

Spatial Self-Organization of Heterogeneous, Modular Architectures

René Doursat

Abstract— On the one hand, natural phenomena of spontaneous pattern formation are generally random and repetitive, whereas, on the other hand, complicated heterogeneous architectures are the product of human design. The only examples of self-organized and structured systems are biological organisms produced by development. Can we export their *precise self-formation capabilities* to computing systems? This work proposes an “embryomorphic engineering” approach inspired by evo-devo to solve the paradoxical challenge of planning autonomous systems. Its goal is to artificially reconstruct complex morphogenesis by integrating three fundamental ingredients: *self-assembly* and *pattern formation* under *genetic regulation*. It presents a spatial computational agent-based model that can be equivalently construed as (a) *moving* cellular automata, in which cell rearrangement is influenced by the pattern they form, or (b) *heterogeneous* swarm motion, in which agents differentiate into patterns according to their location. It offers a new abstract framework to explore the causal and programmable link from genotype to phenotype that is needed in many emerging computational domains, such as amorphous computing or artificial embryogeny.

I. INTRODUCTION

A. Rethinking Computing Systems as Complex Systems

THE trend toward segmentation and distribution of large systems over a multitude of smaller and relatively simpler components is both a growing need and an inevitable fact in many domains of computer science & engineering, artificial intelligence, and robotics (see, e.g., [1]). Faced with an explosion in size and complexity of computing systems, including hardware, software and networks, engineers are gradually led, more or less willingly, to rethink these systems in terms of *complex systems*. Moreover, whether hardware in physical space or software in virtual space, components generally form a *spatially extended* system based on the topology of their interaction network, where “space” (Euclidean or not) is created by the local scope of connectivity, and limited signal or message propagation around each component [2].

In this context, various research works have investigated the possibility of obtaining *self-formation capabilities* from a variety of such complex computing systems. Since functionality is distributed over a great number of components, it would be an insurmountable task to assemble and instruct each of them individually. Rather, in a way similar to biological cells, these components should be easily mass-produced, initially as

identical copies of each other, and only acquire their specialized positions and functions by themselves within the system, once mixed together [2], [7].

For example, the field of amorphous computing has set the stage for a myriad of micro-processors containing the same instructions to self-organize without exact blueprint map or functional reliability, unlike traditional VLSI, e.g., [3], [4]. Such self-assembling components can also represent mobile sensors and actuators in complex self-managing networks [5]. In software applications (servers, security, etc.), a swarm of small-footprint software agents can diversify and self-deploy to achieve a desired level of service [8]. In robotics, too, whether articulated parts of reconfigurable devices [9]–[11], or mobile formations of mini-robots [12], there is also great demand for controllable complex morphologies.

As shown by these examples, computing systems need to transition from a state of exogenously imposed order toward increasing organizational and functional autonomy. Instead of designing systems in every detail, engineers should only “meta-design” them, i.e., focus on the generic conditions allowing their endogenous growth and evolution. In this context, understanding natural complex systems, in particular biological and social, could help launch a new generation of artificial complex systems based on paradigms still largely absent from traditional engineering, such as decentralization, autonomy and adaptation. In the same spirit as neuroscience giving rise to neural networks and neuromorphic engineering, or insect colonies to “ant colony optimization” [13], or genes to evolutionary algorithms, etc., the present work focuses on multicellular *biological development*, combined with evolution in evo-devo [14], as a new source of inspiration toward a form of “embryomorphic engineering” (introduced in [6], [7]).

B. Rethinking Complex Systems as Programmable Systems

In all the artificial self-organized systems cited above [2]–[12], the most important challenge is not simply to *allow* self-organization to happen, but first and foremost to *guide* it. Past a certain fascination for spontaneous “free” order, the next critical research question concerns the reintroduction of *programmability* and *reproducibility* into self-organization, i.e., the attempt to re-engineer emergence. Naturally, this seems to lead to paradoxical objectives: Can autonomy be planned? Can decentralization be controlled? Can evolution be designed? The answer to this puzzle lies in a *change of scale*: instead of a top-down enforcement of macroscopic structures, the new controls take the form of *local* instructions, i.e., a *developmental* “genotype”, inside every microscopic agent of

R. Doursat is Guest Researcher at the Complex Systems Institute, CNRS and Ecole Polytechnique, Paris, France (Web: <http://doursat.free.fr/>; e-mail: rene.doursat@polytechnique.edu).

the system. Through their genotype, agents can be steered toward acquiring specific characteristics and gradually modifying their behavior. The more sophisticated the genotype, the richer the variety and complexity of the overall performance, or “phenotype”. Complex systems engineering can thus be reformulated as meta-designing the genotype (at agent level) as opposed to directly designing the phenotype (at system level). Rooted in developmental and evolutionary biology, this idea was independently adapted to computing systems in “artificial embryogeny” [15]-[17], [6], [7], and amorphous computing [2]-[5]. In most of these works, one key ingredient for agent differentiation, and therefore diversity and programmability of overall system structure, is *positional information*. This article presents a morphogenetic model in which the type of each agent depends on a (limited) awareness of its own position within the system. As the system grows, agents’ positions change, causing them to switch to different subsets of rules inside the complete genotype that they carry.

Self-organized systems of abiotic physical-chemical matter generally form “simple” spatial patterns: spots, stripes, branches, waves, etc. [18]. Despite a great diversity of phenomena, these emergent structures essentially consist of repeated motifs. Their order arises from amplified fluctuations and shows statistical uniformity and randomness similar to textures. On the other hand, still outside biology, all complex, intricate structures made of segments and parts arranged in specific ways are the product of human inventiveness: computers, cars, buildings, etc. But contrary to physical systems, human constructions are made of a diversity of modules that are statistically heterogeneous and information-rich. Hence, so far, the only emergent *and* nontrivial, self-dissimilar architectures that we can see around us are living organisms. This is because biological agents (cells) carry a sophisticated micro-program (DNA) that endows them with a vast repertoire of highly non-trivial behaviors. Cells do not randomly mix but proactively position themselves in precise ways. Regions of genetic expression are not randomly distributed but highly regulated in number and position. An organism’s shape dynamically unfolds on the basis of calculations and decisions carried out by each cell at every time step. This is because DNA is a repository of stimuli-response rules, vastly superior in quantity of functional information to elementary volumes of inert physical matter. Therefore, genetic-like regulation at the agent’s level, modified by relative positional information, is the key to controlling self-organization in complex computing systems.

Another goal of this work is to clear a few common misconceptions about complex systems and shift the focus toward their often-neglected abilities to exhibit *controllable* properties, at the same time (or despite the fact) that they are self-organizing. It seems that “complex” is abusively associated with qualifiers such as “homogeneous”, “monolithic” and/or “random”. Yet, there can be a wide diversity of agents and *heterogeneity* of patterns, via positions; a complex system can be *modular*, hierarchical, and architecturally detailed at multiple scale; it can also consist of *reproducible* patterns arising from programmable agents.

C. A swarm of sophisticated agents: ABM meets MAS

The trend to break up computing systems into interacting *agents* can be traced back to two broad families of disciplines and techniques, schematically called: agent-based modeling (ABM) and multi-agent systems (MAS). Despite their great similarities (and the fact that these names, or variants thereof, are sometimes used interchangeably), it is revealing that they have different origins. ABM tends to follow the goals of a complex systems *science*, i.e., the understanding and modeling of “natural” complex systems (including spontaneously emergent human activity), whereas MAS relates to a form of complex systems *engineering*, i.e., the design of a new generation of “artificial” complex systems (including harnessing natural systems through hybrid cyber-physical networks).

ABM represents the perspective of social sciences and discrete mathematics. It arose from the need to model systems that were too complex for analytical descriptions, such as social interactions and the economy. Historically related to methodological individualism, and helped by the rise of computing power, it soon became a practical tool in many other scientific disciplines, such as ecology, biology and physics. It also converged with cellular automata (CA), an abstract framework originally created by von Neumann and Ulam to study self-replicating machines. Most of ABM is based on a combination of three types of topologies [19]: fixed grids such as square pixels, arbitrary networks with long-range connections, and 2D or 3D Euclidean space supporting irregular lattices of mobile agents with nearest-neighbor interactions.

On the other hand, MAS comes from the perspective of software engineering and artificial intelligence (AI). The big old monolithic programs were gradually replaced by clean architectural principles based on layers, modules or objects that communicate via application programming interfaces (APIs). It was realized that disentangling and removing cycles from the graph of function calls allows to group functions into code “parts”, thereby fixing, upgrading, or replacing these parts independently from each other, without having to rewrite the rest. In AI, the same idea prevailed when decomposing large systems into smaller, less exhaustive *intelligent agents*. Software agents differ from objects by the fact that they can be proactive and autonomously threaded. Multi-agent systems also differ from traditional distributed (operating) systems in that agents do not attempt to appear together transparently as one coherent, centralized system to the user. The rise of pervasive networking made MAS a practical technology.

ABM and MAS diverge slightly about their view of the role of an agent. MAS emphasizes software agents as *proxies* representing users or other physical entities and their interests (e.g., information-searching internet agents, price-bidding electronic brokers, device-monitoring automation agents [20]). Here, agents try to satisfy goals under the constraints created by the other agents and their environment. ABM rather stresses the *social* interactions among agents toward a collective emergent behavior with higher purpose. To summarize crudely, MAS envisions a limited number of heavy-weight (code-laden), selfish, intelligent agents, while ABM tends to use many light-weight (few rules), social, simpler agents.

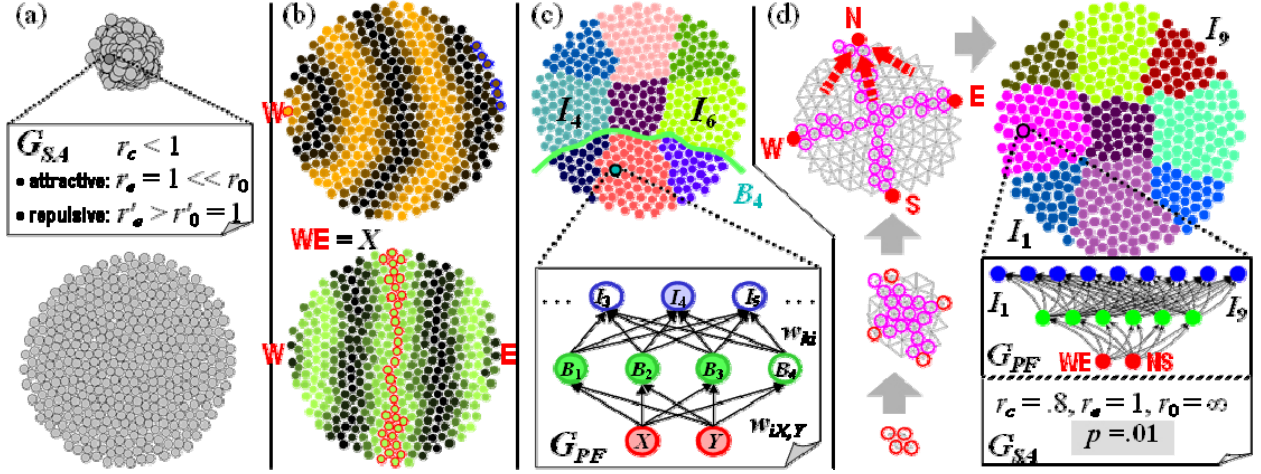


Fig. 1. Combining self-assembly and pattern formation on one level (see text). (a) Deployment of a homogeneous swarm (SA). (b) Propagation of positional information (PF-I). (c) Appearance of programmed patterning (PF-II). (d) Simultaneous growth and patterning (SA + PF).

The present model belongs to a new family of works that tend to combine both ABM and MAS philosophies by introducing a multitude of sophisticated (DNA-laden) agents, which constantly communicate and collaborate to produce a precise spatial formation. It focuses here on the abstract *developmental* computing activity of the agents as a service that they can provide independently of their actual purpose. This developmental program can be associated with any *functional* computing activity, depending on the domain-specific type and purpose of the agents (processors, sensors, robots, software modules, etc.).

II. A MODEL OF ARTIFICIAL DEVELOPMENT

The goal of this work is to understand how complex self-organization can be controlled through a computational model of *programmable* and *reproducible* artificial morphogenesis. It proposes 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) under the control of non-random, structured *genetic regulation* (GR) stored inside each agent of a swarm. To take an artistic metaphor, this is similar to mixing “self-sculpting” and “self-painting” in one composition [21]. 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.

A brief description of the model follows. First, the motion of a homogeneous swarm (pure SA, Fig. 1a) and the patterning by gradient propagation on a fixed swarm (pure PF, Fig. 1b-c) are introduced separately. Then, these two compo-

nents are combined to form reproducible growing patterns (SA + PF; Fig. 1d). The genetic program controlling these arrangements inside every agent is also explained. Finally, this combination is repeated as modules (SA^(k) + PF^(k), Fig. 2e-f) inside a larger, heterogeneous system to create complex morphologies by recursive refinement of details (Fig. 2h).

A. Deployment of a Homogeneous Swarm (SA)

The model consists of a 2-D swarm of agents that integrate two major aspects of cellular biomechanics: cell *adhesion*, in the form of elastic rearrangement, and cell *division*. Agent “shapes” correspond to mutual adhesion affinities, which are implemented by local interaction potentials V among pairs of agents. Similar to other collective motion models [22], V consists of three parts: (i) infinite repulsion for $r < r_c$, (ii) quadratic (elastic) attraction around r_e , and (iii) flat potential for $r > r_0$. Starting from a compressed swarm, agents quickly relax to a resting state, in which they tend to form a quasi-regular hexagonal mesh (Fig. 1a). At this stage, each agent possesses fixed “genetic” parameters, r_c , r_e and r_0 , denoted by G_{SA} .

B. Propagation of Positional Information (PF-I) and Programmed Patterning (PF-II)

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).

Gradient values propagate from neighbor to neighbor and establish *positional information* [23] across the swarm (Fig. 1b). For example, an agent W passes a counter variable $n_W = 0$ to its neighbors, changing it to 1 on the way. These neighbors in turn instruct their neighbors to set n_W to 2, and so on. The result is a roughly circular wave pattern of n_W values centered on W . Together with W , three other gradients, E , N and S contribute to form a 2-D coordinate system (Fig. 1d, top

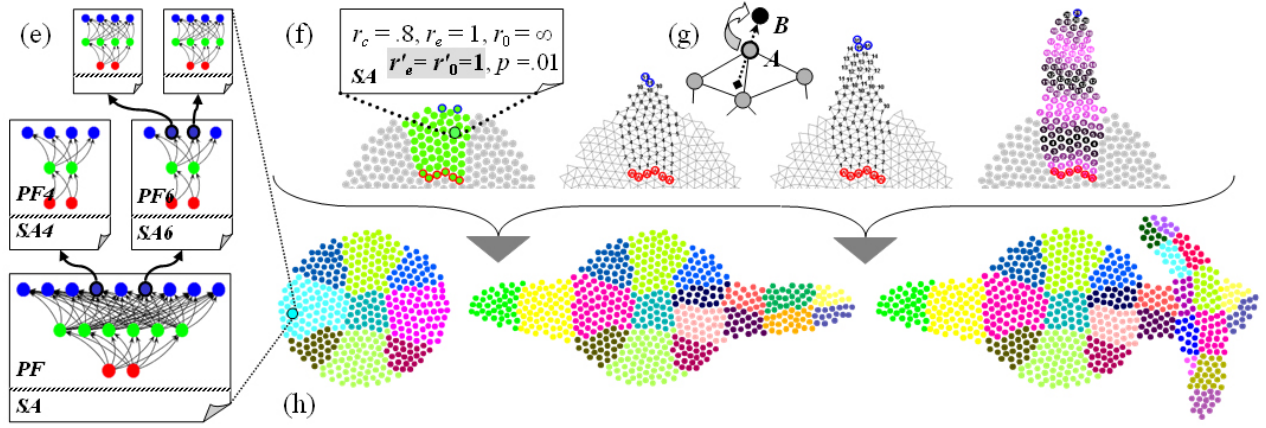


Fig. 2. Multilevel, modular growth and patterning of whole systems (see text). (e) Modular, recursive patterning (PF[k]). (f) Modular, anisotropic growth (SA[k]). (g) Daughter cells are positioned away from the neighbors' center of mass. (h) Modular growth and patterning (SA[k] + PF[k]): the full picture.

left). The combination of two antipodal gradients W and E defines an equator across the swarm, denoted by WE , which contains agents roughly equidistant from W and E , e.g., $|n_W - n_E| \leq 1$. Agents belonging to WE become in turn the sources of a new gradient X , creating a planar wave of n_X counters that propagates back symmetrically toward W and E (Fig. 1b, bottom; same with Y originating from NS). Discrete counter increments and midlines are also used to create positional information in amorphous computing systems [2]-[5].

Pattern values are then calculated on top of gradient values, thereby creating different *agent types*, which in turn will affect the SA behavior (see SA + PF integration below). This process represents the emergence of *heterogeneity*, i.e., the segmentation of the swarm into “identity regions” (Fig. 1c). Each cell contains a *gene regulatory network* G_{PF} that represents the genetic parameters of the PF process, and each identity region corresponds to a high level of expression of a particular gene I , function of variables X and Y above. It is inspired from the early striping process of *Drosophila*, controlled by a regulatory hierarchy containing five main tiers of regulatory genes [21], [14]. The present model relies on a three-tier caricature of the same idea (Fig. 1c): the bottom layer of G_{PF} contains the two positional variables X and Y ; the middle layer, n “boundary” nodes $\{B_i\}_{i=1\dots n}$; and the top layer, m identity nodes $\{I_k\}_{k=1\dots m}$. The boundary nodes compute linear discriminant functions of the positional nodes: $B_i = \sigma(w_{ix}X + w_{iy}Y - \theta_i)$, where $\{w_{ix}, w_{iy}\}_{i=1\dots n}$ are regulatory weights from X and Y to B_i , parameter θ_i is B_i 's threshold and sigmoid function $\sigma(u) = 1/(1 + e^{-2u})$. The effect of a boundary node is to segment the embryo's plane into half-planes of strong and weak expression levels, near 1 and 0 (Fig. 1c: $B_4 = 1$ in bottom third, 0 elsewhere). Finally, the identity gene levels can be derived from logical combinations of the boundary gene values, for example, through products $I_k = \prod_{i, w'_{ki} \neq 0} (w'_{ki}B_i + (1-w'_{ki})/2)$, where $w'_{ki} \in \{-1, 0, +1\}$ represent ternary weights from B_i to I_k . This means that B_i 's factor inside I_k can be either $(1 - B_i)$ or B_i .

C. Simultaneous Growth and Patterning (SA + PF)

After describing the self-assembly of a non-patterned swarm (Fig. 1a) and the patterning of a fixed swarm

(Fig. 1b,c), SA and PF are combined to create growing patterns (Fig. 1d). 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} . Another mechanism, *cell division*, is also introduced at this point. Any agent A may produce a new agent B with probability p (Fig. 2g). Agent B inherits all of A 's attributes, including genotype G_{SA+PF} and internal PF variables. It immediately starts contributing to SA forces and PF gradients, which maintain the pattern's consistency at all times in the swarm.

D. Modular, Recursive Patterning (PF[k])

Embryological patterns, however, do not develop in one shot but in numerous incremental stages [21]. An adult organism is produced through modular, recursive patterning. In the case of *Drosophila*, regions that acquire leg, wing or antenna identity (“imaginal discs”) start developing local coordinate systems of morphogen gradients to form the planned organ. In the present model, the above gene network G_{PF} is extended to include a pyramidal *hierarchy* of network modules able to generate patterns in a recursive fashion (Fig. 2e, h). First, the base network G_{PF} establishes main identity regions as before, then a few subnetworks $G_{PF}^{(k)}$ triggered by nodes I_k in G_{PF} further partition these regions into smaller identity compartments at a finer scale. Modularity is a feature as desirable in the genotype as in any other software architecture or evolvable system [24]. It seems that biological evolution also discovered this principle naturally.

E. Modular, Anisotropic Growth (SA[k])

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

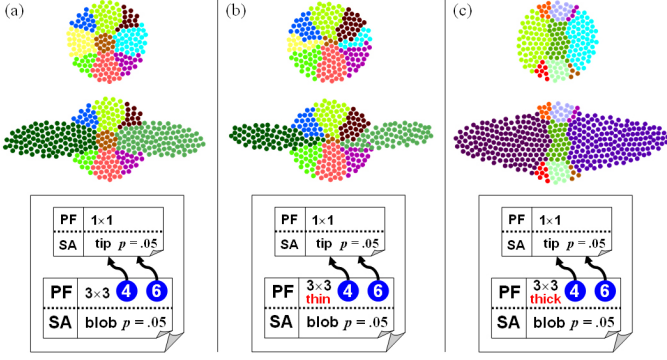


Fig. 3. Body plan mutations in a two-level organism (see text). The genotype is displayed in a schematic way, hiding the GRN weights and showing only the identity genes that give rise to secondary structures (compare with Fig. 2c). (a) “Wild type” organism. (b) “Thin” mutant. (c) “Thick” mutant.

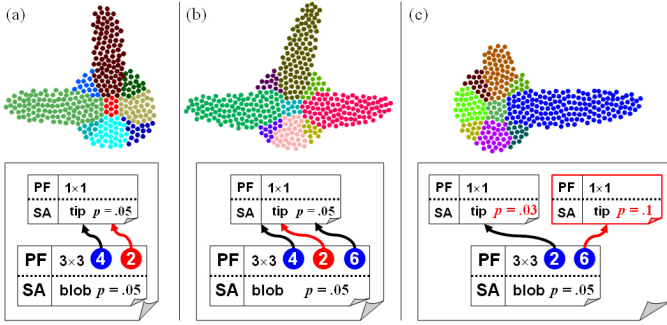


Fig. 4. Limb modularity in a two-level organism (see text). (a) Virtual “antennapedia”. (b) “Three-legged” serial homology by *duplication*. (c) *Divergence* of a serial homology.

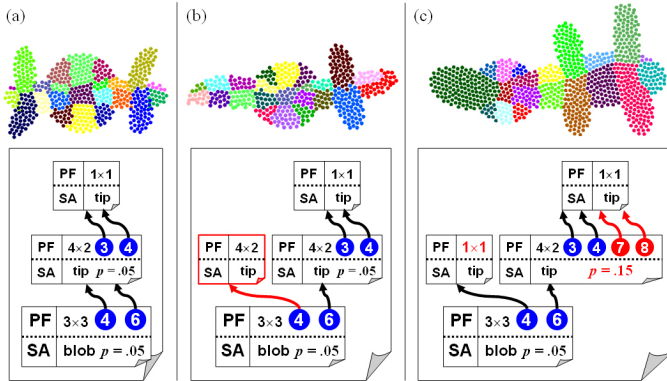


Fig. 5. Examples of three-level organisms (see text). In all these examples, smaller “digits” grow on top of the earlier limbs, which exhibit a new 2×4 or 2×6 checkerboard pattern of identity regions unseen in the previous figures.

coarse imitation of meristematic plant offshoots (Fig. 2f). In this process, only the tip or “apical meristem” of the organ is actively dividing at any time. Moreover, potential V is attractive only among agents within the limb region (green in Fig. 2f), while it is repelling between the limb and other areas (gray). Like inhomogeneous division, differential adhesion is an essential condition of complex shape formation [25], [28].

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

Putting everything together, full morphologies can develop and self-organize from a few agents (Fig. 2h). These mor-

phologies 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. 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 by exact *duplication*. It can also *diverge* from other blocks, i.e., receive different internal genetic SA and PF parameters potentially giving 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.

III. MUTATION AND EVOLUTION

Darwinian evolution consists of random *variation* followed by non-random *selection*, yet variation became the poor child of biology’s Modern Synthesis [14]. New trends in evolutionary engineering, such as artificial embryogeny (AE) [15]–[17], also emphasize the importance of constituting fundamental laws of *developmental* variations before these can be selected on the evolutionary time scale. In the framework of genetic algorithms and evolutionary computation, it means an *indirect* or *implicit* mapping (as opposed to direct or explicit) from genotype to phenotype. Understanding variation by comparing the development of different species is also the concern of “evo-devo”, a fast growing field of biology. 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. Fine-grain, hyperdistributed systems such as multicellular organisms might be in a unique position to provide a “solution-rich” space needed for successful selection.

This paper presents a few experiments involving mutations of the genotypes of embryomorph systems and their corresponding phenotypes. These systems are purely developmental and do not serve a specific function. No organism “fitness” has been defined (neither structural, nor functional) and no selection is performed. The goal is rather to illustrate the link between genotype modularity and phenotype modularity, and the programmable and predictable effect that mutating the former can have on evolving the latter. Fig. 3 shows an example of two-level modular growth and patterning and how mutations in the base module (the “body plan”) can affect its shape. The organism of Fig. 3a is taken as an arbitrary refer-

ence or “wild type”: in this particular example, identity genes 4 and 6, corresponding to the left and right regions of the center row in the base pattern (genotype’s lower SA/PF module), trigger the execution of the same limb growth (genotype’s upper SA/PF module, which does not involve further internal patterning). In Fig. 3b, the same organism has been affected by a “thin” mutation of the base body plan: although not shown, the weights of the base module’s GRN have been modified to create a checkered pattern with a narrower central row, which leaves less space for the limbs to grow and makes them actually thinner. The opposite, “thick” mutation is shown in Fig. 3c.

In Fig. 4, the modularity of the limb component (upper module of the genotype) is demonstrated by various mutations reminiscent of experiments on biological organisms such as *Drosophila*. The identity genes marking the regions (“imaginal discs”) responsible for the growth of a certain appendage are turned on and off in regions different from the wild type of Fig. 3a. For example, a virtual case of “antennapedia” (the growth of a leg where there should be an antenna) is obtained by connecting a new identity region to the limb module, here region 2 instead of region 6. This means rewiring the GRN to reflect the fact that the limb genes’ regulatory sites in the DNA have mutated to now accept gene 2’s proteins as a promoters instead of gene 6’s. In the “three-legged” mutation of Fig. 4b, these regulatory sites have duplicated themselves before mutating, accepting gene 2 *in addition* to gene 6 (not just in replacement), so that the limb module is now executed three times instead of twice. Later in the course of evolution, similar copies of the same organ can diverge and acquire specialized characteristics, as Fig. 4c illustrates. This scenario requires duplicating the entire limb module (i.e., in the DNA, the section of genome that contains all the limb’s genes and their regulatory sites) and mutating these copies independently from each other. In this artificial example, the upper module of the genotype appears now as two copies with different growth rates (as probabilities of division) $p = 0.03$ and $p = 0.1$, respectively lower and greater than the wild type’s rate $p = 0.05$, therefore creating a shorter and a longer limb. Homology is a major evolutionary process, resulting from *duplication* followed by *divergence*. Biological organisms often contain numerous repeated or “serially homologous” parts in their body plan. This is most striking in the segments of arthropods (several hundreds in millipedes) or the vertebrae, teeth and digits of vertebrates. After duplication, these parts tend to diversify and evolve more specialized structures (lumbar vs. cervical vertebrae, canines vs. molars, etc.). Homology exists not only within individuals but also between different species, as classically shown by comparing the forelimbs of tetrapods from the bat to the whale. Homology could also be explored as an important routine of artificial self-developing systems.

Finally, along the same principles, Fig. 5 shows a few cases of *three-level* organisms. After the development of limbs rooted on the basic body plan (middle tier of the genotype), additional smaller “digits” grow on top of these limbs (top tier), based on a new 2×4 checkerboard patterning that these

limbs did not previously exhibit in Figs. 3-4. Wild-type Fig. 5a presents a double bilateral symmetry, along the horizontal and vertical axes. The duplication and divergence of the middle-tier limb module in Fig. 5b prevents region 4’s limb (left) from triggering the digit module and deprives it from its digits, compared to region 6’s limb (right), which reduces the bilateral symmetry to the horizontal axis only. Fig. 5c is a further mutation of Fig. 5b, in which region 6’s limb has accelerated its growth ($p = 0.15$) and expanded into a 2×6 checkered pattern able to support the development of two new digits, whereas, on the contrary, region 4’s limb has continued to wither toward an undifferentiated stump (division slowing down to $p = 0.03$).

IV. CONCLUSION

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 [25], [26], or with lesser morphogenetic abilities [27], [4], are among these few notable achievements. Other interesting studies have explored pairs of mechanisms: SA and PF, no GR—self-assembly based on cell adhesion and signaling pattern formation, but using only predefined cell types without internal genetic variables, e.g., [28]; PF and GR, no SA—non-trivial pattern formation from instruction-driven intercellular signaling, but on a fixed lattice without self-assembling motion, e.g., [29], [3]; SA and GR, no PF—heterogeneous swarms of genetically programmed, self-assembling particles, but in empty space without mutual differentiation signals, e.g., [30].

Naturally, beyond the proof-of-concept simulations presented here, 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. First, future work should investigate *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 fact, the problem is most often reverse: there is a need for precise self-formation capabilities in many systems made of otherwise functionally computing agents, whether processor-carrying micro-units, sensors and actuators, software agents, robot parts, mini-robots, etc. Second, after growth and function, one must also define how the system *evolves*, i.e., how it *varies* (randomly) and how it is *selected* (nonrandomly). Different selection strategies are possible, either focusing on pre-specified forms, pre-specified functions, or allowing unspecified outcomes. Reconciling the antagonistic poles of “planning” and “autonomy” will ulti-

mately hinge 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] R. P. Würtz, Ed., *Organic Computing*, Springer-Verlag, 2008.
- [2] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, Jr., R. Nagpal, E. Rauch, G. Sussman, and R. Weiss, "Amorphous computing," MIT Artificial Intelligence Laboratory memo no. 1665, Aug. 1999.
- [3] D. Coore, *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*, Ph.D. thesis, Dept. of Electrical Engineering & Computer Science, MIT, Feb. 1999.
- [4] R. Nagpal, "Programmable self-assembly using biologically-inspired multi-agent control," *First International Conference on Autonomous Agents*, Bologna, Italy, July 15-19, 2002.
- [5] J. Beal and J. Bachrach, "Infrastructure for engineered emergence on sensor/actuator networks," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 10-19, 2006.
- [6] R. Doursat, "The growing canvas of biological development: Multiscale pattern generation on an expanding lattice of gene regulatory networks," *6th International Conference on Complex Systems*, New England Complex Systems Institute, Boston, MA, June 25-30, 2006.
- [7] R. Doursat, "Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering," in *Organic Computing*, R. P. Würtz, Ed., Springer-Verlag, pp. 167-200 (ch. 8), 2008.
- [8] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443-473, 2000.
- [9] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, pp. 974-978, 2000.
- [10] M. Komosiński and A. Rotaru-Varga, "Comparison of different genotype encodings for simulated three-dimensional agents," *Artificial Life*, vol. 7, no. 4, pp. 395-418, 2001.
- [11] S. C. Goldstein, J. D. Campbell, and T. C. Mowry, "Programmable matter," *IEEE Computer*, vol. 38, no. 6, pp. 99-101, 2005.
- [12] A. Christensen, R. O'Grady, and M. Dorigo, "Morphology control in a self-assembling multi-robot system," *IEEE Robotics & Automation Magazine*, vol. 14, no. 4, pp. 18-25, 2007.
- [13] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [14] M. W. Kirschner and J. C. Gerhart, *The Plausibility of Life: Resolving Darwin's Dilemma*, Yale University Press, New Haven, 2005.
- [15] K. O. Stanley and R. Miikkulainen, "A Taxonomy for artificial embryogeny," *Artificial Life*, vol. 9, no. 2, pp. 93-130, 2003.
- [16] P. Bentley and S. Kumar, "Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem," In *Proc. of the Genetic and Evolutionary Computation Conference*, Orlando, Florida, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., Morgan Kaufmann, pp. 35-43, 1999.
- [17] J. F. Miller and W. Banzhaf, "Evolving the Program for a Cell: From French Flags to Boolean Circuits," in *On Growth, Form and Computers*, S. Kumar and P. Bentley, Eds., Elsevier Academic Press, 2003.
- [18] P. Ball, *The Self-Made Tapestry: Pattern Formation in Nature*, Oxford University Press, 1999.
- [19] C. M. Macal, and M. J. North, "Tutorial on agent-based modeling and simulation part 2: How to model with agents," in *Proc. of the 2006 Winter Simulation Conference*, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, Eds., pp. 73-83, 2006.
- [20] M. Wooldridge, *An Introduction to Multiagent Systems*, John Wiley and Sons Ltd, February 2002.
- [21] E. Coen, *The Art of Genes*, Oxford University Press, UK, 2000.
- [22] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226-1229, 1995.
- [23] L. Wolpert, "Positional information and the spatial pattern of cellular differentiation development," *J. Theor. Biology*, vol. 25, pp. 1-47, 1969.
- [24] R. A. Watson, and J. B. Pollack, "Modular interdependency in complex dynamical systems," *Artificial Life*, vol. 11, no. 4, pp. 445-458, 2005.
- [25] P. Hogeweg, "Evolving mechanisms of morphogenesis: On the interplay between differential adhesion and cell differentiation," *Journal of Theoretical Biology*, vol. 203, pp. 317-333, 2000.
- [26] I. Salazar-Ciudad and J. Jernvall, "A gene network model accounting for development and evolution of mammalian teeth," *PNAS*, vol. 99, no. 12, pp. 8116-8120, 2002.
- [27] B. E. Shapiro, A. Levchenko, E. M. Meyerowitz, B. J. Wold, and E. D. Mjolsness, "Cellerator: Extending a computer algebra system to include biochemical arrows for signal transduction simulations," *Bioinformatics*, vol. 19, no. 5, pp. 677-678, 2003.
- [28] A. F. M. Marée and P. Hogeweg, "How amoeboids self-organize into a fruiting body: Multicellular coordination in *Dictyostelium discoideum*," *PNAS*, vol. 98, no. 7, pp. 3879-3883, 2001.
- [29] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell, "The segment polarity network is a robust developmental module," *Nature*, vol. 406, pp. 188-192, 2000.
- [30] H. Sayama, "Decentralized control and interactive design methods for large-scale heterogeneous self-organizing swarms," *Advances in Artificial Life: Proceedings of the 9th ECAL*, 2007.