# From Single Cell to Simple Creature Morphology and Metabolism

Sylvain Cussat-Blanc, Hervé Luga and Yves Duthen
Institut de Recherche en Informatique de Toulouse
Université de Toulouse - CNRS - UMR 5505
2 rue du Doyen-Gabriel-Marty
31042 Toulouse Cedex 9, France
{sylvain.cussat-blanc, herve.luga, yves.duthen}@irit.fr

**Abstract**

In order to produce diversity in virtual creatures to populate virtual worlds, different techniques exist. Some of these use blocks or sticks. In this morphological approach, blocks and sticks can be considered as organs, which means body parts able to perform different functions. Another approach, artificial embryogenesis, consists in developing organisms from a single cell. In this paper, we propose a bridge between these two approaches : a model that will create creatures with a particular morphology and which is organized in organs. The creature development will start from a single cell. In this paper, we propose a unique model able to produce organisms that perform a specific function and to produce organisms with a user-defined morphology.

## Introduction

Several models exist for creating artificial creatures. These models use different levels of abstraction to produce creatures of various shapes and sizes. Whereas the morphological approach produces relatively large creatures as in (Sims, 1994; Lassabe et al., 2007), embryogenic models produce creatures composed of hundreds of cells starting from a unique cell (Chavoya and Duthen, 2007; Dellaert and Beer, 1994; Stewart et al., 2005).

This paper details our model of cellular development, *Cell2Organ* (Cussat-Blanc et al., 2007). For the purpose of creating complete creatures composed of different organs, we propose a model able to produce organisms that perform specific functions. These organisms respect the biological definition of an organ. In other words, they are a "specialized cell regrouping that performs specific function or a group of functions". Our model contains an environment with a simple artificial chemistry (Rasmussen et al., 2003; Dittrich et al., 2001; Hutton, 2007; Ono and Ikegami, 1999) and cells that perform different actions. Cells are able to self-replicating and to specialize themselves to optimize specific actions instead of others. Moreover, we show that *Cell2Organ* can also produce simple creature shapes. The final aim of our project is to develop a complete creature starting from a unique cell.

This paper is organized in four sections. Section 2 presents related works about artificial creatures development, presenting artificial morphogenesis, cellular automata and already existing works about artificial embryogenesis. Section 3 presents our model of cellular development, *Cell2Organ*, starting with a description of the environment functioning and the mechanisms used by our artificial cell to interact with the environment. Section 4 presents different experiments using this model. The possibilities of the model are shown by the development of two types of organisms : a primitive organ able to move substrate in the environment and two creatures with particular morphologies. These experiments point to the possibility of simulating, in a simplified way, different approaches to organism growth. In the final section, we conclude by outlining different possible development paths for this model.

## Related works

### Artificial morphogenesis

Several projects have tried to generate artificial creatures well adapted to their environment. For example, in his famous works, Karl Sims (Sims, 1994) uses blocks with different properties such as size, shape, contact sensor positions or block layout. Komosinski also creates Framsticks creatures (Komosinski and Ulatowski, 1999) using an equivalent architecture: sticks replace blocks but creature functioning is comparable to Karl Sims' work: he uses a neural network to coordinate creature movements. Nicolas Lassabe improved Sims' work by using a more complex environment (Lassabe et al., 2007). Lassabe's creatures are able to climb a stairway or to practice skateboarding.

The aforementioned creatures use high-level components to create their morphology and their behavioral controller. A more biological-inspired approach was introduced by Dawkins in (Dawkins, 1986). Using simple rules to draw continuous segments, he developed a model able to create small graphic creatures. The addition of behaviors in these simple life forms allows the creation of a complex 2-D virtual world (Ventrella, 1998) where small filiform creatures co-evolve in an environment composed of energy sources.

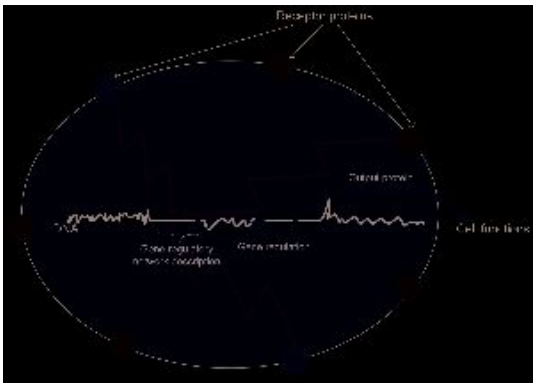

Figure 1: Scheme of the GRN action in cell duplication.

Each creature has a vital energy level and must survive in the environment, looking for food produced by the death of other creatures. This model produces a complete ecosystem with its own food chain. Creatures are also able to reproduce among themselves to create new life forms. EvolGL (Garcia Carbajal et al., 2004) is another 3D pond life project where creatures have different classes, such as herbivorous, carnivorous or omnivorous, which allows the emergence of survival strategies.

Using lower level components, cellular automata use neighborhood rules to evolve a cell matrix. The rules give the *t+1* state of each cell according to the cell neighbor's *t* state. Using this method, John H. Conway (Gardner, 1970) creates interesting patterns such as gliders, pulsars, etc.

### Artificial Embryogenesis

One of the first works on artificial embryogenesis was that of Hugo de Garis (de Garis, 1999). Using a cellular automaton, he developed 2D shapes. The cellular automata rules were evolved with a genetic algorithm. The aim was to generate desired shapes like letters.

Another important goal of artificial embryogenesis is cell specialization. Different works on cell specialization already exist. In most cases, they use a Genetic Regulatory Network (GRN), just as in nature.

In nature, the organism's cells can have different functions, all of which are specified in the organism's genome and regulated by a Gene Regulatory Network (GRN) (Davidson, 2006). Cells get input signals from the environment thanks to receptor proteins. The GRN, described in the organism's genome, uses these signals to activate or inhibit the transcription of different genes in the messenger RNA, the future cell's DNA protein template. The expression of these genes will specify the cell's functions. Figure 1 shows (in a simplified way) the functioning of the GRN.

This nature inspired model was designed by Banzhaf in (Banzhaf, 2003). In this work, each gene beginning is marked by a starting pattern, named "promoter". Before the coding of the gene itself, enhancer and inhibitor sites allow the regulation of its behavior. In (Chavoya and Duthen, 2007), Chavoya and Duthen introduced another model in which the gene regulation system is encoded at the beginning of the genome. It consists of a series of inhibitor sites, enhancer sites and regulatory proteins. The production of each regulatory protein is conditioned by the inhibitor/enhancer sites. The concentration of this protein determines the cell function's activation or inhibition : if the concentration level is over a certain threshold, the gene is activated and so are the corresponding functions.

A different approach is the Random Boolean Network (RBN) first presented by Kauffman (Kauffman, 1969) and reused by Dellaert (Dellaert and Beer, 1994). A RBN is a network where each node has a boolean state: activate or inactivate. The nodes are interconnected by boolean functions, represented by edges in the net. The state of a node at time $t + 1$ depends on its particular boolean function applied to the values of its inputs at time $t$. The mapping to the gene regulatory network is simple: each node of the net corresponds to a gene and each boolean function represents the activity regulation of the gene. The cell function will be determined during the interpretation of the genome.

Eggenberger Hotz (Eggenberger Hotz, 2004) imagines a concept able to produce a simple creature with a user defined shape able to move in an environment just using a GRN. Cells rhythmically emit molecules that modify the adhesion properties between cells and between cells and the environment. He develops a simple simulator and produces a T-shape that grows and move in the environment.

The aim of our work is to make a bridge between artificial morphogeny and artificial embryogenesis to produce virtual creatures. We decide to use the hypothesis that blocks and sticks can be considered as organs, that is to say body parts of the creature able to carry one or more specific functions. Using developmental techniques of creature growth, we could create these organs starting from a single cell. In this way, the cell must be able to specialize itself into a cell more adapted to the environment. The cell organization in tissues (that is in cell groups that have the same function) and then the tissue organization will allow the creation of organs. After creating a library of organs, we will just have to assemble them to create a creature adapted to the environment with a morphological approach. This paper presents the embryogenic approach of the problem, and especially the creature shape development. The next section details the model, starting with the environment and, then, showing the cell mechanisms.

## *Cell2Organ* : a cellular developmental model

### The environment

To reduce the simulation computation time, we implement the environment as a 2-D toric grid. This choice allows an important decrease in the simulation's complexity.

The environment contains different substrates. They spread in the grid, minimizing the variation of substrate quantities between two neighbor crosses of the grid. This spreading is enacted in two stages, as illustrated by Figure 2

- First, the substrate spreads to the 4 cardinal points.
- Then, if the substrate quantity is sufficient, the substrate spreads to the diagonal crosses.

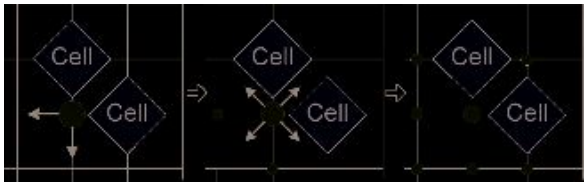


Figure 2: Example of spreading substrate in the environment.

Our model integrates a highly simplified model of artificial chemistry. Many works exist on artificial chemistry (Dittrich et al., 2001; Rasmussen et al., 2003). In these works, the artificial chemistry is highly developed and allows a good simulation of cell mechanisms. For example in (Ono and Ikegami, 1999), the cell division and the cell membrane formation and maintaining are highly realistic. However, the complexity of such a model is very great and does not support a high number of cells. In our model, the properties of artificial chemistry defined in (Dittrich et al., 2001) have been simplified.

Our molecules, named substrates, have different properties like diffusion speed or color, and can interact with other substrates. This interaction between substrates can be viewed as a typical chemical reaction: using different substrates, the transformation will create new substrates, emitting or consuming energy. For example, the transformation $2A + B \rightarrow C\ (+50)$ denotes that, using 2 units of substrate $A$ and 1 unit of $B$, a unit of $C$ is created, emitting 50 units of energy. To reduce the complexity at the maximum, the environment contains a list of available substrate transformations. The substrate reactions can only be triggered by cells. Then, in the previous example, from a biological point of view, $C$ can viewed as waste from a cell which has the ability to convert $A$ and $B$ into energy.

To modify this environment, cells interact with the environment. They have different abilities and must perform a global action defined by the user. This action can be very diverse: harvest substrate, modify environment, create shapes or simply survive as long as possible. The next section describes cell functioning.

## The cells

Cells evolve in the environment, more precisely on the environment diffusion grid. Each cell contains sensors and has different abilities (or actions). An action selection system allows the cell to select the best action to perform at any moment of the simulation. Finally, a representation of a GRN is inside the cell to allow specialization during duplication. Figure 3 is a global representation of our artificial cells.

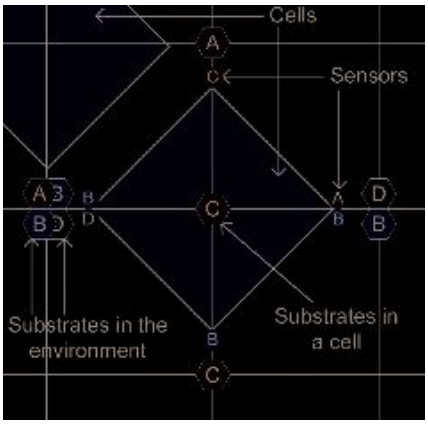


Figure 3: Scheme of a cell in an artificial environment. It contains substrates (hexagons) and corresponding sensors (circles)

**Sensors** Each cell contains different density sensors positioned at each cell corner. Sensors allow the cell to measure the amounts of substrates available in the cell's Von Neumann neighborhood. For each substrate in the environment, a corresponding sensor exists. Only this corresponding sensor can compute the density of the substrate. The list of available sensors and their position in the cell is described in the genetic code.

For example, in Figure 3, the cell has sensors for B and D substrates in the left corner. The results of the measure of the corresponding substrate densities are :

- 2 units for B substrate because of the presence of 2 units of B substrates in the left cross of the cell,
- 1 unit for D substrate.

**Actions** To interact with the environment, cells can perform different actions:

- The *substrate transformation* allows the cell to trigger a substrate reaction as previously described. To start, all the needed substrates on the left part of the equation must be present in the cell, that is, the needed substrates must be in the same intersection as the cell. In result of the reaction, the vital energy is increased or decreased (depending of the reaction properties), the needed substrates are destroyed and the new substrate is created.
- The cell can *absorb* or *reject* substrates in the environment. These actions allow the cell to move substrates

from one place to another. These actions, particularly the first , are important to trigger a substrate transformation.

- The *duplication* action allows the cell to create a new cell. We give details about this action in the next section.
- *Survive* is an action that allows the cell to wait for a signal from the environment to do something.
- *Apoptosis* allows the cell to autodestruct. This action can be useful to free a place for a more specialized cell for example.

The previous list is not final. Our model must be able to allow us to add new actions easily. Like sensors, all actions are not available for the cell: the genetic code will give the available action list.

Cells contain an action selection system. This system is inspired by classifier systems (Holland and Reitman, 1978). It uses data given by sensors to select the best action to perform. The selection system can be viewed as a rule database, where each rule is composed of three parts:

- The *precondition* describes when the action can be triggered. It is composed of a list of sensor value intervals that describe the best substrate densities in the neighborhood to trigger the action.
- The *action* gives the action that must be performed if the corresponding precondition is respected.
- The *priority* that allows the selection of only one action if more than one can be performed. The higher the coefficient, the more probable is the selection of the rule.

Action selection rules can be, for example :

$$\begin{array}{lll} (SensorA = 1) & and & (3 < SensorC < 7)\ and \\ (SensorB = 0) & \rightarrow & (ActionA)\ (23) \\ (SensorC = 3) & \rightarrow & (ActionB)\ (17) \\ & \rightarrow & (ActionC)\ (13) \end{array}$$

In this example, $ActionA$ will be performed if and only if $SensorA$ value is equal to 1 unit, $SensorB$ does not detect the presence of its associate substrate and $SensorC$ value is more than 3 units and less than 7. $ActionC$ does not contain a precondition. It means that this action can always be performed. The priority coefficients sort actions in the order $ActionA > ActionB > ActionC$ if different actions are possible.

In the list of possible actions, the cell can duplicate itself. We will now examine this action in detail.

**Duplication** The duplication is an action that can be performed by the cell if the next conditions are respected:

- The cell must have at least one free neighbor cross to create the new cell.
- The cell must have enough vital energy to perform the duplication. The vital energy level need is defined during the specification of the environment.
- A list of conditions can be added during the modelization of the environment. For example, some substrates can be needed to create a new cell.

The new cell created after duplication is completely independent and interacts with the environment. During duplication, the cell can be specialized to optimize a group of actions instead of others actions. In nature, this specialization is carried out by the GRN. In our model, we imagine a mechanism that plays the part of a GRN. Each action has an efficiency coefficient that corresponds to the action optimization level : the higher the coefficient, the lower the cost of vital energy. Moreover, if the coefficient is null, the action is not yet available for the cell. Finally, the sum of efficiency coefficients must remain constant during the simulation. In other words, if an action is optimized increasing its efficiency coefficient during duplication, another efficiency coefficient (or a group of them) has to be decreased.

The cell is specialized by varying the efficiency coefficients during duplication. A network built as follow gives the rules of these variations:

- the network's nodes represent cell actions with their efficiency coefficients,
- the network's edges are weighted. The edge's weight (a real number in the interval [0,1]) represents the efficiency coefficient quantity that will be transferred during the duplication.

Figure 4 is an example of our GRN. $(A, 35\%)$, $(B, 25\%)$, $(C, 17\%)$, $(D, 23\%)$ are cell actions with their associated efficiency coefficient. The edge between 2 actions represents the amount of efficiency coefficient that will be transferred during duplication. For example, the weighted edge between $A$ and $B$ means that after one duplication, 30 percents of the $A$ action efficiency coefficient will be transferred to the $B$ action. After four duplications, we can see that the actions $B$ and $C$ respectively have been optimized to the detriment of the actions $A$ and $D$. According to this simple example, we can say that the cell function of the organism has been specialized during the duplication process.

We have implemented this model in Java using a multithreaded architecture: cells are coded as independent threads. Cells can communicate using the environment and substrate exchanges. We made such a choice because of the development of massive parallel computer architectures such as multi-processor machines, increasingly connected in computation grid. This parellelization allows an increase in the number of tasks executed at the same time.

Our model must be able to generate two types of artificial creatures: organs and user defined shapes. The next experiments show that it is possible to accomplish this. The first

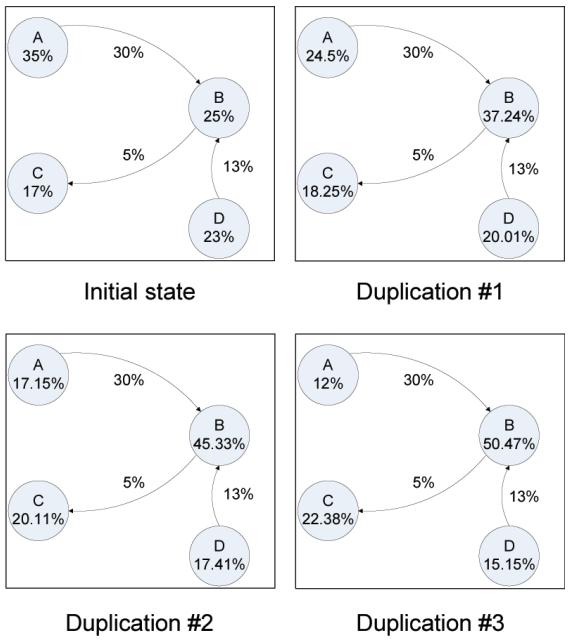


Figure 4: Modelization of an example of the Gene Regulatory Network. A, B, C and D are 4 actions with their efficiency coefficient. The transfer coefficients are given by the arrows.

experiment consists in developing a system able to move substrates in the environment whereas the second one creates simple shapes like starfish or jellyfish.

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 three different chromosomes:

- The list of available actions, a subset of the environment possible actions. This list allows the cell to activate or inhibit some actions.
- The action selection system that contains a list of rule to apply actions.
- The gene regulation network that allows cell specification during duplication.

The creature is tested in its environment that returns the score at the end of the simulation. To increase the genetic algorithm power, we use a computational grid parallelized genetic algorithm. This parallelization allows the computation of hundreds of creatures at the same time.

## Experiments

### Developing a transfer system

The first experimentation consists in developing a simple organ : a transfer system. In other words, the cell structure must be able to transport substrate from one point to another. To do that, we imagine an environment composed of 2 substrates:

- The red is the substrate that must be moved by the organism. This substrate has the specificity not to spread in the environment, in order not to impact on the organism work.
- A gray that will be used by the cell as fuel and duplication material.

The cell can perform the following actions:

- duplicate (needs one gray substrate and vital energy),
- absorb or reject substrate (consume vital energy),
- transform one gray substrate in vital energy.

We place 10 red substrate units into a specific cross of the grid (at the top left of the environment) and diffuse gray substrate all over the environment. The creature's score is given by the squared sum of the red substrate distance to the goal point (at the bottom right of the environment). The parameters of the genetic algorithm are:

- selection: 7 tournament competition with elitism,
- mutation rate: 5%; crossover rate: 65%,
- substitution: worst individuals,
- population size: 500 individuals,

Figure 6 shows the convergence curve of the genetic algorithm. It shows the variation of the minimum, the average and the maximum fitness of the population for each generation. The genetic algorithm's aim is to maximize fitness, which is the creature score. A relevant organism appears quickly. After 3 generations, the organism is able to move the red substrates but not in the right direction. After 10 generations, it is able to move closer to the goal point. The genetic algorithm converges after 22 generations (the average fitness is close to the best).

Figure 5 shows the development of the best organism[1]. We can see that only the cells on the way from the initial point to the end point are created. Moreover, the organism uses absorption and rejection actions to transfer the substrate gradually. Cells that overtake the final point die quickly so as not to interact in the transfer. During the convergence of the genetic algorithm, it is interesting to observe the evolution of the organism strategy towards the best solution. The first step is to learn to survive in the environment, absorbing gray substrate and transforming it in vital energy. The next step is to learn to duplicate in the right direction. Intermediate solution organisms are able to transport the red substrate

[1] Videos of all presented creatures in this paper are available on the website http://www.irit.fr/∼Sylvain.Cussat-Blanc

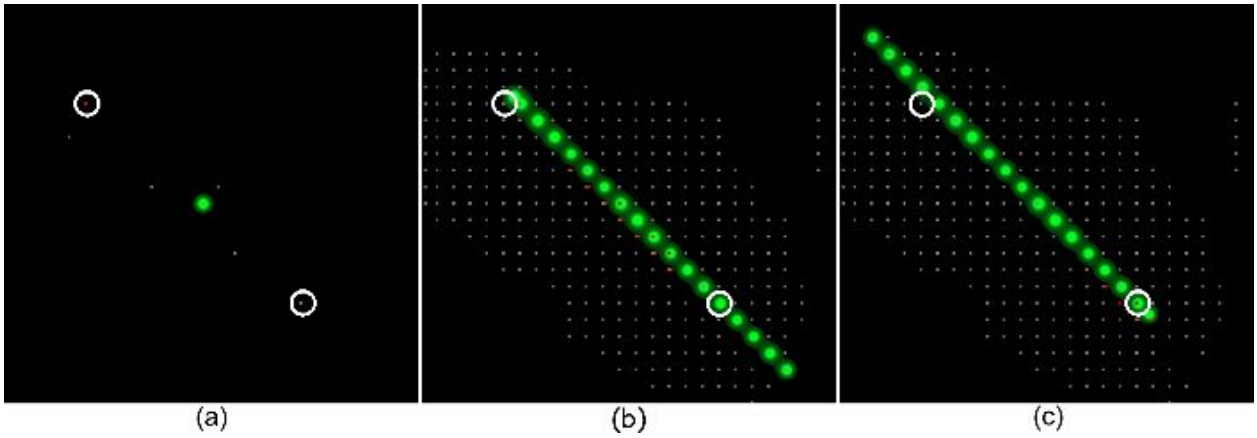


Figure 5: Our artificial transfer system. (a) Beginning of the simulation. (b) The creature develops itself to create the structure and begin the substrate transfert. (c) The creature transfers the substrate from the initial state (circle on top left) to the final state (circle on bottom right).

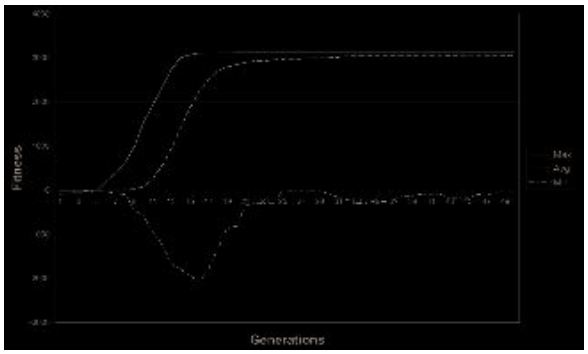


Figure 6: Smooth curve of the minimum, average and maximum organism fitness. The genetic algorithm must minimize the sum of the squared distance from the red substrate to the goal point.

from the initial point near to the goal. The organism also develops itself throughout the environment, scattering some units of the substrate in the environment. As shown in Figure 5, this organism deploys itself only on the best trajectory, decreasing the substrate scattering probability.

## Creating simple shapes

In this experiment, we want to generate simple creatures with a user-designed morphology. The goal of such an experiment is to simulate the growth of more complex creatures, like those of Sims (Sims, 1994).

5 different substrates are needed to generate these shapes:

- *Water* gives energy to cells by transformation ($Water \rightarrow (+30)$). This substrate diffuses in the environment.
- Four different *morphogen* substrates, here named *NW*, *NE*, *SW* and *SE*, show four division directions to cells. These substrates do not diffuse in the environment so as not to interact with the simulation. The designer of the creature positions them in the environment.

4 different actions are associated to these substrates:

- *duplication* consumes energy and one unit of *Water*,
- *water transformation* allows the cell to trigger a transformation of one substrate of *Water* into vital energy,
- *water absorption* allows the cell to pick up water from the environment,
- *apoptosis* allows the cell to autodestruct if it wishes (for example if the cell is not in the desired shape).

To obtain the required creature morphology, the genetic algorithm fitness is calculated after a chosen simulation time and is given by the next simple formula :

- if the cell is inside the desired shape, the fitness value is increased by 2 units,
- if the cell is outside the desired shape, the fitness value is decreased by 1 unit.

The first simple morphology we try to develop using this environment is a starfish[1]. To do that, we place morphogens in the environment to lead the cell divisions. Figure 7 gives the result of the genetic algorithm. We can observe that the desired shape is obtained. It is interesting to study the action selection system rules produced by the genetic algorithm:

$$\begin{aligned} (SensorNE = 1) &\rightarrow (DuplicateNE)\ (6) \\ (SensorNW = 1) &\rightarrow (DuplicateNW)\ (5) \\ (SensorSE = 1) &\rightarrow (DuplicateSE)\ (4) \end{aligned}$$

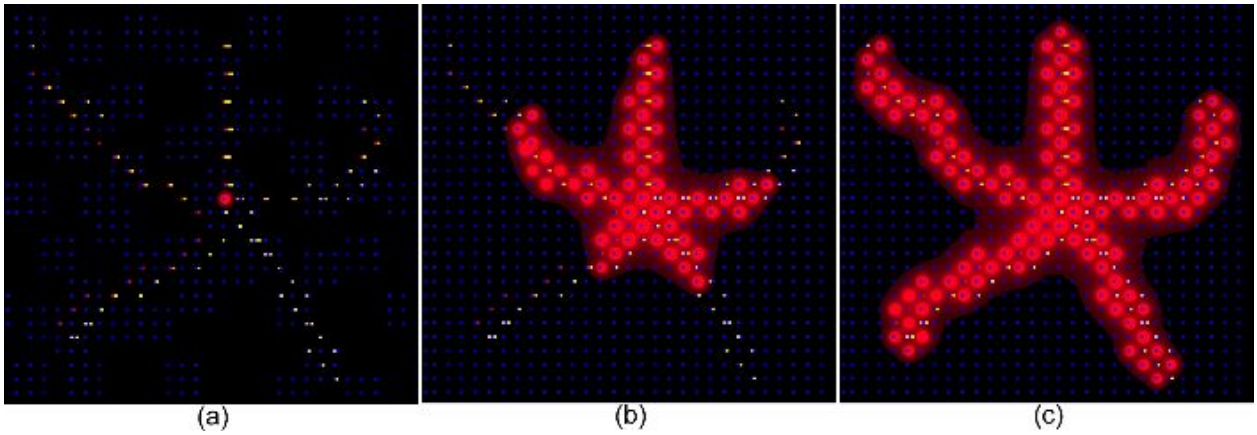


Figure 7: The starfish growth. (a) Beginning of the simulation. (b) The starfish develops itself following the morphogens. (c) The starfish stops its growth when the desired shape is obtained.

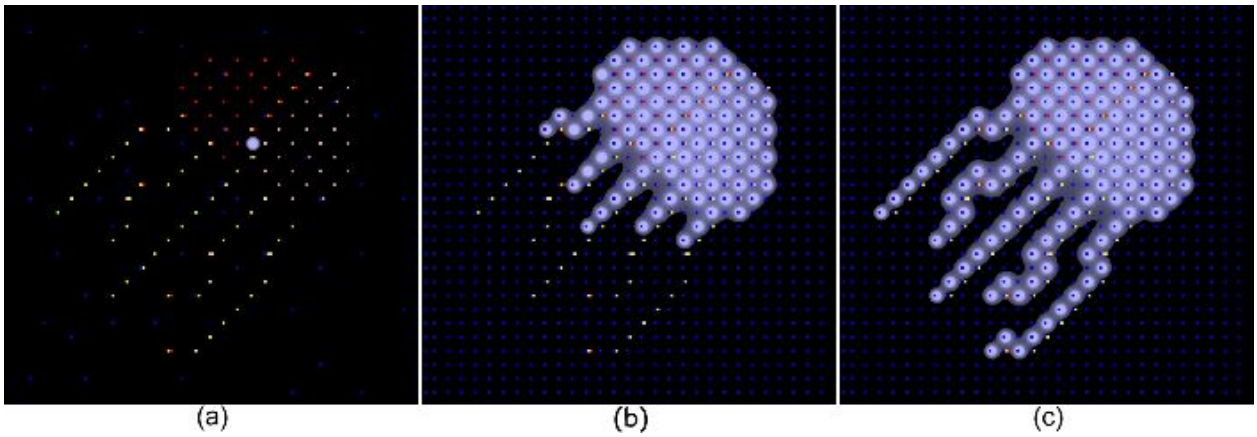


Figure 8: The jellyfish growth. (a) Beginning of the simulation. (b) The jellyfish develops itself following the morphogens. (c) The jellyfish stops its growth when the desired shape is obtained.

$$\begin{aligned} (SensorSW = 1) &\rightarrow (DuplicateSW)\ (3) \\ &\rightarrow (TransformWater)\ (2) \\ (SensorWater = 1) &\rightarrow (AborbWater)\ (1) \\ &\rightarrow (DoNothing)\ (0) \end{aligned}$$

This selection system shows that the genetic algorithm correctly uses the information given by the environment to follow the growth scheme given by the user. Moreover, duplications are always prior in relation to other actions to accumulate vital energy without using it. The last remark we can make about these rules is that the organism never uses *apoptosis* during growth. The organism assumes that morphogens give the correct growth direction.

Observing these rules, we notice that it could be possible to produce all desired creatures with the same genome. Indeed, the rules discovered by the organism allow it to follow any morphogen configuration. To verify the hypothesis, we decided to develop another simple creature: a jellyfish. To do that, we keep exactly the same environment architecture, with the same substrates and the same possible actions, and we only change the morphogen distribution in the environment. Using the starfish genome, we launch the simulation and we obtain the creature[1] shown by Figure 8.

## Conclusion and future works

We propose a model of cellular development. This model is based on a marked simplification of natural development. We ignore the physics rules and the atomic and molecular interactions to focus on the cell abilities. Using a genetic algorithm and specific environment, we create an organism able to develop different organs with different functions. As we have shown during experiments, this model can produce various creatures with very different morphology or different functions.

The continuation of this work presents a wide field of de-

velopment. Developing new organs can be interested. For example, the next one could be an organ able to harvest different substrates and transform them into vital energy and dispose wastes at a specific position. Using different types of such an organ, the wastes of one used as energetic substrate by another, we will produce a complete creature composed of different organs. The different organs will be connected using the presented transfer system.

Another improvement may concern shape generation. For the moment, we use four different morphogens to obtain the creature morphology. We think that with only one morphogen and only giving the development main line, we could obtain the same creature and have an organ that develops itself correctly to produce this morphogenetic substrate. For example, in the case of the starfish, we could have a transfer system that moves the morphogenetic substrate from the center of the environment to the five branches of the starfish. In a second stage, the starfish will grow using the morphogen distribution.

A remark we can make when we watch the starfish growth is that all the branches do not grow at the same speed. The same fact can be noticed in jellyfish growth, where the bell-shape grows too fast in comparison with tentacle development. An idea to control shape development is to calculate fitness at different moments of the simulation. The best creature will then be the one that produces the best shape at each checkpoint.

After few experimentations, the model also seems to be able of self-repairing (Miller, 2004). Killing some cells of the starfish in different parts (center, middle of an arm or a complete arm), the starfish create new cells in these wholes. This self-repairing property must be confirmed by more experiments but are encouraging.

A final development path is the abstraction of this model. Starting from a unique cell, we grow shapes like the starfish or the jellyfish presented in the paper and, after a cell regroupement to different limbs, we want to put the creature in a physical simulator to make it move. The creature movements could be generated, for example, by a neural network, just like in Sims' works (Sims, 1994). We hope that this abstraction will allow us to have a complete creature development, from single cell to a creature able to move in its environment.

**Acknowledgments** Experiments presented in this paper were carried out using ProActive, a middleware for parallel, distributed and multi-threaded computing (see http://proactive.inria.fr), and the Grid'5000 French experimental testbed (see https://www.grid5000.fr)

## References

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. *Genetic Programming Theory and Practice*, pages 43–62.

Chavoya, A. and Duthen, Y. (2007). Evolving an artificial regulatory network for 2d cell patterning. *Proceedings of the 2007 IEEE Symposium on Artificial Life*, pages 47–53.

Cussat-Blanc, S., Luga, H., and Duthen, Y. (2007). A developmental model to simulate natural evolution. *International Conference on Computer Graphics and Artificial Intelligence 3IA*.

Davidson, E. H. (2006). The regulatory genome: gene regulatory networks in development and evolution. *Academic Press*.

Dawkins, R. (1986). The blind watchmaker. *Longman Scientific & Technical*.

de Garis, H. (1999). Artificial embryology and cellular differentiation. In Peter J. Bentley, e., editor, *Evolutionary Design by Computers*, pages 281–295.

Dellaert, F. and Beer, R. (1994). Toward an evolvable model of development for autonomous agent synthesis. In *Artificial Life IV*, Cambridge, MA. MIT press.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries - a review. *Artificial Life*, 7(3):225–275.

Eggenberger Hotz, P. (2004). Asymmetric cell division and its integration with other developmental processes for artificial evolutionary systems. In *Artificial Life IX*, pages 387–392.

Garcia Carbajal, S., Moran, M. B., and Martinez, F. G. (2004). Evolgl: Life in a pond. *Artificial Life XI*, pages 75–80.

Gardner, M. (1970). The fantastic combinations of John Conway's new solitaire game life. *Scientific American*, 223:120–123.

Holland, J. H. and Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. *Pattern-Directed Inference Systems*.

Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artificial Life*, 13(1):11–30.

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theorical Biology*, 22:437–467.

Komosinski, M. and Ulatowski, S. (1999). Towards a simulation of a nature-like world creatures and evolution. In *ECAL '99*, pages 261–265, London, UK. Springer-Verlag.

Lassabe, N., Luga, H., and Duthen, Y. (2007). A New Step for Evolving Creatures. In *IEEE-ALife'07*, pages 243–251. IEEE.

Miller, J. F. (2004). Evolving a self-repairing, self-regulating, french flag organism. In *GECCO (1)*, pages 129–139.

Ono, N. and Ikegami, T. (1999). Model of self-replicating cell capable of self-maintenance. In *ECAL '99*, pages 399–406, London, UK. Springer-Verlag.

Rasmussen, S., Chen, L., Nilsson, M., and Abe, S. (2003). Bridging nonliving and living matter. *Artificial Life*, 9(3):269–316.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial Life IV*, pages 28–39.

Stewart, F., Taylor, T., and Konidaris, G. (2005). Metamorph: Experimenting with genetic regulatory networks for artificial development. In *ECAL '05*, pages 108–117.

Ventrella, J. (1998). Attractiveness vs efficiency (how mate preference affects location in the evolution of artificial ng organisms). *Artificial Life VI*, pages 178–186.