# Morphogen positioning by the means of a hydrodynamic engine

Sylvain Cussat-Blanc, Jonathan Pascalie, Hervé Luga and Yves Duthen

University of Toulouse - CNRS - UMR 5505
2 rue du Doyen-Gabriel-Marty
31042 Toulouse Cedex 9, France
{sylvain.cussat-blanc, jonathan.pascalie, herve.luga, yves.duthen}@irit.fr

**Abstract**

Artificial embryogeny aims at developing a complete organism starting from a unique cell. Nowadays many algorithms exist to synthesize artificial creature shapes or behaviours. With the purpose of shape and high-level behaviour joint evolution, one of the key aspects is the synthesis of positional information. Such pieces of information, called morphogens, are in many developmental models embedded in the environment and interactions are made through simple protein receptors. In this paper, we propose a new and original approach to solve the morphogen-positioning problem. We use a hydrodynamic model to replace the classical spreading algorithm. Mechanical constraints (the cell shape) and a dynamic activity are integrated. Thanks to this improvement, the cell behaviour can affect the spreading algorithm: cells can apply forces on the hydrodynamic environment to create substrate flows. Through experiments, this paper shows the way to develop complex shapes using this kind of simulator and proposes how to extend the simulation in a 3-D world in which physical laws are taken into account.

## Introduction

Literature offers many developmental models able to develop several kinds of creatures starting from a single cell (Stanley and Miikkulainen, 2003). Many goals motivate that kind of research work: to develop a particular shape, to evolve a high-level behaviour, etc. or, at a higher level, to understand living systems by the use of such models to simulate their mechanisms. Nowadays, a complete research field axis is about shape development from a single cell. One of the major problems of this work is morphogen positioning. Morphogens are often used as positional information to lead cells in their development. In nature, positional information is a key aspect in morphogenesis, embryogenesis, organogenesis and in behaviour synthesis at last. Evolvable mechanisms should be used in developmental models to spread their positional information in the environment. This could allow the emergence of a complex structure and/or behaviour. Keeping this goal in mind, we choose to embed morphogen positioning in cellular activity thanks to a hydrodynamic simulator which cells are able to interact with.

Our previous work proposed a developmental model, named *Cell2Organ* (Cussat-Blanc et al., 2008), based on a strong simplification of mechanisms used by living systems. The developmental model is a chemical simulator where organisms have to develop a metabolism, may have self-repairing capacities and have to perform user-defined functions. In this paper, we show the plug of a hydrodynamic engine with the developmental model in order to solve one of its main limitations: manual morphogen positioning. In comparison to a classical spreading algorithm, widely used in developmental models in literature, the use of a hydrodynamic engine allows more possibilities. Organisms will have the ability to create fluid flows, to move substrates or structures to organize the environment at their convenience. Gastrulation stage of vertebrate embryos can be simulated with this kind of system. In this early development stage, morphogens are positioned thanks to a physical invagination that induces many flows in the environment, as explained by some physicists' theories such as (Fleury, 2009).

In our bio inspired approach, the use of a hydrodynamic engine has sense looking at the early development stage. Gastrulation stage is seen as the first step of the morphogenetic process. During this step, high dynamic is observed in the embryo. Undifferentiated cells migrate and the egg membrane invaginates itself. Hydrodynamic forces are generated with a combination of these mechanisms. These forces are constraints for the different actors of the system. The consequence is the positioning of a kind of "mechanical gradients", in other words growth lines take place thanks to the created mechanical constraints. These developmental axis could be seen as an embryogenic pre-pattern. This latter is, as the example of vertebrates, four members positioned in pairs on the anterior and posterior zones of the organism.

This paper is organised as follows. Section 2 gives the related works on artificial development and morphogen positioning. Section 3 summarizes the model *Cell2Organ*. Section 4 details the hydrodynamic layer we add to the model in order to set up morphogens in the environment. Section 5 presents some results we obtain thanks to this new layer. We first develop simple shapes like diamonds or rectangles and a mushroom-shaped creature. We then develop more complex shapes. We conclude these experimentations by hav-

ing a discussion on the practicality of such a morphogenesis process to generate bigger creatures that could populate a 3-D world based on newtonian dynamics. Finally, we expose several options to improve this work.

## Related works

Over the past few years, more and more models concerning artificial development have been produced. A common method for developing digital organisms is to use Artificial Regulatory Networks (ARN). Banzhaf was one of the first to design such a model (Banzhaf, 2003). In his work, the beginning of each gene, before the coding itself, is marked by a starting pattern named "promoter". This promoter is composed of enhancer and inhibitor sites that allow the gene activations and inhibitions regulation. Another different approach is based on Random Boolean Networks (RBN) first presented by Kauffman (Kauffman, 1969) and re-used by Dellaert (Dellaert and Beer, 1994). An RBN is a network in which each node has a boolean state: activate or inactivate. The nodes are interconnected by boolean functions, represented by edges in the net. The cell function is determined during genome interpretation.

Several models dealing with shape generation have recently been designed (de Garis, 1999; Kumar and Bentley, 2003; Stewart et al., 2005; Chavoya and Duthen, 2008; Knabe et al., 2008; Joachimczak and Wróbel, 2009). Most of them use artificial regulatory network and morphogens to drive the development. With the latter approach, morphogens positioning in the environment is one of the main difficulties. In order to produce user-defined shapes as a French flag - that is one of the main benchmarks, a precise morphogen positioning is crucial. Two main methods exist to solve this problem: on the one hand, cells can produce morphogens by themselves that are spread in the environment with a simple spreading algorithm (Stewart et al., 2005; Knabe et al., 2008; Joachimczak and Wróbel, 2009) and, on the other hand, environment can contain built-in fixed morphogens (Chavoya and Duthen, 2008). Various shapes are produced, with or without cell differentiation. The well-known French flag problem was solved by Chavoya and Duthen, Knabe and recently in 3-D by Joachimczak. This problem shows the model differentiation capacity during multiple colour shifts.

Eggenberger was one of the first to propose a model that takes a leaf out of gastrulation (Hotz, 2003). In his work, both physics engine and artificial regulatory network (ARN) are used. The ARN controls cells behaviour whereas a physics engine allows to apply local constraints. Physical interactions could be observed between the cells and between the cells and the environment. Nevertheless, the substrate spread is made by cellular activity but is not influenced by the mechanical activity, that is to say movements made by cells do not spread any morphogen. Some biological theories about embryonic development bring out that hydrodynamic morphogen movements seem to be the basics of organogenesis (organ positioning the early embryo) and an explanation of most living being symmetric morphology (Cartwright et al., 2009; Fleury, 2009). To study the possible benefits of the morphogen flow creation in environments, we proposed to use a hydrodynamic layer whose activity is directly influenced by forces applied by cells.

This paper proposes a new morphogen positioning approach. More bio-inspired than biologically acceptable, we use a hydrodynamic engine to produce morphogen flows in the environment. Special cells have the ability to expulse morphogens with a given force whereas others will use the positional information to produce a defined shaped creature. Because our research axis is more focussed on creature development for virtual reality application than on cell mechanism realistic simulation, this bio-inspired approach is sufficient. Moreover, this kind of method could be used for future modular robots that could have the ability to expulse a substrate.

The next section presents our developmental model. It is based on action optimisation networks and on an action selection system inspired by classifier rule sets. It has been presented in details in (Cussat-Blanc et al., 2008).

## Summary of *Cell2Organ*

We choose to implement the environment as a 2-D toric grid. This choice allows a significant decrease in the simulation complexity keeping a sufficient degree of freedom thus reducing the simulation computation time.

The environment contains several kinds of substrates. They spread within the grid, minimizing the variation of substrate quantities between two neighbouring points. These substrates can spread on the grid at several speeds and can interact with other substrates. Interactions between substrates can be viewed as a great simplification of a chemical reaction: using different substrates, the transformation will create new substrates, emitting or consuming energy. Formally, this chemical reaction can be written as follows:

$$a_1 s_1 + a_2 s_2 + ... + a_n s_n \rightarrow a'_1 s'_1 + a'_2 s'_2 + ... + a'_m s'_m \quad (\delta_{energy})$$

where $s_i$ represents substrates, $a_i \in \mathbb{N}$ and $a'_j \in \mathbb{N}$ ($i \in 1..n$, $j \in 1..m$ are stoichiometric coefficients of the reaction and $\delta \in \mathbb{R}$ the quantity of energy produced (if positive) or consumed (if negative) during the reaction. For example, the reaction $2A + B \rightarrow C$ $(+50)$ produces one unit of C substrate from two units of A substrate and one of B's. The reaction also produces 50 units of energy.

To reduce the complexity, the environment contains a list of available substrate transformations. Only cells can trigger substrate transformations.

### Cells

Cells act in the environment, more precisely on the environment's spreading 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 an ARN is available inside the cell to allow specialization during division.

Each cell contains different density sensors positioned at each cell corner. Sensors allow the cell to measure the amounts of substrates available its Von Neumann neighbourhood. The list of available sensors and their position in the cell are described by the genetic code.

To interact with the environment, cells can perform different actions: perform a substrate transformation, absorb or reject substrates in the environment, divide (see later), wait, die, etc. This list is not exhaustive. The addition of an action is simplified by model implementation. As with sensors, not all actions are available for the cell: the genetic code will give the available action list.

Cells contain an action selection system. A system based on a set of rules is inspired by classifier systems. It uses data given by sensors to select the best action to perform. Each rule is composed of three parts: (1) The *precondition* describes when the action can be triggered. A list of substrate density intervals describes the neighbourhood in which action must be triggered. (2) The *action* gives the action that must be performed if the corresponding precondition is respected. (3) The *priority* allows the selection of only one action if more than one can be performed. The higher the coefficient, the more probable the rule selection.

*Division* is a particular action performable if the next three conditions are respected. First, the cell must have at least one free neighbour to create the new cell. Secondly, the cell must have enough vital energy to perform the division. The vital energy level needed is defined during the environment specification. Finally, during the environment modelling, a condition list can be added.

## Action optimisation

A new cell created after division is totally independent and interacts with the environment. During a division, the cell can optimize a group of actions. In nature, this specialisation seems to be mainly carried out by a gene regulatory network (GRN). In our model, we imagine a mechanism that plays the role of an artificial GRN. Each action has an efficiency coefficient that is linked to the action optimisation level: the higher the coefficient, the lower the vital energy cost. Moreover, if the coefficient is null, the action is not yet available for the cell. Finally, the sum of efficiency coefficients remains constant during the simulation. In other words, if an action is optimised by increasing its efficiency coefficient during a division, another (or a group of) efficiency coefficient has to be decreased. A network represents the transfer rule during a division stage. In this network, weighed nodes represent cell actions with their efficiency coefficients and weighed edges representing efficiency coefficient quantities that will be transferred during the division. Efficiency coefficient variations during division stage allow cell specialisation over divisions.

## Creature's genome

To find the best-adapted creature to a specific problem, we use a genetic algorithm. Each creature is tested in its environment. This latter returns the fitness at the end of the simulation. Each creature is coded with a genome composed of three different chromosomes: the list of *available actions*, an encoding of the *action selection* system and an encoding of the *optimisation network*.

Because of the complexity of developed creatures, the genetic algorithm had to be improved. First, we have decided to parallelise it on a computation grid. We used a middleware, named ProActive, that allows a total abstraction of grid infrastructure (Caromel et al., 2006). We applied a Master/Worker algorithm to parallelise our genetic algorithm. This algorithm is well suited to artificial evolution because the creature genome is small and the fitness computing cost is very important. Because of the small size of the genome, the network bottleneck induced by a Master/Worker architecture deployed on a computational grid will not heavily increase the computation time. Moreover, because the Master/Worker algorithm preserves the properties of a classical genetic algorithm, the number of generations needed by the algorithm to converge and the final solution quality are exactly the same with or without parallelisation.

A second optimisation of our genetic algorithm consists in leading the algorithm in its search. In our experimentation, the fitness function can be broken up with sub-objectives to describe the different evolution stages of the creature. This approach, commonly named *incremental evolution*, has been used in different domains such as behaviour simulation (Kodjabachian and Meyer, 1998; Mouret and Doncieux, 2008) or genetic programming (Walker, 2004). Authors generally conclude that global computation time is the same in comparison to a classical fitness but this algorithm gives more adapted solutions. In our problem, we generally break the fitness up in the three following stages: *metabolism* that is the lowest level function needed by the creature, *cell birth quantity* during the simulation shows the capacity of the organism to develop itself in the environment and *global fitness* that gives the efficiency of the organism to solve the problem (can also be broken up into sub-objectives).

## Example of generated creatures

Different creatures have been generated using this model. For example, we develop a *harvester*, a creature able to collect a maximum of substrate scattered all over the environment and to transform it into division material and waste. The creature has to reject the waste because of each cell limited substrate capacity. Another creature is the *transfer system*. Presented in (Cussat-Blanc et al., 2008), this crea-

ture is able to move substrate from one point to another. This creature is interesting because it has to alternate its behaviour between performing its function and developing its metabolism to survive. Finally, different *morphologies*, such as a starfish, a jellyfish or any user-designed shape, have been obtained (Cussat-Blanc et al., 2008). Once again, the organism must develop its metabolism to be able to sustain its activity.

All generated creatures have a common property: they are able to repair themselves in case of injury (Cussat-Blanc et al., 2009). This feature is an inherent property of the model. It shows the phenotype plasticity of produced creatures.

The last model's interesting feature is organ cooperation capacity to produce bigger structures. We have developed organs separately and built an organism composed of these organs that has a higher-level purpose. We create for example a self-feeding structure composed of four organs: two transfer systems and two producer-consumers.

Concerning the morphology development, one limitation of the model is the necessity to position morphogens by hand in the environment. In order to solve this problem, we propose a hydrodynamic layer that allows morphogen flow creation by cells. The organism has to make a morphogenetic blueprint of the shape in the environment before it develops itself by following division information. The next section details the hydrodynamic model we use and its set up options. The integration to the developmental model is also detailed.

## Hydrodynamic layer

This simulator manages hydrodynamic substrate interactions of our model. Its main aim is to propose a method inspired by the gastrulation of some living beings to position morphogens. This early stage of the organism development allows the morphogen positioning of the embryo in its immediat environment. It then allows the development of its organs. By the use of a hydrodynamic simulator in our model, we can produce the apparition of flows in the environment that correspond to flows created by the organism when it performs its actions (division, substrate absorption or rejection in particular). Thus, cells can for example expulse a substrate to be positioned in the environment in a specific direction and with a specific strength.

### Hydrodynamic model

Because of the computation cost induced by the hydrodynamic simulator complexity, we use a method that reduces the resource usage of the hydrodynamic layer on our simulation but keeps enough realism and degree of freedom. We base our work on Jos Stam's solver (Stam, 2003). This model is mainly used for image processing. This quite simple approach is interesting because its ability to solve Navier and Strokes' equations has been proved.

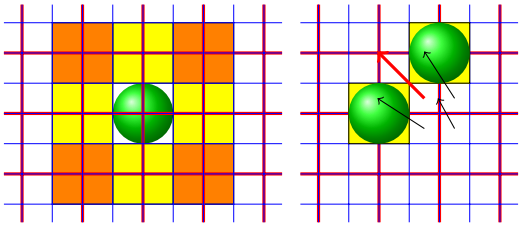



Figure 1: (a) Relative positioning of the chemical (red bold lines) and hydrodynamic (blue thin lines) environment. (b) Velocity vectors (red bold arrow) allow the spreading of few substrates on the other side of the cellular membrane.

In this model, the environment is a grid on which fluids particles are moving following speed vectors. Particles here represent our substrates. Our simulated cells are impassable obstacles. When a particle hits a cell membrane, the speed vector that corresponds to the collision point is modified in order to redirect the particle along the cell edge. In a first step, to simplify the simulation, all substrates will be spread separately, that is to say independently of one another. In other words, substrate flow interactions are not simulated with model. In our experimentation of morphogen positioning, this limitation has been overtaken bringing together all morphogens in a unique substrate and then breaking it up in the developmental model into several morphogens.

To ripen border conditions, the hydrodynamic simulator grid size has been doubled in comparison with the chemical simulator grid. Indeed, the smaller the grid subdivision, the more precise the border condition computation. In other words, fluid flows will be more precisely described. Because the grid subdivision strongly increases the computation cost, the hydrodynamic grid has only been subdivided by two in comparison to the chemical grid. The algorithm has also been adapted to take into consideration the inter-cell spreading allowed by our previous spreading algorithm. Because obstacles represented by cells are stuck together, no fluid flow is possible between cells. In our model, the organism's external speed vectors are able to modify the organism's internal speed vector in order to create internal flows. Figure 1 is a scheme of the subdivision grid and force applications in the environment.

The non-conservation material quantity is one of the main limitations of this model. Indeed, during the simulation, the hydrodynamic engine can generate a small loss of material. Such a loss could be unacceptable for the developmental model on little quantities or on application linked with real data such as real cell simulation. The main aim of the hydrodynamic engine is to spread morphogens in the environment in order to develop a shaped creature. Such a loss of material could generate a non-desired growth of the organism. However, several methods exist to fix the problem. The first

one consists in the implementation of an energy conservation law, which equilibrates the substrate leaks due to equation reductions. A proportional distribution of lost material on the entire grid has been preferred because the energy conservation method is expensive in computation resources and will be difficult to apply to our simulator.

The number of adjustable parameters is another strength of this model. Many properties are implied in fluid movements. The first parameter is the viscosity coefficient. This coefficient is used to describe the fluid movement. the higher the coefficient, the easier the outflow on its support. The second parameter of the model is the substrate density. This latter represents the capacity of the substrate to be spread during its spread. The higher the coefficient, the higher the links between substrates particles. Finally, the last parameter on which the user should act is the intensity of the force applied on the environment. The higher the force intensity, the bigger the induced activity.

The integration in our cellular simulation is simple: the hydrodynamic engine totally replaces the traditional spreading algorithm previously used to spread substrates. Cells interact with the environment, in particular by absorbing or rejecting substrates. Without a hydrodynamic layer, their actions could not create the fluid flows due to molecular movement. Now, the hydrodynamic engine can simulate this kind of phenomenon. Expulsion strength with a particular direction can be given to the cell. According to hydrodynamic forces, cells can position now a substrate everywhere in its environment. Cells can also create flows to produce global movement in the environment. Substrate absorption can create suctions in the same way. Lastly, as defined in the developmental model *Cell2Organ*, during a division stage, future cell position must be empty before the daughter cell creation. In other words, substrates in the mother cell neighbourhood must be spread in the close environment in order to clean up the space to the daughter cell. The addition of a hydrodynamic engine instead of a classical spreading algorithm induces the creation of multiple complex flows (vortex in particular) near the division that can modify the behaviour of close cells.

Preliminary results of such an engine use with our developmental model has been presented in (Cussat-Blanc et al., 2010). Through several experimentations, we showed the capacity of this kind of model to create hydrodynamic flows by using a cell that rejects substrates in a chosen direction. We also showed the possibility to lead the flow with the use of other cells, these latter acting as obstacles in the environment. Finally, we showed a possible extension of the model *Cell2Organ* in a physical world through the experimentation of a muscular joint.

In this paper, the previously presented hydrodynamic engine is used to position morphogens in the environment. A cell able to reject morphogens in the environment by giving them a defined force is used to create a pattern that an organism endowed with a shape generation genome will follow. Thanks to this method, we develop several shapes presented in the next section.

# Experiments

## Experimental conditions

To provide comparable results, the environment composition is the same in all next experiments. In order to develop several shaped creatures, several hydrodynamic engine parameters (viscosity, expulsion force and density) and initial cell possibilities are tested. We first present the used environment and cell capacities, which are always the same in next experimentations. The results of these experimentations are then presented.

The environment is composed of 5 substrates: energetic substrate $W$ that provides energy to cell by chemical reaction $W \rightarrow Energy$ (30), morphogen substrates $NE$, $NW$, $SE$, $SW$ that provide division information to cells. Whereas $W$ can spread and is massively present in the environment to develop an easy and efficient metabolism (the latter is not the main goal of the experiments), few morphogens are positioned in the environment to be only expulsed by cells.

Two kinds of cells are available in the environment.

***Pusher cells*** have two actions: reject morphogen in the environment and wait for a signal. Because the cells' genome is very simple, it is hand-coded: cells can reject morphogens while they have units into their membranes; when they have no more substrate, they wait indefinitely.

***Development cells*** can follow morphogens to develop a shaped-creature. The used genome has been evolved by a genetic algorithm and is detailed in (Cussat-Blanc et al., 2008). To summarize its functioning, cells have to manage their metabolisms provided by the energetic substrate $W$ and their development functions (follow morphogens to produce a shape). A good genome has been found by a genetic algorithm and can produce any desired shape if morphogens are correctly positioned in the environment.

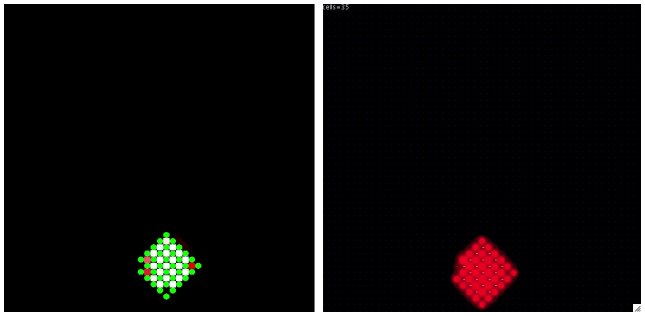
The rest of this section presents three experiments: simple shapes development, the development of a mushroom-like shaped creature and a four-armed creature. The aim is to study the impact of the hydrodynamic engine parameter modifications on the developed shapes. Videos of all these experiments are available on the website http://www.irit.fr/∼Sylvain.Cussat-Blanc.

## Simple shapes

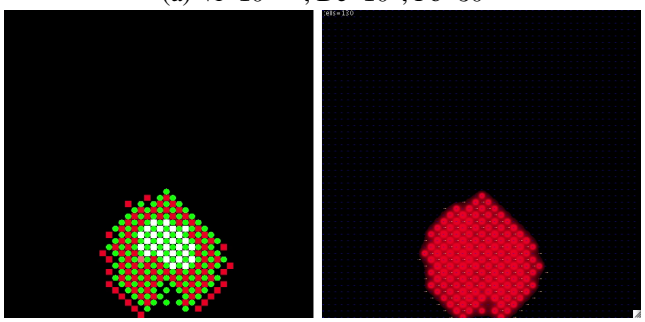
The aim of this first experiment is to give a range of possible shapes that can be produced by the model and to evaluate the

| Viscosity | Density | Force |
|---|---|---|
| $10^{-6} < Vi < 10^{-28}$ | $1 < De < 10^5$ | $30 < Fo < 50$ |

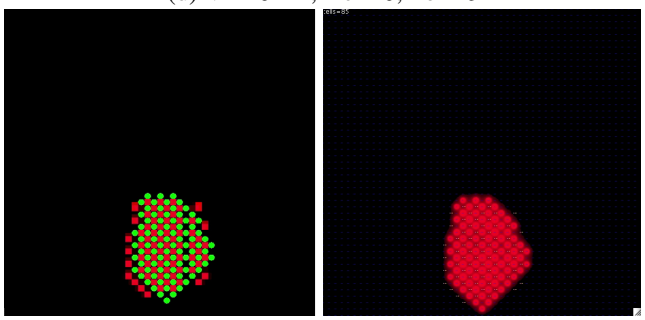
Table 1: Parameter acceptable value ranges



(a) Vi=$10^{-26}$, De=$10^{5}$, Fo=30



(d) Vi=$10^{-18}$, De=10, Fo=45



(c) Vi=$10^{-6}$, De=10, Fo=50

Figure 2: Influence of viscosity (Vi), density (De) and expulsion force (Fo) on developed shapes. On the left, hydrodynamic world where cells (in green) are obtacles and morphogen densities are represented with a gradient from white to red. On the right, the chemical world where cells (in red) are developping by following morphogens.

acceptable range of each parameter. In a first step, we empirically modify the parameters to develop as many shapes as possible. The parameter ranges are presented in table 1.

Figure 2 shows examples of produced shapes. As expected, parameter variations allow the development of different shape sizes (width) and statures (height). It is interesting to notice that figure 2(a) shows the capacity of the model to develop a square, a common problem of the literature (first step of the French flag problem). A high-density value ($De = 100000$) has been used here to keep morphogens grouped and make the production of such a shape possible.

With a low-density value, we develop the mushroom-shaped creature presented in figure 3. As previously introduced, the density parameter configures the stickiness force between substrates. The result is the development of a mushroom "cap" on the top of the shape, due to the vortex formation along the "stalk" that creates depressions. This accumulation produces two big vortexes of substrates on the top that produce the "cap".

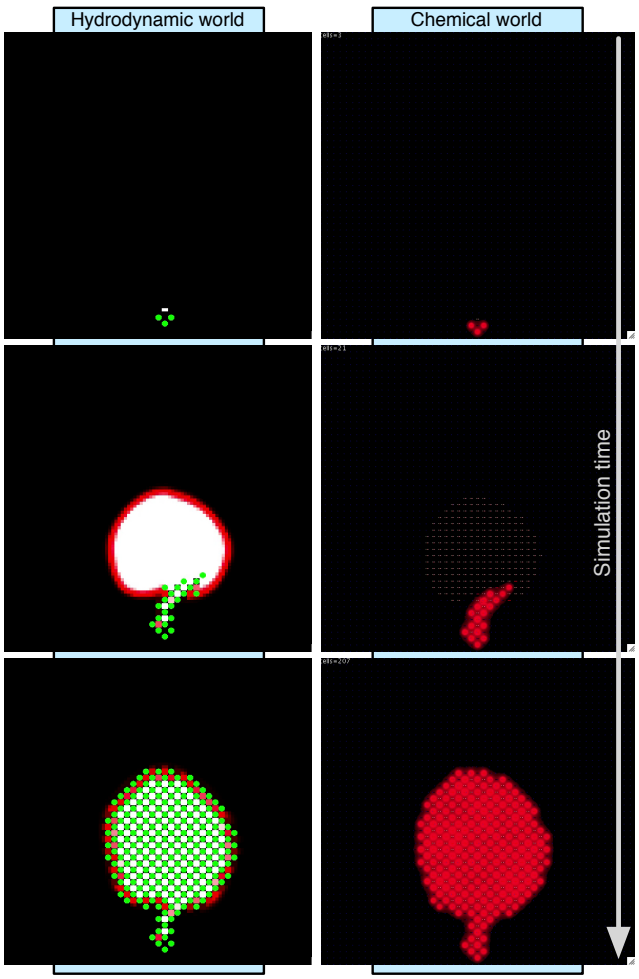
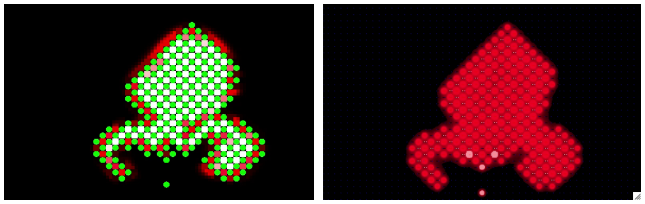


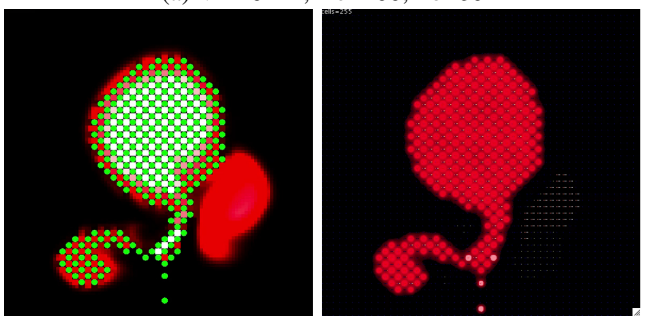

Figure 3: Development of a mushroom with morphogens positioning: a high fluid viscosity allows the cap formation.

## Cell configuration influence on morphogen flows

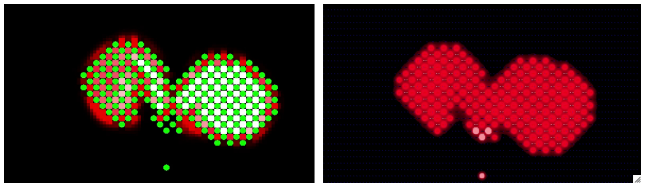
Modifying the initial cell configuration in the environment strongly influences the produced shape. Because cells are considered as obstacles in the hydrodynamic world, when a morphogen flow hits one of them, it is automatically divided in two flows that interfere. In these experiments, medium values of viscosity, density and expulsion forces are used. Depending on the cell position and the hydrodynamic engine parameters, many shapes can be obtained. Figure 4 presents some examples of initial configurations influences. Some interesting shapes appear in this figure: a kind of body endowed of tow tentacles in figure 4(a), an stomach-like shape



(a) Vi=$10^{-28}$, De=100, Fo=50



(b) Vi=$10^{-10}$, De=100, Fo=50



(c) Vi=$10^{-22}$, De=10, Fo=50

Figure 4: Influence of viscosity (Vi), density (De), expulsion force (Fo) and initial configuration on developed shapes (initial cells are highlighted in the chemical world).

on figure 4(b) and two wings on figure 4(c). This kind of shapes can be mixed to produce a complex creature and allow to jiggle in a simulated physical world. We will present an idea of such an improvement later in this paper.

## The four-armed creature

In order to produce a bigger creature that could move and act in a physical world, we develop a creature endowed with four arms. Based on the same environment as before, we modify the pusher cell to give it the possibility to expulse substrates in the four cardinal directions (up, down, left and right) in order to produce four morphogen flows in the environment. According to previous results, we choose the hydrodynamic parameters to produce rectangular sets of cells that will represent the arms. The initial configuration is also based on a simple shape development: a 4-direction pusher cell is set in the centre of the environment and four development cells are positioned on its diagonals, all around the pusher cell. Figure 5 shows the development of this four-armed creature.

Artificial creatures, with a morphology such as the four-armed creature previously presented, could be endowed with locomotive abilities in a simulated physical world. We already develop a physics engine that we plug in our model. This simulator, presented in details in (Cussat-Blanc et al., 2010), is linked to the chemical environment (*Cell2Organ*) and allows the simulation in a 3-D physical world of these developed organisms. We already showed the movement of a "muscular joint" where two "bones" rotate around a "kneecap" thanks to a "muscular fibre". All these components are produced by the developmental model and then linked in the physical world. Muscular fibre cells are able to change their shapes in order to produce a global movement. This kind of mechanism could be applied to the four-armed creature: each cell could be able to rotate around each other in order to produce a global movement of such a structure. With the intention of realising this behaviour, a high-level controller (neural network, classifier system, etc.) must be added to the cell to manage the rotation.

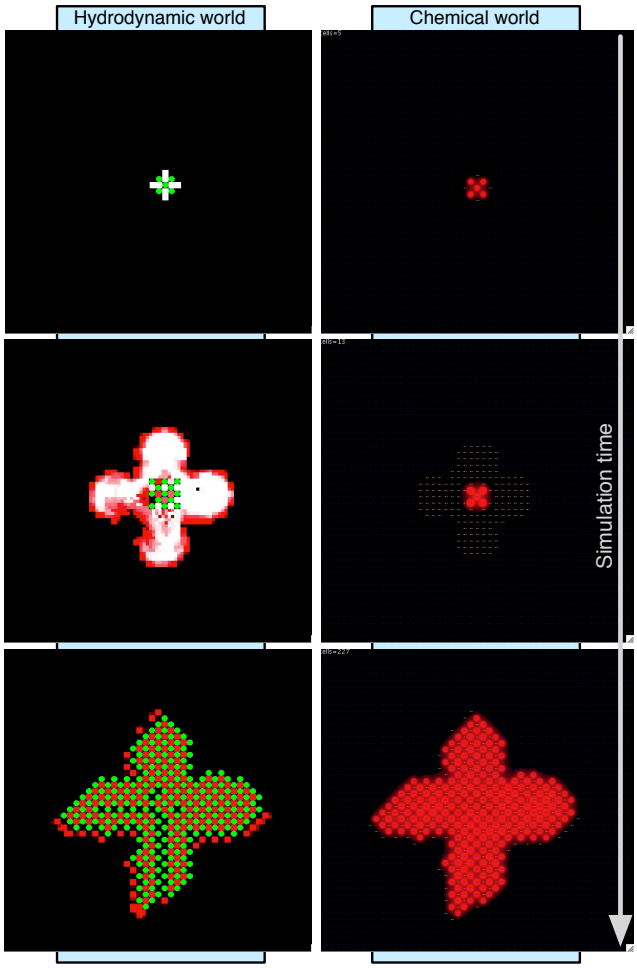



Figure 5: Development of a four-armed creature

## Conclusion

In this paper, we have presented the last features added to our developmental model. We have plugged a hydrodynamic en-

gine to automatically position morphogens in the environment. This first stage prepares the environment by positioning morphogens in the environment. A creature can then develop its morphology by following division information. Thanks to this add-on, we develop various shapes, simple or more complex. The hydrodynamic model we choose for a simulation allows us an interesting parameterisation of fluid properties: whereas most models are hard to tune, Stam's model allows a simple modification of viscosity, density and forces applied to substrates. We show that several morphologies can be obtained.

This work can be improved in many ways. First, it could be interesting to evolve the presented parameter set with an evolutionary algorithm. The use of such a research algorithm could help us to produce user-defined morphologies just by giving a fitness function that describes the shape of the creature (that is a common problem in literature).

To produce more complex creatures, we imagine a cell differentiation inspired from nature: in real living systems, after a given number of divisions, embryonic stem cells can produce differentiated cells (neurons, epithelial cells, etc.). The mechanism could be used in our model to produce rotations or morphology modifications in creatures: a pusher cell produces an initial morphogenetic pattern. Developing cells have a given division credit to produce a shape. When this credit is depleted, the developing cell turns into a pusher cell that produces a new morphogenetic pattern. Surrounding developing cells continue the shape development following the previously produced pattern and so on. A grammar based on L-Systems could give the division credit and pusher parameters (expulsion force and direction) and could be evolved by an evolutionary algorithm in order to produce complex creature morphologies.

Lastly, as presented at the end of the previous section, creatures must also be simulated in a 3-D physical world to produce high-level moves. This feature will bring us closer to our goal: producing a creature from a single cell able to move in a 3-D environment.

## References


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

Caromel, D., Delbe, C., di Costanzo, A., and Leyton, M. (2006). Proactive: an integrated platform for programming and running applications on grids and p2p systems. *Computational Methods in Science and Technology*, 12.

Cartwright, J., Piro, O., and Tuval, I. (2009). Fluid dynamics in developmental biology: moving fluids that shape ontogeny. *Frontiers of Interdisciplinary Research in the Life Sciences*.

Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.

Cussat-Blanc, S., Luga, H., and Duthen, Y. (2008). From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141. MIT Press, Cambridge, MA.

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

Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010). Three simulators for growing artificial creatures. In *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC 2010), to be published*.

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.

Fleury, V. (2009). Clarifying tetrapod embryogenesis, a physicist's point of view. *The European Physical Journal Applied Physics*, 45(3):30101–30101.

Hotz, P. (2003). Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1.

Joachimczak, M. and Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *10th European Conference on Artificial Life (ECAL09)*. Springer Verlag.

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

Knabe, J., Schilstra, M., and Nehaniv, C. (2008). Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. *Artificial Life XI*, 11:321.

Kodjabachian, J. and Meyer, J. (1998). Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812.

Kumar, S. and Bentley, P. (2003). Biologically inspired evolutionary development. *Lecture notes in computer science*.

Mouret, J. and Doncieux, S. (2008). Incremental Evolution of Animats' Behaviors as a Multi-objective Optimization. *Lecture Notes in Computer Science*, 5040:210–219.

Stam, J. (2003). Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, volume 18.

Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

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

Walker, M. (2004). Comparing the performance of incremental evolution to direct evolution. In *Second International Conference on Autonomous Robots and Agents*, pages 119–124.