Programmable Architectures That Are Complex and Self-Organized: From Morphogenesis to Engineering

René Doursat[1]

[1]Institut des Systèmes Complexes, CREA, CNRS & Ecole Polytechnique, Paris, France
doursat@shs.polytechnique.fr

Abstract

Outside biological and social systems, natural pattern formation is essentially "simple" and random, whereas complicated structures are the product of human design. So far, the only self-organized (undesigned) *and* complex morphologies that we know are biological organisms and some agent societies. Can we export their principles of decentralization, self-repair and evolution to our machines, networks and other artificial constructions? In particular, can an "embryomorphic" engineering approach inspired by evo-devo solve the paradoxical challenge of planning autonomous systems? In this work, I wish to better understand and reproduce complex morphogenesis by investigating and combining its three fundamental ingredients: *self-assembly* and *pattern formation* under *genetic regulation*. The model I propose can be equivalently construed as (a) *moving* cellular automata, in which cell rearrangement is influenced by the pattern they form, or (b) *heterogeneous* collective motion, in which swarm agents differentiate into patterns according to their location. It offers a theoretical framework for exploring the causal and programmable link from genotype to phenotype.

Introduction

Faced with a rapid growth in size and complexity of computer systems, whether hardware, software or networks, engineers are gradually led to rethink ICT in terms of *complex systems*. In particular, as the field of Artificial Life demonstrates, it is compelling and fruitful to seek inspiration from biological and social examples such as organism development, neural networks, insect colonies, or human communities. Understanding natural emergence should help design a new generation of artificial complex systems by importing into our machines highly desirable properties that are still largely absent from traditional engineering: *decentralization*, *autonomy* (self-organization, homeostasis) and *adaptation* (learning, evolution). Simply formulated, the new challenge is: How can we make a multitude of agents get together and do something useful without placing them by hand? Emergent engineering will be less about direct design than developmental and evolutionary *meta-design*. Changing from micro-managers to law-makers, future engineers would "step back" from their creation and only set generic conditions for systems to self-assemble and evolve, instead of building them directly.

Darwinian evolution consists of random variation followed by non-random selection. Concerning evolutionary engineering, the present work stresses the importance of establishing fundamental laws of *developmental variations* before these can be selected on the evolutionary time scale [20]. Understanding variation by comparing the development of different species is the concern of "evo-devo", a fast growing field of biology [4, 12]. The genotype-phenotype link cannot remain an abstraction if we want to unravel the generative laws of development and evolution—and ultimately transfer them to artificial self-organized systems. Moreover, fine-grain, hyperdistributed architectures (i.e., many light-weight agents, as opposed to a few heavy-weight agents) such as multicellular organisms might be in a unique position to provide the "solution-rich" space needed for successful selection.

Within this framework, the goal of this article is to understand and model the *self-organization of complex morphologies*. To this aim, it proposes to combine three ingredients: morphogenetic *self-assembly* (SA) and *pattern formation* (PF) under the control of *non-random, structured genetic regulation* (GR) stored inside each agent of a swarm.

Toward Self-Organized Complex Architectures

Non-biological/social self-organization exhibits "simple" patterns. Self-organized systems of physical-chemical matter generally form random, repetitive spatial patterns: ripples in sand dunes, convection cells in hot liquids, spots and stripes in reaction-diffusion solutions à la Turing [1], etc. Despite a huge and fascinating diversity of pattern formation behaviors across many scales and substrates, emergent structures at the macroscopic level are fairly regular, essentially consisting of repeated motifs. They display a statistical uniformity and relative "poorness of information" similar to textures. Moreover, most of these pattern formation phenomena rely on instabilities and amplification of fluctuations to generate order. Because of this inherent stochasticity, the number and position of emerging entities (spots, stripes, etc.) are generally unpredictable. The only self-organized systems able to create truly complex structures are biological organisms and agent societies (e.g., termite mounds, cities, markets, Internet).

Non-biological/social complex structures are deliberately designed. Outside biology and agent societies, most complicated structures made of segments and parts arranged in specific ways are the product of direct human control: computers, cars, buildings, etc. Contrary to physical pattern formation systems, human constructions are fundamentally reproducible and programmable. They are made of a diversity of modules that are statistically heterogeneous and information-rich.

However, the cost of such complexity is *heteronomy*: these structures rely entirely on centralized design and deterministic planning at the macroscopic level, imposing order from the outside. Again, the only complex forms that are also truly undesigned, i.e., naturally emergent, are biological and social.

Re-creating structures that are complex *and* self-organized. Compared to physical pattern formation, the unique feature of biological and social morphogenesis is that it relies on *agents* (cells, insects, computers, humans) that carry *sophisticated instruction sets* (DNA, stigmergy, program, cognition). This functional information endows the agents with a repertoire of non-trivial behaviors, vastly superior to units of inert matter. Most importantly, it opens the door to agent *diversity* through differentiation and evolution, which in turn allows rich combinations and recombinations of agents into modules and hierarchical constructions. Therefore, focusing for now on multicellular organisms, can we strive toward a new kind of morphogenesis-inspired or "embryomorphic" engineering? It is the purpose of this work to show how genetic-like regulation at the agent level can be used to control an artificial process of complex self-organization.

Integrating Self-Assembly and Pattern Formation Under Agent-Level Genetic Regulation

In this modeling work, I propose that, from an abstract viewpoint, self-organized complex morphologies such as biological development can be best understood as a combination of *self-assembly* (SA) and *pattern formation* (PF). To take an artistic metaphor, this would be similar to mixing "self-sculpting" and "self-painting" in one composition [6]. On the one hand, embryogenesis can be seen as a "self-made puzzle", i.e., a spontaneous sculpting process in which the puzzle pieces (the cells) reshape and reassemble themselves dynamically. On the other hand, it can also be seen as a "deformable screen", i.e., a spontaneous painting process where color strokes (gene expression levels) modify each other on top of an irregular and shifting geometry.

Self-assembly (SA). Research in natural or artificial *self-assembling* systems, mostly following "molecular soup" models, has traditionally focused on pre-existing components endowed with fixed shapes. Biological development, by contrast, dynamically creates new cells that acquire selective adhesion properties through differentiation induced by their neighborhood. I propose here a model of self-organized swarm in which the agents undergo dynamical *positioning* from neighbor forces, dynamical *creation* by division, and dynamical *reshaping* by non-uniform modification of their interactions. These elementary SA behaviors are induced and controlled by agent differentiation (see PF in next subsection).

Pattern formation (PF). Pattern formation phenomena are generally construed as orderly states of activity on top of a continuous 2-D or 3-D substrate. Yet, again, the spontaneous patterning of an organism into regions of gene expression arises within a multicellular medium in perpetual expansion and reshaping. In the present model, agents undergo dynamical *differentiation* into various types and subtypes. The swarm becomes inherently *heterogeneous*, breaking up into local groups. PF activity is based on the exchange of two categories of signals within and among these groups: *positional information* (spread of gradients or signalling "counters") and *identity information* (gene-expression levels). These elementary PF behaviors are induced and controlled by agent positions (see SA in previous subsection).

Genetic regulation (GR). Finally, traditional SA and PF are often thought of in terms of *stochastic* events, i.e., collisions and fluctuations. By constrast, biological cells are not randomly mixed but pre-positioned where divisions occur (before migrating). Genetic identity regions are not randomly distributed but highly regulated in number and position. They dynamically *unfold in time*, on the basis of simple calculations and decisions carried out by each agent at every time step. Agents contain a complete *genotype G*, of which they execute only a small portion at any time, depending on their current differentiation type and input from their neighborhood.

From Biology to Engineering

This study is inherently interdisciplinary, as it closely follows biological principles at an abstract level, but does not attempt to model detailed data from real genomes or organisms. Thus, it lies at crossroads between different families of works, from developmental and systems biology to artificial life, in particular spatial computing, evolutionary programming and swarm robotics. It is an original attempt to integrate the three mechanisms of SA, PF and GR discussed above. Only few previous theoretical models of biological development or bio-inspired artificial life systems have combined them in various ways. The evo-devo works of [11, 17], or [19, 16] with lesser morphogenetic abilities, are among these notable achievements. Other interesting studies have explored the combination of two out of three: SA and PF, no GR—self-assembly based on cell adhesion and signalling pattern formation, but using only predefined cell types without internal genetic variables (e.g., [15]); PF and GR, no SA—non-trivial pattern formation from instruction-driven intercellular signalling, but on a fixed lattice without self-assembling motion (e.g., [7]); SA and GR, no PF—heterogeneous swarms of genetically programmed, self-assembling particles, but in empty space without mutual differentiation signals (e.g., [18]).

Model

This section presents a computational model, with illustrative numerical simulations, of *programmable* and *reproducible* artificial morphogenesis. The differential properties of cells (adhesion, division) are determined by the regions of gene expression to which they belong, while at the same time these regions further expand and segment into subregions due to the self-assembly of differentiating cells. The model can be construed from two different vantage points: either (a) pattern formation on *moving* cellular automata, in which the cells spatially rearrange under the influence of their activity pattern, or (b) collective motion in a *heterogeneous* swarm, in which the agents gradually differentiate and modify their interactions according to their positions and the regions they form.

First, the motion of a homogeneous swarm (pure SA) and the patterning by gradient propagation on a fixed swarm (pure PF) are introduced separately. Then, these two components are combined to form reproducible growing patterns

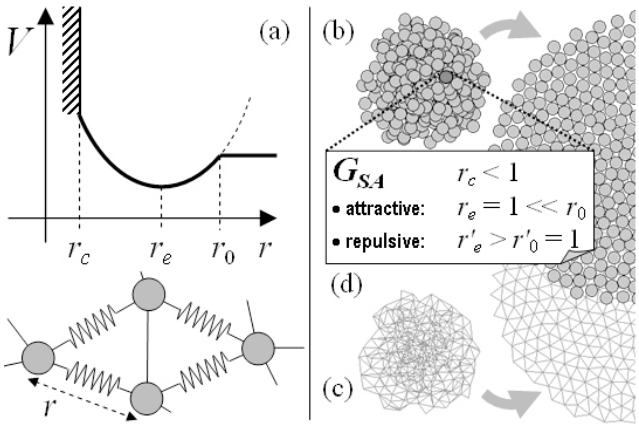

Figure 1: Deployment of a homogeneous swarm (SA; see text). (a) Agent-level interaction potential V similar to elastic springs. (b) Relaxation of a 400-agent swarm from an initially compressed state (only half of the end state shown). (c) Same swarm without nodes, showing interaction mesh obtained by Delaunay triangulation and pruning of edges longer than r_0. (d) Genetic SA parameters inside every agent (here, attractive mode only).

(SA + PF). The genetic program controlling these arrangements inside every agent is also explained. Finally, this combination is repeated as modules ($SA^{(k)} + PF^{(k)}$) inside a larger, heterogeneous system to create complex morphologies by recursive refinement of details. All swarm formations presented in the figures result from actual simulations (in Java).

Deployment of a Homogeneous Swarm (SA)

Exploring the principles of multicellular development as an inspiration for self-organized artificial systems, the model incorporates two major aspects of cellular biomechanics: cell *adhesion*, in the form of elastic rearrangement, and cell *division* (addressed in a later subsection). Schematically, a self-assembling swarm is composed of agents or "puzzle pieces" described by their *geometrical variables*, *motion dynamics* and *interaction network*. In 2-D, each agent A has a position $\mathbf{r}_A = (x_A, y_A)$, velocity $d\mathbf{r}_A/dt$ and shape at a certain orientation. In this model of swarm dynamics, agent shapes actually represent mutual adhesion affinities implemented by local *interaction potentials* $V(\mathbf{r}_A, \mathbf{r}_B)$ around the agents. Thus, swarm motion is caused by agent-centered forces derived from V. Here, simple discs of diameter r_c are used, creating isotropic potentials $V(\| \mathbf{r}_A - \mathbf{r}_B \|) = V(r_{AB})$ in their vicinity. Similar to other collective motion models [21], $V(r)$ consists of three parts (Fig. 1a): (i) infinite repulsion for $r < r_c$ representing non-deformable particles, (ii) elastic (quadratic) attraction around an equilibrium distance r_e representing the resting length of small springs, and (iii) flat potential for $r > r_0$ representing the absence of force beyond a certain "visibility" horizon. Agents interact through a dynamic network topology that depends on their positions. Edges $A \rightarrow B$ are created and removed according to a given connectivity scheme, e.g., circular scope ($r_{AB} < r_0$), k-nearest neighbors, Delaunay triangulation, or a combination thereof. For low values of r_0 and $k = 6$, these schemes are roughly equivalent. Starting from a compressed swarm, agents quickly relax to a resting state, in which they tend to form quasi-regular triangular meshes (Fig. 1c). Experiments in the rest of this paper are based on the Delaunay triangulation with additional pruning for $r > r_0$.

Thus, at this stage, each agent in the swarm possesses (a) fixed "genetic" SA parameters, denoted by G_{SA} (Fig. 1d), and (b) dynamic SA state variables—its position and connections with other agents. The genetic parameters consist of V's parameters r_c, r_e and r_0. Typically, $r_c < r_e = 1 << r_0$ for attractive potentials, but V can also become neutral or repelling if $r_0 \le r_e$. Repulsion will be later used between different types of agents (see last subsection about modular development).

Propagation of Positional Information (PF-I)

Pieces of a jigsaw puzzle are defined not only by their position and shape but also by the "image" they carry. In the self-organized swarm, this translates into state variables inside each agent that determine their PF activity. The present model distinguishes between two kinds of PF-specific state variables: *gradient* variables (PF-I) and *pattern* variables (PF-II), addressed in the next subsection. Gradient values propagate from neighbor to neighbor and establish *positional information* across the swarm [23]. Pattern values are calculated from the gradient values and create different *agent types*, which in turn affect the SA behavior (see SA + PF integration below).

Thus, agents not only interact mechanically according to the SA forces, but also exchange activity signals on the same graph edges according to PF rules. Halting the SA dynamics for now, let us consider a fixed swarm such as the one produced by Fig. 1b. Assume that one agent denoted by W contains a counter variable $n_W = 0$ and passes messages to its neighbors, instructing them to set their own n_W to 1. These neighbors in turn instruct their neighbors to set n_W to 2, and so on. To avoid back-propagation effects, the actual value of n_W remains the minimum of all received values. The result is a roughly circular wave pattern centered on W (Fig. 2a), which represents a discrete approximation of a heat-like diffusive gradient in continuous space. Discrete counter increments are also the method of choice for spreading positional informa-

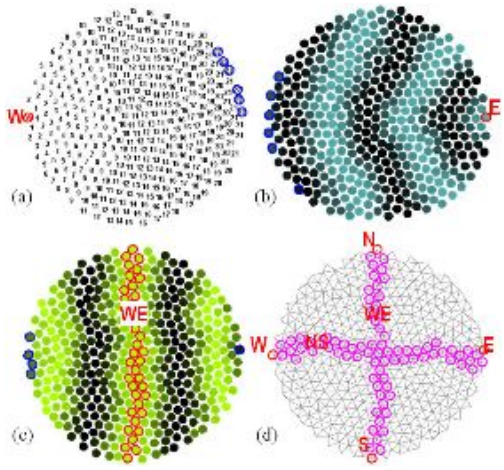

Figure 2: Propagation of positional information (PF-I; see text). (a) Circular gradient of counters originating from source agent W in red (gradient ends in blue). (b) Opposite gradient coming from antipode agent E viewed by a cyclic color map. (c) Planar gradient triggered by agents WE, whose W and E counters are equal ±1. (d) Full coordinate compass on mesh, with midlines.

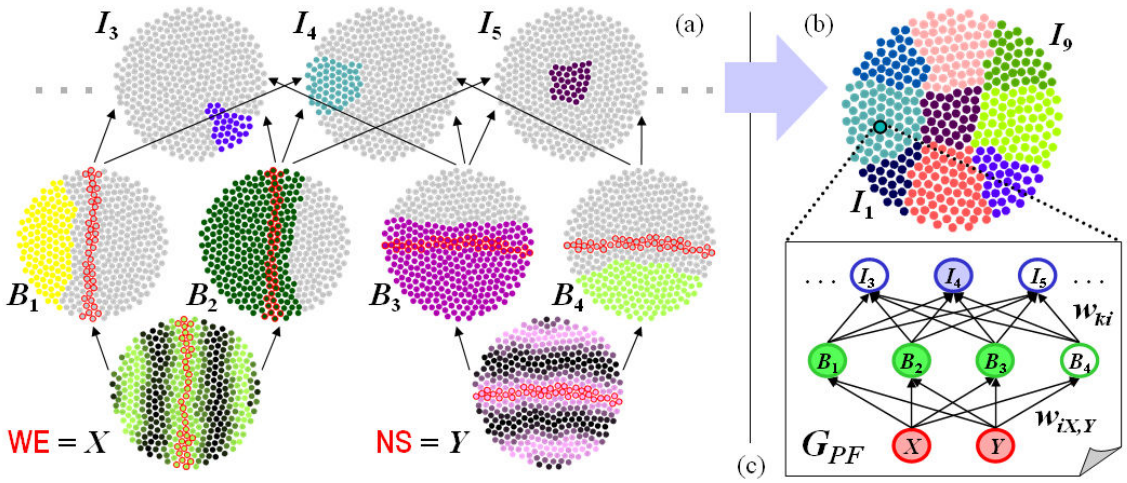

Figure 3: Programmed patterning (PF-II; see text). (a) The same swarm in different colormaps to visualize the agents' internal patterning variables X, Y, B_i and I_k (virtual equivalent of *in situ* hybridization in biology). (b) Consolidated view of all identity regions I_k for $k = 1...9$. (c) Gene regulatory network used by each agent to calculate its expression levels, here: $B_1 = \sigma(1/3 - X)$, $B_3 = \sigma(2/3 - Y)$, $I_4 = B_1B_3(1 - B_4)$, etc.

tion in amorphous and spatial computing systems [7, 16, 2]. In the present model, the role of source W can be transferred to another agent, thereby shifting the entire gradient landscape in successive corrective waves, as agents continually communicate with each other to adjust their counters.

In parallel to W, assume that another gradient propagates from a source agent E located at a certain distance from W, e.g., at two antipodes of the swarm (Fig. 2a,b). All agents have now two counters, n_W and n_E. In the example of Fig. 2, $n_W = 0$ and $n_E = 22$ in W and conversely in E. Together, these two gradients define a midline across the swarm, denoted by WE. It is the subset of agents that are equidistant from W and E, i.e., $n_W \approx n_E$, for example $|n_W - n_E| \leq 1$ (see, e.g., [16]). Agents belonging to the WE midline become in turn the sources of a new gradient, creating a planar wave of n_{WE} counters that propagates symmetrically toward W and E (Fig. 2c; $n_{WE} = 0$ in WE, and 11 in W or E). Finally, assume that two other gradient sources, N and S, are located at two other antipodes of the swarm on the WE midline. This creates a second midline NS perpendicular to WE and a second planar wave of n_{NS} counters (Fig. 2d). Each agent now has 6 counters: n_W, n_E, n_{WE}, n_N, n_S and n_{NS}. Together, they establish a 2-D *pattern coordinate system* (X, Y) in the swarm—distinct from the physical coordinates (x, y) of the SA process—for example by setting: $X = \text{sign}(n_W - n_E)n_{WE}$, and $Y = \text{sign}(n_S - n_N)n_{NS}$. To obtain normalized coordinates, each agent can also divide X and Y by local estimates of the global width w and height h of the swarm: $X' = X/w$, where $w = \max_A(n_{WE})/2 \approx n_{WE} + n_W$ for $X < 0$ and $w \approx n_{WE} + n_E$ for $X > 0$—and similarly for the vertical axis, replacing X, W, E and w by Y, S, N and h.

Naturally, the polar and equatorial locations of the four sources N, S, W and E are not imposed by hand, but are themselves the result of a self-organizing process via a feedback loop between gradients and sources. This is explained below in the subsection about SA + PF integration.

Programmed Patterning (PF-II)

On top of the coordinate system created by the gradient variables, each agent calculates another set of variables that are responsible for the swarm's patterning or "image". This process represents the emergence of *heterogeneity*, i.e., the segmentation of the swarm into *different types* of agents. In principle, any arbitrary pattern I (at the level of resolution offered by the swarm) could be programmed into the agents as a direct function of the gradient coordinates $I(X, Y)$. However, for reasons explained below (see modular patterning), it is preferable to proceed stepwise and let the swarm build itself in a *modular* fashion. The present model uses elementary patterns such as stripes and checkerboards. Naturally, unlike Turing patterns, each region is controlled here by a different gene set.

A biological embryo is a swarm of cells, where each cell contains a *gene regulatory network* (GRN) coding for its signalling and mechanic behavior. Through intercellular coupling between neighboring GRNs, the embryo becomes patterned into identity regions of differentiated gene expression, creating a "hidden geography" revealed by *in situ* hybridization. Essentially, logical combinations of regulatory switches ('or', 'and') translate geometric combinations of precursor patterns into new patterns (by union and intersection). Developmental genes are roughly organized in tiers or "generations". Earlier genes map the way for later genes, and gene expression propagates in a cascade. This principle has been beautifully demonstrated in the *Drosophila* embryo (see [4]). The intersection of various striping patterns along its three main axes gives rise to smaller regions such as the organ primordia and "imaginal discs," which are groups of cells marking the location and identity of the fly's future appendages (legs, wings, antennae). Going back in time, the whole process begins with concentration gradients of maternal proteins that diffuse across the initial cluster of cells and create the functional equivalent of a coordinate system, in a way similar to the PF-I process described in the previous subsection.

The early striping process of *Drosophila* is controlled by a regulatory hierarchy containing five main tiers of regulatory genes [4]. The present model relies on a three-tier caricature of the same idea, the *positional-boundary-identity* gene network [8, 9], which represents the genetic parameters of the PF process and is denoted by G_{PF} (Fig. 3c). In each cell-agent of our 2-D virtual embryo-swarms, the bottom layer of G_{PF} con-

tains the two positional variables X and Y seen previously; the middle layer, n "boundary" nodes $\{B_i\}_{i=1...n}$; and the top layer, m identity nodes $\{I_k\}_{k=1...m}$. Variables X, Y, B_i and I_k denote the gene expression levels or "activity" of the nodes. The boundary nodes compute linear discriminant functions of the positional nodes: $B_i = \sigma(w_{ix}X + w_{iy}Y - \theta_i)$, where $\{w_{ix}, w_{iy}\}_{i=1...n}$ are the regulatory weights from X and Y to B_i, parameter θ_i is B_i's threshold and sigmoid function $\sigma(u) = 1/(1 + e^{-\lambda u})$. The effect of a boundary node is to segment the embryo's plane into half-planes of strong and weak expression levels, 1 and 0 (Fig. 3a, middle row). Finally, the identity gene levels are given by logical combinations of the near-binary boundary gene values, for example, by calculating the products $I_k = \prod_i |w'_{ki}|(w'_{ki}B_i + (1-w'_{ki})/2)$, where $w'_{ki} \in \{-1, 0, +1\}$ represent ternary weights from B_i to I_k. This means that the i-th factor inside I_k can take three possible values: $(1 - B_i)$, 0 or B_i.

With this type of gene regulatory network, the "identity regions", i.e., the regions of high I expression, take the form of polygons at the intersection between several boundary lines (Fig. 3a, top row). When viewed together, they create a checkered pattern (Fig. 3b). These different colored regions represent different agent types and will be the starting point of new local SA and PF processes (see below). At this stage, similar to SA, each agent in the swarm also possesses (a) fixed "genetic" PF parameters in G_{PF} and (b) dynamic PF state variables—the gradient values n and the activity of G_{PF}'s nodes.

Simultaneous Growth and Patterning (SA + PF)

After describing the self-assembly of a non-patterned swarm and the patterning of a fixed swarm, SA and PF are now combined to create growing patterns (Fig. 4). Agents continually adjust their positions according to the elastic SA constraints, *while* continually exchanging gradient values and PF signals over the same dynamic links. This dual dynamics is guided by both genotypes G_{SA} and G_{PF} (Fig. 4d). Another mechanism, *cell division*, is also introduced at this point. Any agent A may divide with probability p at every time step and produce a new agent B, which is initially positioned a small distance from A with a random angle (Fig. 4c). Then the position of B and its neighbors rearrange under potential V as usual. Agent B inherits all of A's attributes, including genotype G_{SA+PF} and internal PF variables. It immediately starts contributing to the traffic of PF gradients that maintain the pattern's consistency at all times in the swarm.

From the SA point of view, a dividing swarm starting from few agents reliably grows through successive round shapes (Fig. 4a,c). In Fig. 1, the number of agents was constant and the expansion of the swarm was only due to elastic relaxation. In Fig. 4, agents are perpetually added while the swarm remains approximately in mechanical equilibrium at all times. From the PF point of view, the pattern is also maintained at all times by the continual propagation and readjustment of the gradients but also by the continual self-positioning of the four source agents N, S, W and E. To achieve a well-deployed compass as the one of Fig. 4b, source *migration* rules are added. Each agent contains four binary variables or "source flags" s_W, s_E, s_N and s_S, which are 0 almost everywhere and 1 in one of the four sources. According to the first migration rule, the W source must then always transfer value 1 of the s_W flag to a neighbor that has a greater n_E count than itself, and vice versa for E (same between N and S). This makes labels W and E move away from each other, hopping from agent to agent. The second migration rule stipulates that the W and E agents must also seek to minimize $|\, n_S - n_N \,|$, i.e., hop toward the NS midline (and symmetrically for N and S toward WE).

Modular, Recursive Patterning (PF[k])

Embryological patterns do not develop in one shot but in numerous incremental stages [6]. An adult organism is produced through gradual morphological refinement, following a cascade of genetic regulation from precursor developmental genes to secondary genes, tertiary genes, and so on. Importing this critical feature into the present model, the above gene network G_{PF} is extended to include a pyramidal *hierarchy* of network modules (Fig. 5g) able to generate patterns in a recursive fashion. First, the base network G_{PF} establishes main identity regions as before (Fig. 5a). Then a few subnetworks $G_{PF}^{(k)}$ further partition these regions into smaller identity compartments at a finer scale (Fig. 5e,f). The execution of $G_{PF}^{(k)}$ is triggered by the activity of node I_k in G_{PF}. This means that all agents with a high value of I_k start trading new local gradient counters $n_W^{(k)} \ldots n_S^{(k)}$, $n_{WE}^{(k)}$ and $n_{NS}^{(k)}$ (Fig. 5c,d).

Moreover, the sources of the four cardinal gradients are positioned at the *borders* of the I_k regions by "induction" from neighbors (Fig. 5b). This means that high-I_k agents set their source flags $s_W^{(k)} \ldots s_S^{(k)}$ to 1 if they are connected to agents from other regions $I_{k'}$. The exact flags that are switched on depend on the relative location of the regions, for example, $s_E^{(4)} = 1$ for all I_4 agents in contact with region I_5, while $s_S^{(4)} = 1$ for I_4 agents in contact with I_1, and so on (Fig. 5b,c). In cases where a particular gradient is missing because there is no adjacent border, e.g., the W sources in I_4, its sources are created from the *ends* of the opposite gradient (blue circles in Fig. 5d). Locally, an agent can recognize that it is the end of a gradient if it is a local maximum of that gradient counter n with respect to its neighborhood. Thus, in addition to source

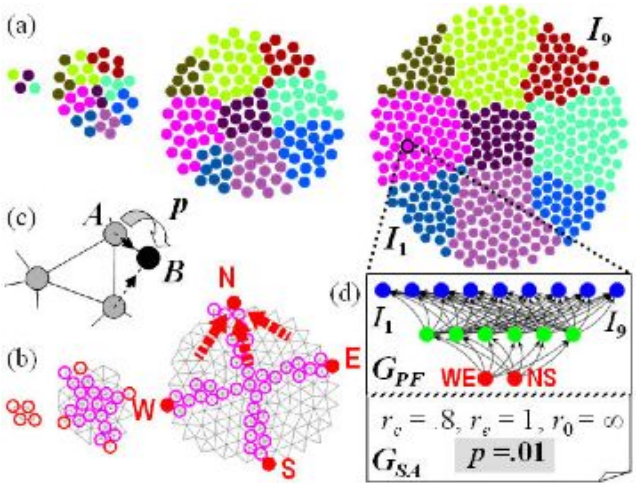

Figure 4: Simultaneous growth and patterning (SA+PF; see text). (a) Swarm growing from 4 to 400 agents by division. (b) Swarm mesh, showing gradient sources and midlines continually maintained by source migration, e.g., N moves away from S and toward WE. (c) Detail: an agent B created by A's division submits to SA forces and PF traffic. (d) Combined genetic programs inside each agent.

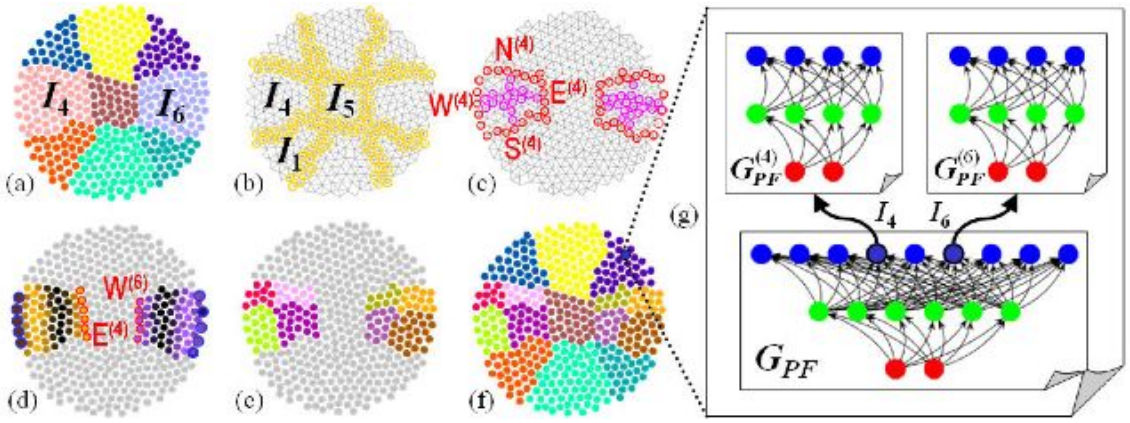

Figure 5: Modular, recursive patterning (PF[k]; see text). (a) 9-region swarm, as in Fig. 4a. (b) Border agents highlighted in yellow circles. (c) Border agents become new gradient sources at a lower scale inside certain identity regions. (d) Missing border sources arise from the ends (blue circles) of other gradients. (e,f) Subpatterning of the swarm in I_4 and I_6. (g) Corresponding hierarchical gene regulation network.

flags s, agents also contain "end flags" e_W … e_S that are switched on if the proper local-maximum conditions are filled. For example, $e_E^{(4)} = 1$ (hence $s_W^{(4)} = 1$) where $n_E^{(4)}$ is maximum, and conversely in region 6.

Modular, recursive patterning is similar to the imaginal discs of *Drosophila*; once a region has been marked to be the future site of a leg, wing or antenna (high I_k activity), a local coordinate system of morphogen gradients arises inside this region to form that organ [4]. From the artificial-life engineering viewpoint, recursive patterning is also preferable to one-shot patterning. In theory, Fig. 5f could also be produced by a direct $I(X, Y)$ mapping, but as the swarm continues to increase it would require maintaining global gradients over longer distances and would be unstable. Building a complicated image $I(X, Y)$ directly would also require maintaining a large number of pattern variables in each agent to implement every detail, and thus would be difficult to evolve. Modularity, by contrast, is an essential condition of *evolvability* [22]. In Fig. 5g, mutating G_{PF} would modify the whole body plan of Fig. 5a, whereas mutating $G_{PF}^{(4)}$ or $G_{PF}^{(6)}$ would only modify the "organs" of Fig. 5e. Moreover, without modules it would not be possible to have *differential* SA, necessary for the growth of *morphogenetic* structures and "limbs" other than blob swarms (see next subsection). Finally, modules can be *reused*, e.g., I_4 and I_6 could point to a common G_{PF} block. In summary, modularity is a desirable feature in genotypes just as in any software architecture or evolvable system. It seems that biological evolution discovered this principle naturally [3].

Modular, Anisotropic Growth (SA[k])

What is so far missing from the model is a true topological *deformation* dynamics, or "morphodynamics", that can confer non-trivial shapes to the organic system beyond simple blobs. To this aim, agents must be able to diversify their SA characteristics, *depending on their PF type and spatial position*, thus closing the feedback loop between SA and PF. In particular, they have to exhibit *inhomogeneous*, *anisotropic* cell division (varying p) and *differential* adhesion (varying V). For example, the growth of limb-like structures can be achieved by a coarse imitation of meristematic plant offshoots. In this process, only the tip or "apical meristem" of the organ is actively dividing at any time (Fig. 6). It is implemented here by letting agents have a non-zero probability of division p if and only if they are ends of a gradient (blue circles in Fig. 6c,d). These dividing cells can also control the angle of the "plane of cleavage". For example, a daughter cell B spawned by cell A can be placed opposite to the center of mass of A's neighbors (Fig. 6b). Almost equivalently, that position can be computed by factoring in the gradient values of the A's neighbors, i.e., calculating a discrete estimate of the local gradient slope in A.

Biological cells also stick to each other by means of adhesion proteins that cover their membrane. A great diversity of

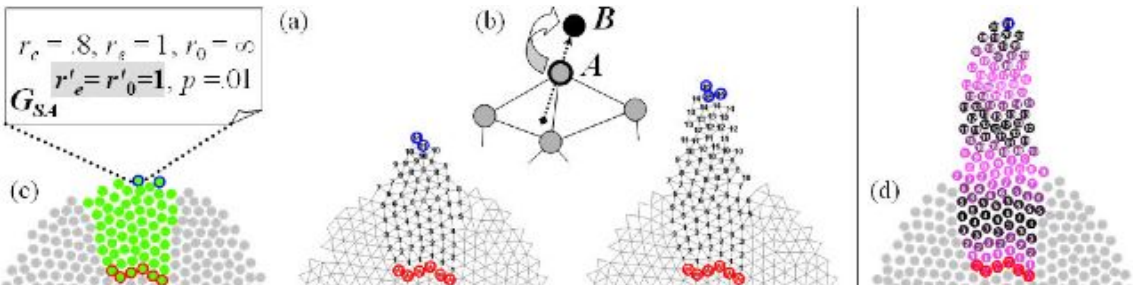

Figure 6: Modular, anisotropic growth (SA[k]; see text). (a) Genetic SA parameters are augmented with repelling V values r'_e and r'_0 used between the growing region (green) and the rest of the swarm (gray). (b) Daughter agents are positioned away from the neighbors' center of mass. (c) Offshoot growth proceeds from an "apical meristem" made of gradient ends (blue circles). (d) The gradient underlying this growth.

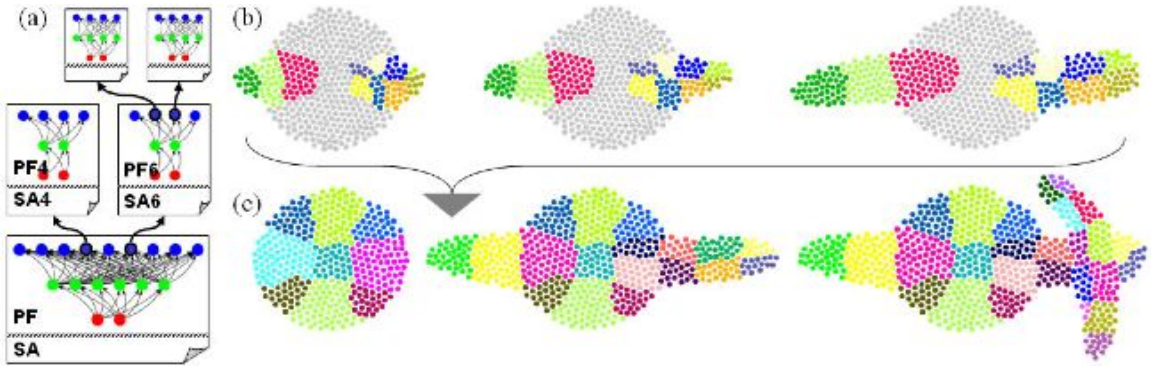

Figure 7: Modular growth and patterning (SA[k] + PF[k]; see text). (a) Example of a three-tier modular genotype giving rise to the artificial organism on the right. (b) Three iterations detailing the simultaneous limb-like growth process (Fig. 6) *and* patterning of these limbs during execution of tier 2 (modules 4 and 6). (c) Main stages of the complex morphogenesis, showing full patterns after execution of tiers 1, 2 and 3.

these proteins gives cells the ability to selectively recognize one another, thereby modulating the intercellular adhesion force or "stickiness". Some cells slide along one another without attaching, while others form tight, dense clumps. In the simple elastic force model, differential adhesion can be mimicked by varying V's parameters r_c, r_e, and r_0 depending on the agent types (Fig. 6a). For example, if agent A belongs to the limb region (green area) then $V(r_{AC})$ is attractive ($r_c < r_e = 1 << r_0$) for all neighboring agents C in that region, while it is repelling ($r'_0 < r'_e$) for all agents C outside that region (gray area). This can be decided locally by comparing the types of A and C, i.e., whether their respective highest-valued I_k nodes are the same or not. Just like inhomogeneous division, differential adhesion is an essential condition of complex shape formation [11, 15].

Modular Growth and Patterning (SA[k] + PF[k])

Putting everything together, full morphologies can develop and self-organize from a few agents (Fig. 7). These morphologies are *complex*, *programmable* and *reproducible*. They are architecturally complex because they can be made of any number of various modules and parts that are not necessarily repeated in periodic or trivial ways. They are programmable phenotypes emerging from the same genotype carried by every agent of the swarm (Fig. 7a). They are also reproducible, as their morphological structures are not left to chance but dictated by the genotype. The exact agent positions at the microscopic level are still random, but not the mesoscopic and macroscopic regions that they form.

The modularity of the phenotype is also a direct reflection of the modularity of the genotype: the hierarchical SA + PF dynamics recursively unfolds inside the different regions and subregions that it creates. Each $SA^{(k)} + PF^{(k)}$ block can be reused, either by convergent I_k links (not shown here) or by exact *duplication*. It can also *diverge* from other blocks, i.e., receive different internal genetic SA and PF parameters that give each region a different morphodynamic behavior and activity landscape. Duplication followed by divergence is the basis of *serial homology* (e.g., vertebrae, teeth, digits), a major natural evolutionary mechanism. The integration between SA and PF is controlled through the identity nodes I_k: just as these nodes turn on gene expression activity in subordinate $G_{PF}^{(k)}$ modules to create new local segmentation patterns, they also simultaneously turn on behavioral changes in subordinate $G_{SA}^{(k)}$ modules to create new morphodynamical behaviors.

There remains to determine the *scheduling* policy of genotype execution inside each agent. When does an agent decide to follow the latest $SA^{(k)} + PF^{(k)}$ branch opened by a new identity gene I_k? Since there is no centralized control in the swarm, module-switching decisions must be asynchronous. However, starting a new module k as soon as I_k's activity is high would not be a good strategy, especially while the agent's current region is still developing. For example, in the early stages of Fig. 4a, cells often change type (color) and should not start creating new subpatterns before they reach maturity. Thus there must be some regional synchronization mechanisms to help agents make scheduling decisions. The present model, however, only adopts a primitive clocked scheme based on the number of iterations. For now, all agents simply switch to the next $SA^{(k)} + PF^{(k)}$ stage if their internal timer exceeds a time point t_k set in advance and added to their genetic baggage.

Discussion

The goal of this work was to contribute to a better theoretical understanding of complex morphogenesis, especially biological, in order to reproduce it artificially and pave the way for development-based evolutionary innovation. It presented a model of pattern formation in self-assembling swarms that contained a large number of agents and displayed complex but reproducible phenotypic emergence from a modular genotypic program. As e*mbryomorphic engineering*, it essentially advocated a "fine-grain" approach to systems design based on relatively simple programmed agents. Naturally, beyond the proof-of-concept simulations presented here, and other preliminary work [8, 9], a more systematic exploration is needed. Next steps should involve the mass-production of virtual organisms to support (a) statistical analysis of shape and (b) evolutionary search based on module variation and *function*.

Future Work

From form to function. While the task of "meta-designing" laws of artificial development inspired from biology is challenging, it only constitutes the first part of an embryomorphic engineering effort. Another important question is *functional*

meta-design: once a self-developing infrastructure is mature, what computing capabilities can it support? What do its cell-agents and organ-regions actually represent in practice? In biological organisms, although cell physiology often partakes in development (e.g., electrical signals of neurons guiding synaptogenesis), there seems to be a broad distinction between developmental genes and the rest of the genome. In computing systems, these two modes could also be decoupled into two different sets of agent variables. After reaching developmental maturation, and while still fulfilling maintenance and self-repair tasks, morphogenetic SA and PF activity (i.e., division, position information and patterning signals) would give way to another type of activity subserving functional computation. Obviously, the type of computation would entirely depend on the nature of the agents: processors, software, robot parts, mini-robots, etc. In fact, in many computing domains, there is already a demand for precise self-formation capabilities. A multitude of micro-components containing the same code could self-organize without traditional VLSI precision or reliability [7, 16]. Mobile sensor and actuators could dynamically connect in self-managing networks [2]. Small-footprint software objects could diversify and self-deploy to achieve a desired level of application functionality (e.g., "immune" security). Articulated robotic parts, reconfigurable devices [14, 13, 10], or mobile robot formations [5] could also be guided by complex *and* controllable morphologies.

From ontogeny to phylogeny. After growth and function, one must also define how the system *evolves*, i.e., how it *varies* (randomly) and how it is *selected* (non-randomly). Different selection strategies are possible, either focusing on pre-specified forms, or pre-specified functions, or allowing unspecified outcomes. When *selecting for form*, a hard reverse engineering problem must be addressed: given a desired phenotype, what is the genotype that can produce it? While deterministic reverse compilation is possible in some cases [16], parameter search is difficult in general. Fitness criteria that reward only the target shapes create jagged landscapes of unreachable peaks. A smoother approach is to define a "shape distance" as an increasing function of favorable mutations. It is conjectured here that this kind of gradual search might actually *benefit*, not suffer, from the high genotype dimensionality of an embryomorphic model, compared to the direct mappings of genetic algorithms. Hierarchical gene regulatory networks might be better at providing the fine-grain mutations required by the gentle-slope search. Complex systems inherently have greater variational power, as they allow combinatorial tinkering on highly redundant parts.

However, beside gaining self-repair properties, why constrain a self-assembling system to produce a pre-defined shape? More benefits might come from such systems by *selecting for function* while leaving freedom of form. Gradual optimization could rely on a distance of *performance* to pre-defined goals, instead of shapes, allowing the most successful candidates to reproduce faster and mutate. Functional selection under free form is used in evolutionary robotic systems [14, 13], but mostly based on macroscopic genotype-phenotype encodings. Here, too, a larger number of agents, such as in multicellular embryogenesis, could prove more favorable to a successful search. Finally, in a third scenario, specifications could be relaxed to the point of *being open to surprise* and harvesting unexpected but useful organisms from a free-range menagerie. Reconciling the antagonistic poles of "planning" and "autonomy" ultimately hinges on two complementary aspects: (a) fine-grain variation-by-mutation mechanisms yielding a large number of search paths and (b) loose selection criteria yielding a large number of fitness maxima. With more search paths covering more fit regions, evolution is more likely to find good matches.

References

1. Ball, P. (1999). *The Self-Made Tapestry.* Oxford University Press.
2. Beal, J. and Bachrach, J. (2006). Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intell. Sys.*, 21(2): 10–19.
3. Callebaut, W. and Rasskin-Gutman, D. editors. (2005). *Modularity: Understanding the Development and Evolution of Natural Complex Systems*. The MIT Press, Cambridge, MA.
4. Carroll, S. B., Grenier, J. K. and Weatherbee, S. D. (2001). *From DNA to Diversity*. Blackwell Scientific, Malden, MA.
5. Christensen, A., O'Grady, R. and Dorigo, M. (2007). Morphology control in a self-assembling multi-robot system. *IEEE Robotics & Automation Magazine*, 14(4): 18–25.
6. Coen, E. (2000). *The Art of Genes*. Oxford University Press, UK.
7. Coore, D. (1999). *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*, Ph.D. thesis, Dept. of Elec. Eng. & Computer Science, MIT.
8. Doursat, R. (2006). The growing canvas of biological development: Multiscale pattern generation on an expanding lattice of gene regulatory networks. *InterJournal: Complex Systems*, 1809.
9. Doursat, R. (2008). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In Würtz, R. P., ed., *Organic Computing*, pages 167-200. Springer.
10. Goldstein, S. C., Campbell, J. D. and Mowry, T. C. (2005). Programmable matter. *IEEE Computer*, 38(6): 99–101.
11. Hogeweg, P. (2000). Evolving mechanisms of morphogenesis: On the interplay between differential adhesion and cell differentiation. *Journal of Theoretical Biology*, 203: 317–333.
12. Kirschner, M. W. and Gerhart, J. C. (2005). *The Plausibility of Life: Resolving Darwin's Dilemma.* Yale University Press, New Haven.
13. Komosiński, M. and Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4): 395–418.
14. Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature* 406: 974–978.
15. Marée, A. F. M. and Hogeweg, P. (2001). How amoeboids self-organize into a fruiting body: Multicellular coordination in *Dictyostelium discoideum*. *PNAS*, 98(7): 3879–3883.
16. Nagpal, R. (2002). Programmable self-assembly using biologically-inspired multi-agent control. *First International Conference on Autonomous Agents*, Bologna, July 15-19.
17. Salazar-Ciudad, I. and Jernvall, J. (2002). A gene network model accounting for development and evolution of mammalian teeth. *PNAS*, 99(12): 8116–8120.
18. Sayama, H. (2007). Decentralized control and interactive design methods for large-scale heterogeneous self-organizing swarms. *Advances in Artificial Life: Proceedings of the 9th ECAL*.
19. Shapiro, B. E., Levchenko, A., Meyerowitz, E. M., Wold, B. J. and Mjolsness, E. D. (2003). Cellerator: Extending a computer algebra system to include biochemical arrows for signal transduction simulations. *Bioinformatics*, 19(5): 677–678.
20. Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2): 93–130.
21. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I. and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75: 1226–1229.
22. Watson, R. A. and Pollack, J. B. (2005). Modular interdependency in complex dynamical systems. *Artificial Life*, 11(4): 445–458.
23. Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation development. *J. Theoret. Biology* 25: 1–47.