be $1 - c_{r_k}$
  - let d be $d * q$
- let n_t be N
- For each k $(1 \leq k \leq t)$, rule r_k is applied n_k times.

B. The bearded vulture: a case study

A multienvironment probabilistic P System with active membranes which models an ecosystem related with Scavenger birds, and that is located in the Catalan Pyrenees,

has been presented in [2]. Specifically, the real ecosystem under investigation consists of 17 different types of animals corresponding to 13 species which compose. Two types of animals for the Red Deer have been considered due to the fact that males are highly valued by hunters and this implies that the mortality rate of males is higher than that of females. Also two types of animals for domestic ones (except for horses) have been considered because of some of them spend only six months in the mountain. By using this kind of P System, it has been possible to study the dynamics of the mentioned ecosystem adding new ingredients. That framework allows us to analyze how the ecosystem would evolve when different biological factors were modified either by nature or through human intervention.

A JAVA software tool with a friendly user-interface sitting on the P-Lingua 2.0 JAVA library [6] has also been developed. This application implements the simulation algorithm previously described, and it provides a flexible way to check, validate and improve computational models of ecosystem based on P systems instead of designing new software tools each time new ingredients are added to the models. Furthermore, it is possible to change the initial parameters of the modeled ecosystem in order to make the virtual experiments suggested by experts. These experiments will provide results that can be interpreted in terms of hypotheses. Finally, some of these hypotheses will be selected by the experts in order to be checked in real experiments.

V. Conclusions and Future Work

P systems provide a high level computational modeling framework for population biology which integrates the structural and dynamical aspects of cellular systems in a comprehensive and relevant way while providing the required formalization to perform mathematical and computational analysis.

A P systems based general framework for modeling ecosystems dynamics is presented. Syntax has been specified by means of multienvironment P systems and the semantics is captured by using a probabilistic strategie. A real ecosystems have been briefly described as case study: scavenger birds at the Catalan Pyrenees. We are working on the design of a software tool implementing new simulation algorithms improving the previous one, specifically implementing real parallelism. These algorithms enables us to run simulations of the model and then validate it experimentally.

Acknowledgment

The authors wish to acknowledge the support of the project TIN2009-13192 of the Ministerio de Ciencia e Innovación of Spain, cofinanced by FEDER funds, and the support of the project of excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200.

References

[1] M. Cardona, M. A. Colomer, M.J. Pérez–Jiménez, D. Sanuy, A. Margalida. Modelling ecosystems using P systems: The Bearded Vulture, a case of study. *Lecture Notes in Computer Science*, 5391 (2009), 137–156.

[2] M. Cardona, M.A. Colomer, A. Margalida, I. Pérez–Hurtado, M.J. Pérez–Jiménez, D. Sanuy. A P system based model of an ecosystem of some scavenger birds. *Lecture Notes in Computer Science*, 5957 (2010), 182–195.

[3] M. Cardona, M.A. Colomer, A. Margalida, A. Palau, I. Pérez–Hurtado, M.J. Pérez–Jiménez, D. Sanuy. A Computational Modeling for Ecosystems Based on P Systems. *Natural Computing*, online version (`http://dx.doi.org/10.1007/s11047-010-9191-3`).

[4] S. Cheruku, A. Paun, F.J. Romero, M.J. Prez-Jimnez, O.H. Ibarra. Simulating FAS-induced apoptosis by using P systems. *Progress in Natural Science*, **17**, 4 (2007), 424-431.

[5] F. Fontana, L. Bianco, V. Manca. P systems and the modelling of biochemical oscillations. *Lecture Notes in Computer Science*, **3850** (2005), 199–208.

[6] M. García–Quismondo, R. Gutiérrez–Escudero, I. Pérez–Hurtado. P–Lingua 2.0: added features and first applications. *Lecture Notes in Computer Science*, **5957** (2010), 264-288.

[7] M.A. Gibson, J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry*, **104**, 25 (2000), 1876–1889.

[8] D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, **22** (1976), 403–434.

[9] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, **81** (1977), 2340–2361.

[10] D.T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, **188** (1992), 404–425.

[11] D.T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, **115** (2001), 1716–1733.

[12] D.T. Gillespie, L. Petzold. Improved leap-size selection for accelerated stochastic simulation. *Journal of Chemical Physics*, **119** (2003), 8229–8234.

[13] Kell, D.B. (2002) Genotype-phenotype mapping: genes as computer programs, *Trends in Genetics*, **18**, 11, 555 – 559.

[14] Kell, D.B. (2005) Metabolomics, machine learning and modelling: towards an understanding of the language of cells, *Biochemical Society Transactions*, **33**, 520 – 524.

[15] T.C. Meng, S. Somani, P. Dhar. Modelling and simulation of biological systems with stochasticity. *In Silico Biology*, **4** (2004), 293–309.

[16] L. Nunes de Castro (ed.) *Fundamentals of Natural Algorithms. Basic concept, algorithms and applications*, Chapman and Hall/CRC, Taylor and Francis Group, Boca Raton, FL, 2006.

[17] G. Păun. *Membrane Computing: An Introduction.* Springer-Verlag, 2002.

[18] M.J. Pérez-Jiménez, F.J. Romero-Campero. P systems, a new computational modelling tool for systems biology, *Transactions on Computational Systems Biology VI, LNBI*, **4220** (2006), 176–197.

[19] D. Pescini,D. Besozzi, G. Mauri, C. Zandron. Dynamical probabilistic P systems, *International Journal of Foundations of Computer Science*, **17**, 1 (2006), 183–195.

[20] A. Phillips, L. Cardelli. A correct abstract machine for the stochastic Pi-calculus. *Electronical Notes in Theoretical Computer Science*, (2004).

[21] C. Priami, A. Regev, E. Shapiro, W. Silverman. Application of a stochastic Name-Passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, **80** (2001), 25–31.

[22] A. Regev, E. Shapiro. The π-calculus as an abstraction for biomolecular systems. *Modelling in Molecular Biology*,2004, pp. 219–266.

[23] F.J. Romero-Campero, M.J. Pérez-Jiménez. Modelling gene expression control using P systems: The Lac Operon, a case study. *BioSystems*, **91**, 3 (2008), 438–457.

[24] F.J. Romero–Campero, M.J. Pérez–Jiménez. A model of the quorum sensing system in Vibrio Fischeri using P systems. *Artificial Life*, **14**, 1 (2008), 95–109.

[25] C.M. Waters, B.L. Bassler. Quorum Sensing: Cell-to-Cell Communication in Bacteria. *Annual Review of Cell and Developmental Biology* **21** (2005), 319–346.

[26] The Stochastic Pi-Machine `http://www.doc.ic.ac.uk/~anp/spim/`