Bio-inspired Artificial Creatures for Populating Virtual Worlds

Nedjma Djezzar, Noureddine Djedi
LESIA Laboratory/Med Khider Biskra University
Department of Computer Science
Biskra, Algeria
nedjmas@gmail.com,djedi_nour@yahoo.fr

Sylvain Cussat-Blanc, Hervé Luga, Yves Duthen
Institut de Recherche en Informatique de Toulouse
Université de Toulouse-CNRS-UMR 5505
Toulouse, France
{sylvain.cussat-blanc, herve.luga, yves.duthen}@irit.fr

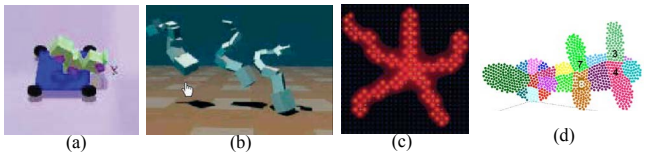
Abstract—**With the aim of populating virtual worlds with more and more different artificial creatures, we propose an ontogenetic and phylogenetic hybrid model. This model must generate bio-inspired artificial creatures, from an initial single cell and possesses metabolism, morphology, and behavior. The initial purpose of our work is to generate organisms which are thereafter used to define complete creatures. Thus, we introduce in this paper, a cellular development model using the two approaches from the field of ontogenesis systems: grammatical and cell chemistry approaches. Ontogenesis models simulate the development of multicellular organisms starting from a single cell. For the grammatical part, we propose an alternative to parametric L-systems (*APL-systems*) in order to simulate morphogenesis of organisms according to their internal states. The developed organisms have a metabolism using environmental molecules to grow and act. Moreover, they are able to exhibit almost perfect self-repairing characteristics when subjected to severe damage.**

Keywords- virtual world; ontogeny; phylogeny; metabolism; evolutionary parametric L-systems

I. INTRODUCTION AND MOTIVATIONS

A number of different models exist to populate virtual worlds. These models use different levels of abstraction to produce creatures of various shapes and sizes.

Phylogenetic approaches attempt to evolve sophisticated behaviors through the simultaneous evolution of the brains and bodies of creatures in a 3-D physically simulated world. The creatures are assembled from pieces such as blocs or sticks and evaluated on their ability to walk, swim, or even make a skateboard [1]. This research is originated by the seminal work of Karl Sims [2], who showed in 1994 that artificial evolution was able to create fully virtual creatures. Many authors such as Maciej Komosinski [3] or Hod Lipson [4] have since extended this work.



(a) (b) (c) (d)

Figure 1. Some artificial creatures. (a [1], b [2]): phylogenetic approach, (c [6], d [7]) ontogenetic approach.

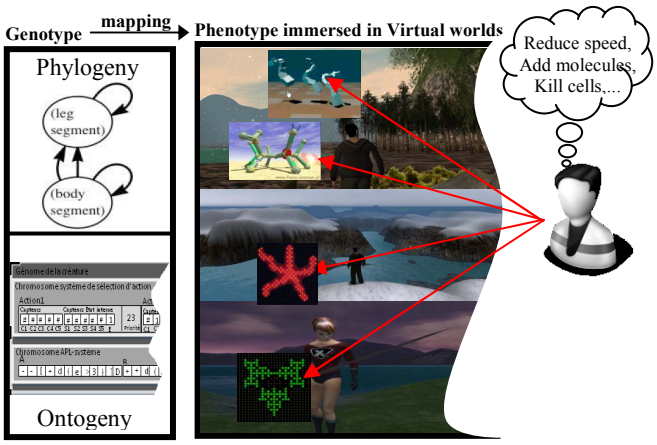


Figure 2. Two different aproaches to generate artificial creatures interactiong in virtual worlds: phylogeny, ontogeny. Interaction can occur with the objetcs of the environment or with the user.

Instead of using high-level component like blocs as organs of creatures, *ontogenetic* approaches [5, 6, 7] moves closer to nature, considering the creature as the result of a developmental process of a single element: a cell. Inspired by biology, the purpose of this subfield of artificial life, is to generate more complex artificial beings interacting in virtual worlds but also to advance evolutionary robotics with prototypes capable of self-modeling, self-reproduction and self-repairing.

Furthermore, the previous work in artificial ontogeny falls into two broad categories, grammatical [8, 9, 10] and cell chemistry approach [6, 11, 12]. This classification proposed by Stanley and Miikkulainen [13] is based on the dynamics used to control development i.e. biological development mechanisms that have been simulated or taken into account in the model.

Indeed, grammar based techniques simulate development via a cell lineage mechanism, simulated by a set of production rules applied successively to transform an initial cell on a multicellular organism. Cell chemistry approaches generally utilize lower level representations and are more strictly motivated by the biological mechanisms of development such as: diffusion, reaction-diffusion,

978-1-61284-732-0/11/$26.00 ©2010 IEEE

expression and regulation of genes, metabolism, genetic regulatory networks, differentiation, cell division, etc.

Thus, we propose in this paper a unified model based on these two approaches to take advantage of both. As a grammatical approach, L-systems are considered to be an appropriate formalism for describing many growth processes in organisms [14]. They are convenient for describing cell lineage[1] and genetic control of cell division [15]. They allow a necessary and very suitable distinction between genotype and phenotype, and provide a well-defined process (morphogenesis) for generating the latter from the former. Repetition and similarity, intrinsic concepts of L-systems, are also necessary ingredients for natural morphogenesis. In addition, many studies have successfully explored the evolution of these systems notably for the design of models that best describe natural target structures. We cite: the development of plants and flowers [10, 16], modeling the blood vessels of the eye [17] or proteins [18], evolutionary development of neural networks [8, 9] and also the generation of artificial creatures [3] and real robots [4]. All these reasons motivate us to take advantage of the flexibility of evolutionary L-system in a bio-inspired cellular developmental model.

So, in order to populate virtual worlds with complete creatures possessing a morphology, a metabolism and a behavior, we propose an ontogenetic and phylogenetic hybrid approach. To develop initially multicellular organisms from a single cell (the ontogeny part) and which could be used thereafter to synthesize complete creatures (the phylogeny part), we introduce in this paper, a bio-inspired cellular development model unifying grammatical and cell chemistry approaches. To develop multicellular creatures, we simulate their metabolism (section B), their cellular and environmental interactions (section C), their morphogenesis (section D) and their evolution (section E).

II. THE CELLULAR DEVELOPMENTAL MODEL

A. Embryonic environment and artificial cells

Our artificial multicellular creatures evolve in an environment represented by a 2-D toric grid similar to the one presented in [6]. This environment contains various molecules which spread into the grid. Each molecule can diffuse towards the eight neighboring points in the grid. Diffusion acts in two stages as illustrated in figure 3:

- First, the molecule diffuses towards the four cardinal points.
- Then, if the quantity of molecules is sufficient, the molecule diffuses on the diagonals.

The cells evolve in the embryonic environment and more precisely in its diffusion grid. Each cell has: a shape (diamond, circle,...), an internal state which represents the cell constitution of intracellular molecule and energy level, a

[1] Lineage mechanisms are employed where an individual module determines its own fate using information passed from parent to child module.

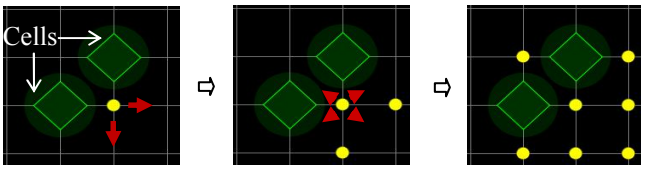


Figure 3. The diffusion of molecules in the environment.

set of sensors positioned on its membrane (described in section C), a list of capacities or actions (described in section C) and an action selection system allowing the cell to choose the best action to perform at every moment of the simulation (described in section C).

B. Metabolism

For the purpose of simulating the metabolism of our artificial multicellular creatures, our model integrates a simplified artificial chemistry. Formally, an artificial chemistry is defined by a triple (*S; R; A*) [19], where:

- S is the set of all possible molecules,
- R is a set of collision or reaction rules representing the interaction among the molecules,
- and A is an algorithm describing the reaction vessel or domain and how the rules are applied to the molecules inside the vessel.

In our model, we define a simple chemical system consisting of a finite number of molecules that we also call substrates. These molecules diffuse in the grid and have a unique name (A, B, C...) and different properties such as diffusion rate, color and type (intracellular or environmental). Molecules can interact via a finite set of reaction rules.

The algorithm:

molecules: $S= \{s_1, s_2, \ldots\}$;

reactions: $R= \{r_1, r_2, \ldots\}$;

begin

while (not terminate())

begin

n:= random_ number ([S]);

s_1 := random_ molecule (S);

s_2 := random_ molecule (S);

s_n := random_ molecule (S);

if ($\exists$ r = ($a_1s_1, a_2s_2, \ldots, a_ns_n \rightarrow a'_1s_1', a'_2s_2', \ldots, a'_ms_m'$) $\in$ R

and can_be_triggered (r))

then execute r;

end while

end

To simulate the dynamics of a population of molecules, we propose a stochastic molecular collision approach. A

typical algorithm takes a sample of molecules randomly from the set S and checks whether a rule r ∈ R can be applied. If so, the molecules are replaced by the right hand side molecules given by *r*. Otherwise, no rule can be applied and the process is repeated.

Note that these reaction rules can be triggered only inside the cells. Thus, when a cell carries out the molecule transformation action (described in next section), using some molecules, a reaction rule creates new molecules by consuming or producing energy. For example, the transformation A + 2B → C (-62) produces one unit of C molecule, with a unit of A molecule and two units of B molecule. This transformation consumes 62 units of vital energy. The produced molecules can be rejected in the environment and thus another cell can absorb and transform them into other molecules.

C. *Environmental and cellular interactions*

The cells can interact with the environment and also with other cells via an action selection system. To simulate this interaction, cells are provided with membranous sensors and can carry out various cellular actions.

Sensors are positioned on the membrane of each cell. They measure the amount of molecule available in its Von Neumann neighborhood. Thus, for each possible environmental molecule, the cell has an associated sensor. Only the sensor corresponding to a given molecule can measure its density. For example, in Figure 4, the cell has sensors for A and D molecules in the bottom corner. The results of the measure of the corresponding molecule densities are: 1 unit of A molecule and 2 units of D molecules because of the presence of two units of D molecule in the bottom corner of the cell.

Cellular actions, that we simulate in our model, are as follows:

1) Absorption or release of a molecule. These actions can trigger (or respectively, be triggered by) a transformation of molecules.

2) Molecule transformation. This action can trigger a stochastic collision of molecules within the cell (The algorithm of the cell chemistry). The molecule transformation consumes or produces energy.

3) Survival. This action allows the cell to await a signal from the environment or its internal state. It consumes vital energy.

4) Apoptosis. This action allows the cell to commit suicide if it does not have enough vital energy to survive.

5) Cell division: "d" is a symbol denoting a cell division action. When a cell divides, it creates a new daughter cell towards the direction specified by the L-systems controlling the morphogenesis of the creature. Cell division can be carried out by the cell, only if the following conditions are respected:

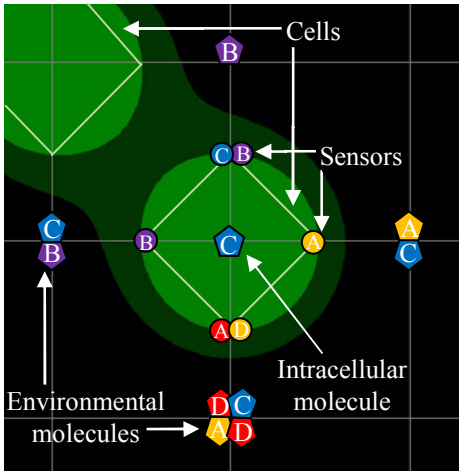


Figure 4. The artificial cell in its environment. It contains sensors (circles) to measure the density of valid molecules (pentagons) in its neighborhhod.

- The cell must have enough vital energy to divide.
- The quantities of certain molecules, necessary for the creation of a new cell, must exceed a certain threshold.
- The orientation of cell division corresponds to a position non-occupied by another cell.

The action selection system allows the cell to select the best action to perform at every moment of the simulation. The action selection system of each creature is inspired by Pittsburgh [20, 21] classifier system. It uses a genetic algorithm for its evolution and its evaluation which is applied to a population of classifier systems. The individuals are thus a set of classifiers that are trying to solve the problem. In our action selection system, each classifier has three parts: condition, action and priority:

- *The condition* serves to capture the signal coming from the environment or the internal state of the cell. It constitutes the cell sensors.
- *The action* part determines the action to be carried out by the cell if the corresponding condition is fulfilled.
- *The priority* allows to choose only one action if several classifiers are active at the same time. The higher the coefficient is, the more probable is the selection of the rule.

Each creature's action selection system works by using data from the cell sensors and matching them with the condition part of each classifier to produce a matching set. Once completed, it selects the best action to perform from this set, using the highest priority.

D. *Morphogenesis*

The growth of an artificial creature occurs via the cell division. The orientation of cell division is specified by the L-system growth rules. L-systems are based on formal

grammars with recursive applications of production rules. The possibility of simultaneous productions reflects the biological motivation of L-systems, intended to capture cell division in multicellular organisms [14]. Although, our representation is somewhat different from that of L-systems. Indeed, instead of using a symbol to represent a cell, we use a symbol to represent a cell division action. Moreover, the L-systems that we use are an alternative to parametric L-systems.

1) L-systems alphabet: The alphabet we use is described in the table below. Uppercase characters are non-terminal symbols and denote the predecessors of rules, while lowercase letters represent terminal symbols.

2) APL-systems: To simulate the growth of our artificial multicellular organisms, we propose a variant of parametric L-systems, which we call APL-systems (Alternative Parametric L-systems). This extension to basic L-systems allows the simulation of the development of multicellular organisms according to the internal state of each cell. However, unlike traditional parametric L-systems, all production rules can be triggered. The predecessors of rules do not have pre-conditions unlike terminal symbols which do have pre-conditions.

Formally, an APL-systems can be defined as an ordered quadruplet (V,Σ,ω,P), where:

- $V=V1\cup V2$ is the alphabet of L-systems, with V1 is the set of non-terminals and V2 is the set of terminals,
- Σ is the set of formal parameters,
- $\omega\in$ V1 is the axiom,
- $P\subset V\times(V1\times(V2\times C(\Sigma)))^*$ is a finite set of production rules. V is the set of the production predecessors,
- and $V1\times(V2\times C(\Sigma))$ is the set of the production successors where $C(\Sigma)$ is the set of preconditions.

A precondition $C(\Sigma)$ attached to a terminal means that it cannot be interpreted unless its precondition is fulfilled. For example, the terminal "d" cannot be interpreted in a cell division action unless the cell division conditions are respected by the cell concerned. In this manner, this alternative will allow the interpretation of all the substrings, produced by the application of a production rule in a given generation, to the first terminal that does not fulfill its precondition. Whereas, in traditional parametric L-systems, all substrings generated by this rule would not be generated since the rule would not be triggered. In this way, all cells that wish to divide, during this generation, can do so if they fulfill their preconditions.

TABLE I. THE L-SYSTEMS ALPHABET

Symbol	Function
A …Z	Non-terminals representing predecessors of rules.
d	Terminal denoting a cell division action.
+,-	Terminals representing a two-dimensional rotation.
[,]	Terminals indicating branching.

E. Evolution

To find the creature the most adapted to a specific problem, we use a genetic algorithm. Each creature is coded with a genome composed of two different chromosomes: the *APL-systems*, specifying the rules of growth, and the action selection system, that contains a rule list to apply available actions. The developed creature is evaluated at the end of the simulation.

III. EXPERIMENTS AND RESULTS

We have implemented our model in Java using a multi-threaded architecture. All cells are coded as independent threads running in parallel and sharing common resources: environmental molecules.

A. Development of simple organisms

We wish to synthesize multicellular creatures with the proposed developmental model. Self-similarity is used as a selection criterion for the morphological development of our multicellular organisms. We made this choice, to produce creatures similar to those of Karl Sims [2]. From the long view perspective, they could be immersed in a physical simulator and have a high level behavioral module enabling them to move in their environment.

In order to reduce the convergence time of the genetic algorithm and have a better comprehension of the results, the evolutionary experiments are conducted in two stages Thus, initially, we use a fitness favoring bilateral self-similarity (The results of this experiment are shown by Figure 5). Then, once the best chromosomes describing cell lineage are found, the second stage consists in teaching the creatures how to survive in the environment by developing their metabolisms and to study their ability to use this metabolism for growth thanks to our *APL-systems*.

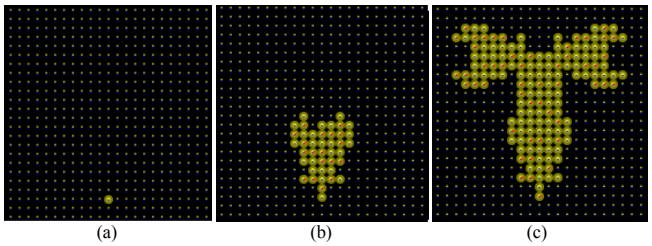
To develop our multicellular organisms, we define an artificial chemistry model with a set of molecules composed of 3 molecules:

- A and B: blue (respectively yellow) environmental molecule used by the organism as nutriment.
- C: red intracellular molecule produced by cells using environmental molecules and used as material of cell division.

The grid size is 100*100. The diffusion property of the molecule will spread rapidly environmental molecules in the grid. The set of reactions of the artificial chemistry consists of two reactions, R = {r1: A + 2B → C (-10), r2: A → {} (+ 60)}. The mother cell is initialized with some units of vital e-

(a) (b) (c)

Figure 5. morphologies emerged from the evolution of L-system chromosome.



(a) (b) (c)

Figure 6. Growth of an organism (artificial crayfish) composed of 146 cells. (a) Beginning of the simulation. (b) The organism develops by using environmental resources. (c) End of the development

nergy and C molecule (so it can survive in the early moments of the simulation).

The genetic algorithm uses tournament selection. Fitness function of the organism is divided into two sub-functions:

- The longevity of the organism, *length* (duration of the simulation in milliseconds)
- The size of the organism measured by the number of cells composing it, *nbCells*.

The final evaluation function is given by the formula:

*fitness=a*length+b*nbCells/(a+b), with a= 10, b=100.*

The coefficients of this formula are given to more take into account the final goal of the organism which is its capacity to develop than its metabolism. The goal of the genetic algorithm is to maximize this fitness. The parameters of this algorithm are as follows:

- selection algorithm: 6 tournament selections with elitism,
- mutation rate: 5%; crossover rate: 65%,
- substitution algorithm: worst individuals,
- population size: 100 individuals.

Figure 6 shows the development of the best creature produced by evolution (an artificial crayfish2). Development starts from a single mother cell and proceeds over discrete time steps. It stops either when a maximum time step is reached or when an individual embryo exhausts its initial energy. When the mother cell divides, it creates a new cell which is placed in the position specified by the APL-systems, controlling the growth of the multicellular organism. At the initial state, the expression of the APL-system starts with the axiom. The cell, at this state, uses its initial intracellular molecules to create its first daughter cells. Subsequently growth requires the acquisition of the environment resources.

Observing the action selection system produced by the genetic algorithm, we notice that it could be possible to produce other creatures with the same chromosome. To verify the hypothesis, we decide to develop another creature: a bat. To do that, we keep the same molecules and the same possible actions. Using the L-systems chromosome of Figure 5 (c), we launch the simulation and we obtain the creature shown by Figure 7 (j).

[2] Videos of all presented creatures in this paper are available on the website http://siva.univbiskra.net/Fichiers/SIVA_Videos.htm.

B. Self-repairing abilities

Our resulting organisms show a remarkable ability to repair themselves when subjected to severe damage.

To demonstrate this ability, we remove a small part of the artificial bat's cell structure in two phases: during and after its development. In the first experiment, we inflict a wound in the center of the creature. The results show that the organism reacts positively by regenerating dead cells In the second experiment, the wound is performed during the development of the creature, and particularly on the newly created cells. Figure 7 illustrates this experiment. In this case, the regeneration of the dead cells takes more time since the cell division requires energy, and the newly created cells not yet having interactions with the environment, do not have enough energy to divide. Before the end of development, we proceed again with two simultaneous injuries:

- a first in the lower center of the creature,
- and a second on the upper left.

In the same manner, the organism was able to adapt to environmental changes (Figure7 (h)-(j)). The left side branch of the organism will, however, take more time to develop, because the hole here affects a greater number of cells (29 cells) than the one in the center (17 cells).

I. CONCLUSION AND FUTURE WORKS

We proposed in this paper, a cellular developmental model based on the combination of two approaches recognized in the field, grammatical and cellular chemistry based approaches. On the one hand, this model simulates the major elements of a cellular chemistry model, such as chemical reactions in a simplified manner, the diffusion of molecules in the environment, cellular and environmental interactions, and some cellular actions such as cell division or apoptosis. On the other hand, morphogenesis is based on a grammatical approach using growth rules coded in *APL-systems*, an alternative parametric L-system, which we proposed to simulate the adaptive growth of multicellular organisms.

More bio-inspired than biologically plausible, this model is able to produce various artificial creatures, starting from a single cell. The developed creatures, have a morphology generated by a lineage mechanism, based on L-systems, and a metabolism allowing them to grow and act. In the lineage of our previous works [6] the proposed model is among the rare cellular development models taking metabolism into account, often omitted in the classical models. However, metabolism is primarily essential for the integrity of each multicellular organism, because it allows each cell composing the organism to be survived. Moreover, our model is able to exhibit almost perfect self-repairing proprieties, if the organism is wounded afterwards or even during its development.

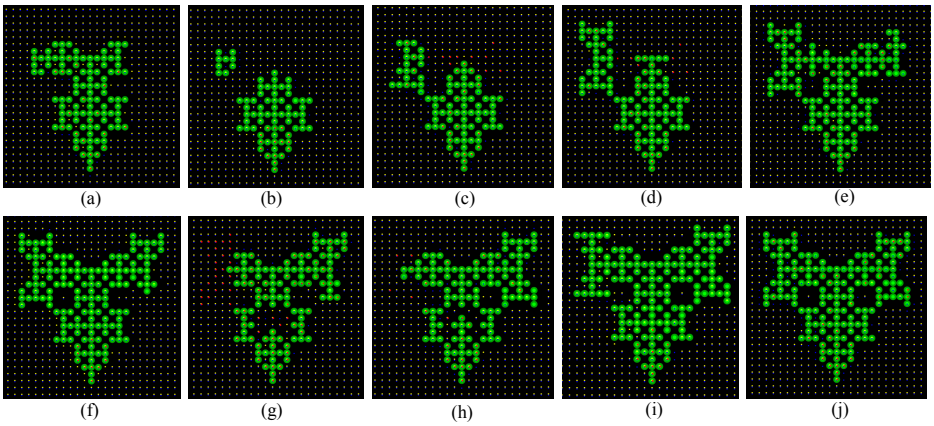


Figure 7. Wounds inflicted during the development of the artificial bat. (a) The creature develops. (b) Realization of a central hole in the newly created cells. (c, d, e, f) The creature regenerates itself. (g) Simultaneous realization of two wounds: the first on the left lateral branch of the creature, the second in the lower center of the creature. (h, i, g) The creature is regenerated.

This model proposes the use of L-systems, for directing the morphogenesis of an artificial multicellular creature. We made this choice, in order to explore the potential of these systems widely used with success in many fields. Furthermore, our *APL-system* is a unique alternative to parametric L-systems. It has the particularity of being independent of the proposed model and can thus be used in a broad range of applications which use parametric L-systems.

The automatic design of artificial creatures as artifacts in virtual worlds or real machines exhibiting some properties of living organisms represents a considerable challenge. Compared to Karl Sims's creatures [2], metabolism introduced into our cellular developmental model can provide more autonomy to these models of creatures, because metabolism is a means of obtaining energy. Thus, molecules, for example, can be considered as metaphor for batteries that artificial creatures can retrieve from the environment and use to perform specific tasks.

REFERENCES

[1] N. Lassabe, H. Luga, , and Y. Duthen, "A new step for evolving creatures," in IEEE-ALife'07, pp. 243–251.IEEE, 2007.

[2] K. Sims, "Evolving 3D morphology and behavior by competition," Artificial Life IV, MIT Press, Cambridge, MA, pp. 28–39, 1994.

[3] M. Komosinski, "Framsticks: a platform for modeling, simulating and evolving 3D creatures," In: Adamatzky A, Komosinski M, editors. Artificial Life Models in Software. New York: Springer, pp. 37-66, 2005.

[4] G.S. Hornby, H. Lipson and J. B. Pollack, "Generative Representations for the Automated Design of Modular Physical Robots," IEEE Transactions on Robotics and Automation, Vol. 19, August 2003.

[5] H. de Garis, "Artificial embryology and cellular differentiation," in Peter J. Bentley, e., editor, Evolutionary Design by Computers, pp. 281–295, 1999.

[6] S. Cussat-Blanc, H. Luga, and Y. Duthen, "Cell2organ: Self-repairing artificial creatures thanks to a healthy metabolism," in Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC), 2009.

[7] R. Doursat, Facilitating evolutionary innovation by developmental modularity and variability. In Proceedings of the 11th Conference on Geneticand Evolutionary Computation, pp. 683-690, 2009

[8] F.Gruau, D.Whitley and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," in Proceedings of the first annual conference on genetic programming, 1996.

[9] A. Cangelosi, S. Nolfi, and D. Parisi. "Artificial life models of neural development," A., (Ed.), On growth, form and computers, 2003.

[10] P. Prusinkiewicz and P. Barbier de Reuille, "Constraints of space in plant developmen," Journal of Experimental Botany, Vol. 8, pp. 2117-2129, 2010.

[11] A. Chavoya, and Y. Duthen, "A cell pattern generation model based on an extended artificial regulatory network," Biosystems, Vol. 12, pp. 95–101, 2008.

[12] S. Harding, J. F. Miller, W. Banzhaf, "Self modifying cartesian genetic programming: finding algorithms that calculate pi and e to arbitrary," in the Proceedings of the 12th annual conference on Genetic and evolutionary computation, pp-579–586, 2010.

[13] K.Stanley, R. Miikkulainen, "A taxonomy for artificial embryogeny," Artificial Life, Vol. 2, pp. 93–130, 2003.

[14] P. Prusinkiewicz, and A. Lindenmayer, "The Algorithmic beauty of plants," Springer, Berlin, 1990.

[15] K. Fleischer and A.H. Barr, " A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis," In Artificial Life III, C.G. Langton, Ed. Addison-Wesley, Redwood City, pp. 389–416, 1994.

[16] S. Bornhofen and C. Lattaud, "Competition and evolution in virtual plant communities: A new modeling approach," Natural Computing, Springer Netherlands, Vol. 8, n. 2, pp. 349–385, 2009.

[17] G. Kokai, Z. Toth and R. Vanyi, " Modelling blood vessels of the eye with parametric L-systems using evolutionary algorithms," Jurnal of Artificial Intelegence in Medecine, Berlin: Springer-Verlag, pp. 433–442, 1999.

[18] G. Escuela, G. Ochoa and N. Krasnogor, " Evolving L-systems to capture protein structure native conformations," Proceedings of the 8th European Conference on Genetic Programming (EuroGP), LNCS, Lausanne, Switzerland, Vol. 3447, pp. 73–83, 2005.

[19] P. Dittrich, J. Ziegler and W. Banzhaf, "Artificial chemistries-a review," Artificial Life Vol. 7, pp. 225 - 275, 2001.

[20] S. Smith, "A learning system based on genetic algorithms," PhD. Thesis, Computer Science Department, University of Pittsburgh, 1980.

[21] J. Bacardit, D. E. Goldberg and M. V. Butz, "Improving the Performance of a Pittsburgh Learning Classifier System Using a Default Rule," Lecture Notes In computer Science. pp. 291-307, 2007.