Doursat, R., Sayama, H. & Michel, O. (2013) A review of morphogenetic engineering. *"Frontiers of Natural Computing" (FNC 2012) Special Issue.* M. Lones, A. Tyrrell, S. Stepney & L. Caves, eds. *Natural Computing* **12**(2): 517-535.

A Review of Morphogenetic Engineering

(**Running title:** A Review of Morphogenetic Engineering)

René Doursat*[1], Hiroki Sayama[2], and Olivier Michel[3]

[1]School of Biomedical Engineering, Drexel University
3141 Chestnut Street, Philadelphia, PA 19104, US (rene.doursat@drexel.edu) **corresp. author*

[2]Department of Bioengineering, Binghamton University SUNY
PO Box 6000, Binghamton, NY 13902, US (sayama@binghamton.edu)

[3]Faculté des Sciences et Technologie, Université de Créteil
61, ave du Général de Gaulle, 94010 Créteil Cedex, France (olivier.michel@u-pec.fr)

Abstract

Generally, phenomena of spontaneous pattern formation are random and repetitive, whereas elaborate devices are the deterministic product of human design. Yet, biological organisms and collective insect constructions are exceptional examples of complex systems that are both architectured *and* self-organized. Can we understand their precise self-formation capabilities and integrate them with technological planning? Can physical systems be endowed with information, or informational systems be embedded in physics, to create autonomous morphologies and functions? To answer these questions, we have launched in 2009, and developed through a series of workshops and a collective book, a new field of research called *Morphogenetic Engineering*. It is the first initiative of its kind to rally and promote models and implementations of complex self-architecturing systems. Particular emphasis is set on the *programmability* and computational abilities of self-organization, properties that are often underappreciated in complex systems science—while, conversely, the benefits of self-organization are often underappreciated in engineering methodologies[1].

Keywords: *agent-based modeling, artificial life, collective construction, complex systems, evolutionary development, generative grammars, morphogenesis, self-organization, swarm robotics, systems engineering.*

---

[1] This paper is an extended version of [1].

1 Introduction

Classical engineered products (mechanical, electrical, computer, civil) are generally made of a number of unique, heterogeneous components assembled in very precise and complicated ways. They are expected to work as deterministically and predictably as possible following the specifications given by their designers (Fig. 1d). By contrast, self-organization in natural systems (physical, biological, ecological, social) often relies on myriads of identical agents and essentially stochastic dynamics. Admittedly, here, nontrivial patterns and collective behavior can emerge from relatively simple agent rules—a fact often touted as the hallmark of *complex systems* (Fig. 1a). Yet, the great majority of these naturally emergent motifs (spots, stripes, waves, clusters, etc. [2]) are essentially stochastic and can be guided or reshaped only through external boundary conditions. They are fully described with a few statistical variables, such as order parameters, but do not exhibit an *intrinsic architecture* like machines and industrial systems do in their hardware and software.

There are, however, major exceptions that blur this apparent dichotomy and show a possible path toward the alliance of pure self-organization and elaborate architecture.

1.1 Self-Organized Systems Already Showing an Architecture

Certain types of biological systems distinguish themselves by strong "morphogenetic" properties (Fig. 1b), which are much more sophisticated than texture-like pattern formation. This is especially the case with embryogenesis, the self-assembly of myriads of cells into detailed anatomies. It is also seen in insect colonies, where swarm collaboration by "stigmergy" (communication via traces left in the environment) create giant constructions. Multicellular organisms are composed of organs and appendages finely arranged in very specific ways, yet they entirely self-assemble through a decentralized choreography of cell proliferation, migration and differentiation. This unfolds under the guidance of genetic and epigenetic information spontaneously evolved over millions of years and stored in every cell [3, 4]. Similarly, but at a higher scale, social insects such as termites, ants or wasps are also capable of collectively building extremely complicated and well organized nests [5] without the need for any overall blueprint or chief architect directing them from the outside. They, too, are individually guided by a diverse repertoire of local coordination rules on how to respond to different types of visual, tactile or chemical stimuli.

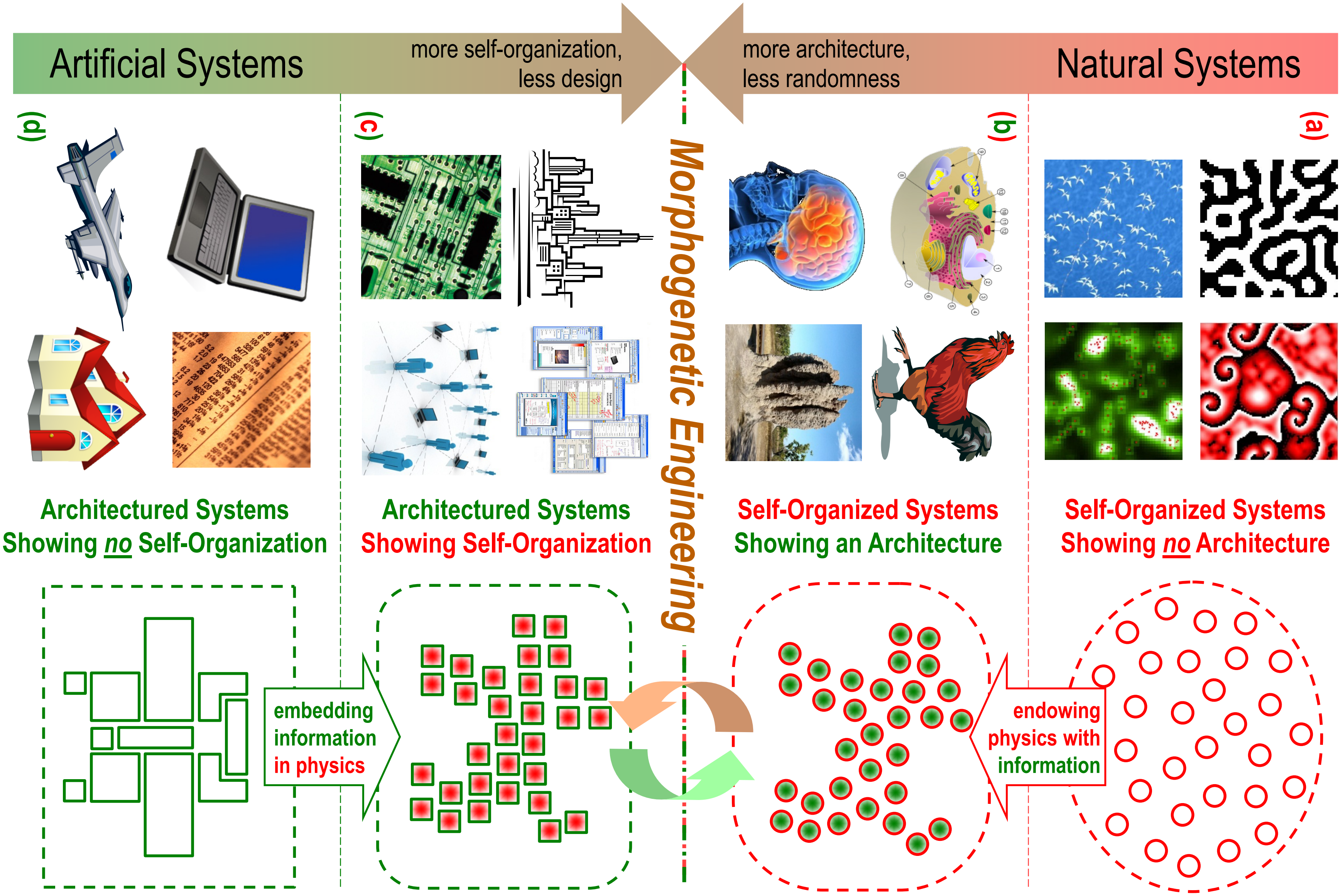
Artificial Systems
more self-organization, less design
more architecture, less randomness
Natural Systems
(d)
(c)
Morphogenetic Engineering
(b)
(a)
Architectured Systems Showing *no* Self-Organization
Architectured Systems Showing Self-Organization
Self-Organized Systems Showing an Architecture
Self-Organized Systems Showing *no* Architecture
embedding information in physics
endowing physics with information

Figure 1: Four families of systems representing various degrees of self-organization (vs. design), and architecture (vs. randomness): (a) Most natural complex systems are characterized by stochasticity, repetition and statistical uniformity: activator-inhibitor pattern formation (stripes and spots), traveling waves in chemical reaction, bird flocks, slime-mold aggregation (all screenshots of NetLogo simulations). (d) At another extreme, human-made devices (computers, programs, vehicles, buildings) are centrally and precisely designed, leaving almost no room for autonomy. There, self-organization and emergence are much more of a nuisance than a desired outcome. (b-c) *Morphogenetic Engineering* (ME) is positioned in the middle. (b) On the one hand, ME strives to understand how certain natural self-organized systems exhibit a specific architecture, i.e. how physical systems can be endowed with more information and sophisticated computational abilities. For this, it proposes—and extends into the virtual domain of artificial life, i.e. "life as it could be"—new models for biological cells, multicellular organisms, nervous systems, and collective insect constructions. (c) Conversely, ME also looks at architectured systems that have reached unplanned levels of distribution and self-organization (urban sprawl, open-source software, automatically designed processors, techno-social networks), i.e. how informational and computational artifacts can be embedded in the physical constraints of space and "in materio" granularity. There, it pushes the envelope of "emergent engineering" [6] by inventing new systems that replace improvised with *programmed* complexification. See Fig. 2 for a zoom into the ME domain.

These natural cases trigger whole new questions: How do biological populations (of cells or organisms) achieve morphogenetic tasks so reliably? Can we export their self-formation capabilities to engineered systems? What would be the principles and best practices to create such morphogenetic systems?

1.2 Architectured Systems Already Showing Self-Organization

Conversely, human-made artifacts already exhibit complex systems effects on a large scale (Fig. 1c). For example, the explosion in size and *distribution* [7] of information and communication technology (ICT) systems over a multitude of smaller entities has become an inescapable reality of computer science and engineering, artificial intelligence and robotics at all scales—whether in hardware (components, modules), software (objects, agents), or networks (services, applications). Similarly, human superstructures have become "naturally" self-organized complex systems through their unplanned, spontaneous emergence and adaptivity arising from a multitude of rigidly designed individual structures: cities have emerged from buildings, traffic jams from cars, Internet from routers, markets from companies, and so on. Finally, ubiquitous ICT capabilities, connecting human users and computing devices in unprecedented ways, have also given rise to complex techno-social "ecosystems" in all domains of society. The old centralized oligarchy of providers (of data, knowledge, applications, goods) is being gradually replaced by a dense heterarchy of proactive participants (patients, students, users, consumers).

In all these domains, the challenge is in fact complementary to the previous section: it is to regain some form of guidance or control over collective effects, but without reinstating a centralization that would compromise the benefits of local interactions. We want to better understand and steer these phenomena—although we will never again place every part or participant, foresee every event, or control every process.

1.3 Toward Programmable Self-Organization

In sum, while certain natural complex systems seemingly exhibit all the attributes of architectured systems, certain artificial systems have also become full-fledged objects of research on self-organization. Such cross-boundary examples open two opposite avenues converging toward a new central field, which we call *Morphogenetic Engineering* (ME) and define as follows:

> Morphogenetic Engineering explores the design, implementation, and control (directly, by programming, and/or indirectly, by learning or evolution) of the agents of complex systems capable of giving rise autonomously and reproducibly to large heterogeneous architectures that will support a set of desired functionalities, without relying on any central planning or external drive.

Said otherwise, while the existing phenomena described above testify to the possibility of *programmable self-organization*, the challenge of ME is to tap into this vast potential by inventing new "programmable complex systems" or, symmetrically, "self-organized engineered systems". Continuing with the two perspectives exposed above, this can be achieved in two complementary, and ultimately equivalent, ways:

- **Endowing physical systems with information** (Fig. 1a→b): Starting from the scientific understanding and modeling of “random” natural complex systems, such as patterns and flocks, and “architectured” ones, such as embryogenesis and termite mounds—especially focusing on what distinguishes them—ME aims to generalize the transition from one to the other and push the envelope to obtain new morphogenetic abilities from original systems. For example, making bird flocks virtually heterogeneous by diversifying their cohesion and alignment parameters, as if mixing different species (“swarm chemistry” [8]), can result in surprisingly complex and robust morphologies. Similarly, giving virtual wasps a pheromone that they can lay down and follow like termites (“waspmites” [9]) enhances their computation abilities, and transforms their usually repetitive nests into more elaborate constructions. In modern biotechnological endeavors such as synthetic biology [10, 11], real-world genomic information can also be tampered with in specific ways to steer the emergent collective behavior of cellular populations toward new outcomes, whether for biomedical applications (such as organ growth) or “natural computing” [12] (such as organic processors).

- **Embedding informational systems in physics** (Fig. 1d→c): In the other direction, the *de facto* and ever increasing trend for technical systems to comprise a heterarchy of numerous small components, as in parallel computing, swarm robotics, multi-agent software, or peer-to-peer networks, should be amplified, not fought, and taken to new levels of programmable complexity. Engineers will have to abandon top-down imposed design and rethink their devices in terms of natural complex systems, approaching them rather by bottom-up “meta-design”, i.e. the generic mechanisms allowing their self-assembly, self-regulation and evolution. The project of embedding ICT systems in the physical constraints of space and “in materio” granularity has been pioneered by innovative fields such as amorphous computing [13], spatial computing [14, 15, 16], organic computing [17], complex systems engineering [18] or emergent engineering [6]. ME, for its part, focuses on the strong *architectural* and complex functional properties of these emergent systems, and how these properties can be influenced or programmed at the microlevel.

As the works reviewed here will show or hint at (Section 4), the many potential applications of ME in artificial systems and hybrid "techno-natural" systems include self-assembling mechanical components and robots, self-organizing builder robots, self-morphing particle swarms, self-coding software, self-balancing pervasive services, but also self-constructing buildings, self-configuring manufacturing lines, or self-managing energy grids. They are all based on a multitude of components, modules, software agents, devices and/or human users that create their own network and collective dynamics solely on the basis of local rules and peer-to-peer interactions.

The new core challenge posed by ME is then a reverse engineering one: *How can the agents' micro-rules be inferred from the system's macro-objectives?* In a way, the paradox that must be solved is "directing the decentralization", i.e. preparing the conditions favorable to a nonrandom, reliable self-organization of a highly distributed system. At the same time, it is also letting the parameters of this process freely evolve in order to generate innovative structures and functions. Finding useful ME systems will require matching loose selection criteria with productive variation mechanisms. The first point concerns the openness of meta-designers to "surprising" outcomes; the second point concerns the intrinsic ability of complex systems to create a "solution-rich" space [18] by combinatorial tinkering on highly redundant parts.

In any case, the rallying call toward meta-design is: Don't build a system directly, but shape its building blocks in such a way that they build it for you—and can also come up with new systems you hadn't thought of.

2 Endowing Physical Systems with Information

2.1 Natural Complex Systems

Complex systems (CS) are generally defined as large sets of elements that interact locally, among each other and with their nearby environment, to produce an emergent collective behavior at a macroscopic scale. They are characterized by a high degree of decentralization, and the ability to self-assemble and self-regulate. Most CS are also *adaptive*, and named CAS [19] for that matter, in the sense that they are able to learn or evolve by themselves on the longer term toward further innovation. In general, this happens by feedback from their external fitness, i.e. overall level of success in their environment, onto their internal structure and the behavior of their elements—

whether directly via internal learning mechanisms, or indirectly via external selection mechanisms.

The elements or "agents" composing CS follow local rules that can be more or less sophisticated. Often, these rules are themselves internally structured as networks of smaller entities. For example, one cell can be modeled as a self-regulatory network of genetic switches; one social agent (ant, software process) as a decision graph or finite state machine. On the other hand, agents can also interact more collectively at the level of local clusters or subnetworks that combine in a modular fashion to form larger structures, and so on. Thus, from both perspectives, CS can often be described as "networks of networks" on several hierarchical levels. Generally, the higher levels connecting elements or clusters of elements are spatially extended (cell tissues, cortical areas, ant colonies, computer networks), whereas the lower levels inside elements are nonspatial (gene networks, neural assemblies, rulesets). Elements follow the dynamics dictated by their inner network and, at the same time, influence neighboring elements through the emission and reception of signals (chemical, electrical, software packets).

2.2 Augmented Complex Systems

In this vast interdisciplinary field of complex systems, a less addressed, yet critical, research ambition is to look beyond the usual fascination for "free-range" order or unstructured patterning (Fig. 1a), and explore the interplay of *programmability* with self-organization (Fig. 1b-c). It is an often underappreciated ability of CS to be controllable at the same time that they are self-organizing. Too often, the emergent patterns and behaviors of CS are construed as "homogeneous", "monolithic", or "random" aggregates of micro-level components, especially in statistical physics and other analytical research areas. Yet, CS can contain a wide diversity of agents and *heterogeneity* of patterns, via positions; they can be *modular*, hierarchical, and architecturally detailed at multiple scales; they can also consist of *reproducible* structures arising from programmable agents.

With the goal of "re-engineering emergence", the most important challenge is not simply to observe how any kind of self-organization can happen, but to understand how self-organization is, and can be, *guided*. Thus models relevant to ME will not be found in the traditional statistical approaches to natural CS (Fig. 1a), such as random patterning [20, 21], uniform flocking [22], or undirected networking [23, 24, 25], but rather in virtual, extrapolated versions of these models, where homogeneous, stateless agents are replaced with heterogeneous, stateful and *computational* ones. Other models will come directly from genuinely *morphological* CS, such as embryogenesis and collective insect constructions (Fig. 1b). In both cases, ME resides in (i) the relative sophistication and variety of the elements and (ii) their ability to combine together in many different ways to form precise and reproducible architectures.

Naturally, this ambition seems to lead to paradoxical objectives: Can autonomy be planned? Can decentralization be directed? The answer lies in a change of scale: instead of a top-down enforcement of macroscopic structures, the new ME controls take the form of microscopic instructions inside every agent of the system. These instructions should also diversify by varying with the agent's current type and position, creating subtypes that will in turn trigger new rules, and so on. This process introduces the required degree of *heterogeneity* in order for a system to exhibit a new range of behaviors, more sophisticated than simple random patterning, flocking or clustering.

3 Embedding Informational Systems in Physics

3.1 Artificial Life Designs

The interdisciplinary field of artificial life (Alife) is chiefly concerned with the simulation of life-like, organismal processes through computer programs, robotic devices, or even new uses of biotic components. Researchers in Alife attempt to design and construct systems that have the characteristics of living organisms or societies of organisms out of nonliving parts, whether virtual (software agents) or physical (electromechanical parts, chemical compounds, etc.). Alife is therefore a bottom-up synthetic attempt to recreate biological phenomena in order to produce adaptive and intelligent systems. In this sense, it can be contrasted with the historical top-down analytical approach of artificial intelligence (AI), which was based on symbolic systems. Alife actively promotes biology-inspired engineering as a new paradigm that would complement or replace classical physics-based engineering. This opens entirely new perspectives in software, robotic, electrical, mechanical or civil engineering: Can a device or edifice construct itself from a reservoir of components? Can a robot rearrange its parts and evolve toward better performance without explicit instructions? Can software agents collectively innovate in problem-solving tasks?

Among the great variety of biological systems that inspire and guide Alife research, three broad areas can be distinguished by the scale of their components: (a) at the micro-scale, chemical, cellular and tissular systems; (b) at the meso-scale, organismal and architectural systems; and (c) at the macro-scale, population and societal systems. Artificial molecular and cellular models focus on the spontaneous organization of complex chemical and organic structures, such as DNA/protein self-assembly [26], or embryonic development [27]. Applications are linked to nanotechnologies for biomedical or integrated electronic purposes ("smart materials", MEMS). On the anatomical and functional level, robotic parts (limbs, sensors, actuators) and local behavioral modules are coupled to produce a global behavior in a single autonomous device, aiming toward adaptivity and non-symbolic intelligence. This is the scope of reactive, behavior-based [28] or embodied robotics [29], exemplified by insect-like robots and evolving or reconfigurable mechanical morphologies [30, 31]. Entire populations of virtual or robotic creatures also constitute important objects of interest for their unique properties of collective self-organization and diversity-inducing evolution. Generically termed "swarm intelligence", new methodologies such as ant colony optimization (ACO) [32, 5] or particle swarm optimization (PSO) [33] are derived from the observation of animal societies and applied to problem-solving tasks. Finally, these three scales can be integrated in different ways to create complete systems.

3.2 Artificial Life Complexity

Generally, but not always, Alife devices are of a distributed nature and operate on a multitude of interacting components. From its origins in cellular automata (CA), and by its very definition, Alife covers or intersects several other paradigms of *distributed* systems, which are the rule in biotic systems: neural networks (learning, sensorimotor faculties), complex networks (from gene regulation to ecosystems), swarm intelligence (insect colonies, collective motion), generative and developmental systems (embryogenesis, morphogenesis). Yet, despite the inherent propensity of Alife to study decentralized and self-organized processes, a great number of researchers in bio-inspired engineered disciplines such as artificial neural networks (ANNs) or evolutionary computation (EC, which comprises genetic algorithms) have generally taken a rather different course and, in contrast with natural systems, have shifted their focus to classically designed, centralized and nondevelopmental systems. Their efforts have been mainly invested in *optimization* problems, where "emergence" is no more a desired property to be exploited. It is striking that a large class of ANN models have never included population coding, recurrent connections or temporal correlations, although Hebb's cell assemblies [34] and Hopfield's distributed attractor dynamics [35] have pioneered the field. Similarly, today's EC conferences include only a minority of complex systems topics, although the inventor of genetic algorithms, John Holland (and a long-time affiliate of the Santa Fe Institute) always referred to evolutionary search *within* the framework of complex adaptive multi-agent systems [19]. Therefore, just as we saw in Section 2 that there was little engineering thinking in the complex systems community, there is also surprisingly little complex systems thinking in a sizeable part of the Alife community.

Can we put back the "bio" in bio-inspiration and design genuinely decentralized and self-organized artifacts? Although themselves emerging from a hundred billion neurons, our human cognitive faculties create the illusion of a central consciousness or viewpoint, which must make great efforts to comprehend truly parallel processes. We are strongly biased toward identifying central causes, and spontaneously tend to ascribe the generation of order and meaning to a single entity equipped with a lot of information (one gene, one cell, one neuron, one individual). Even when we know that this entity does not have intentions or does not even exist as such, we cannot help but follow anthropomorphic stereotypes such as controller, organizer, manager, or leader. This is why we traditionally refer to systems containing multiple, intricate causal and influence links as "complex"—whereas in fact those so-called complex systems might well turn out to be "simpler" than our familiar contraptions with their uniquely exact arrangement. This is also why we so irresistibly pull back our objects of study and technologies toward the centralized paradigm, to the detriment of the distributed paradigm, even in bio-inspired Alife disciplines.

Heteronomous human-designed order is probably the most sophisticated of all forms of organization, as it requires an external intelligence to come to existence. In natural living systems, by contrast, autonomously evolved and decentralized order is the natural norm because it is the most cost-effective: information is distributed over a large number of relatively ignorant agents, making it easier to create new order states by evolving and recombining their local interactions. To imitate Ulam's famous quip about nonlinear systems being the "non-elephant species" of physics (i.e. the overwhelming majority), the pervasiveness of self-organized systems (as opposed to designed ones) make them the non-elephant species of systems science—yet they are still today the least familiar. Biological systems are *not* engineered and human-made systems could learn much more from them.

Table 1: Taxonomy of ME works presented in this review, Sections 4.1 to 4.4.

Morphogenetic Engineering			
Category I: by Constructing or Assembling/Fitting	**Category II: by Coalescing** or Sync-ing/Swarming	**Category III: by Developing** or Growing/Aggregating	**Category IV: by Generating** or Rewriting/Inserting
based on:	based on:	based on:	based on:
• Self-rearranging robotic parts	• Robotic agents	• Artificial (evolutionary) development	• Biologically inspired grammars
• Self-assembling mobile robots	• Software particles	• Developmental animats	• Graph and swarm grammars
• Block constructions	• Programming matter, computing in space	• Morphogenetic patterning	• Evolutionary grammars

4 A Review of Morphogenetic Engineering Approaches

There are many research works that incarnate the ME manifesto presented in Sections 1-3, whether they were developed before we launched this field or they were intended as contributions to the workshops and edited book that we have organized (starting in 2009). In any case, we chose to group them for this review into four main categories according to the type of *dynamical process* characteristic of their morphogenetic models (Table 1, Fig. 2). They can be summarized as follows:

- **Category I: Constructing** (or Assembling, Fitting): A small number of mobile agents or components attach to each other or assemble blocks to build a precise "stick-figure" structure.
- **Category II: Coalescing** (or Synchronizing, Swarming): A great number of mobile agents flock and make together dense clusters, whose contours adopt certain shapes.
- **Category III: Developing** (or Growing, Aggregating): The system expands from a single initial agent or group by division or aggregation, forming biological-like patterns or organisms.
- **Category IV: Generating** (or Rewriting, Inserting): The system expands by successive transformations of components in 3D space, based on a grammar of "rewrite" rules.

Here, "agent" refers to a robot (physical or simulated), a biological model (molecule, cell or insect), or an abstract dot in virtual space, while "component" refers to a structural piece (physical or simulated). Naturally, this four-part division is neither canonical nor clear-cut; it is only one among various alternatives. Not only do certain studies include several models belonging to more than one category, but they also intersect or differ along several other dimensions including: physical vs. virtual agents, biologically motivated vs. engineering-based approaches, cell-inspired vs. insect-inspired algorithms, 2D/3D Euclidean space vs. network topologies, degree of robustness and self-repair capabilities, and so on. Other terms such as "swarm" and "self-assembly" are used extensively in the majority of publications, hence do not constitute clearly distinguishing features. The above categorization, however, appeared to be one of the most meaningful and to best highlight the diversity of ME systems types.

4.1 Category I: Constructing

In this first category, which could also be named "Assembling" or "Fitting", ME systems are characterized by relatively few robots or components (possibly originating from a larger, ambient pool) that create precise formations by either attaching to each other or bringing blocks together. The built structures are coarse-grained and often look like "stick figures", i.e. combinations of chains and crosses made of 1-unit wide segments. The space is the 2D plane, with occasional vertical elevation into 3D by stacking or folding. Most works in this category feature physical and/or virtual morphogenetic robotic systems, while others distinguish themselves by the mechanical self-assembly of inert pieces (e.g. by shaking).

In a review of *morphogenetic robotics*, Jin and Meng [36, 37] (Fig. 2:3) examine methodologies for designing self-organizing, self-reconfigurable and self-adaptive robots inspired by biological morphogenesis. They categorized these systems into three main areas: swarm robotics, modular robots and body-brain co-design, and discuss their relationships with *epigenetic robotics*, which deals with cognitive development, and *evolutionary robotics*, which is concerned with the automated design of controllers. We follow in this section the same broad distinction between "swarm" and "modular" robots, keeping our focus on morphogenetic abilities, and with the additional limitation to small-scale and exact-fit construction systems. In a third subgroup, we look at nonrobotic block structures, whether made by external "worker" robots or arising spontaneously from physical interactions.

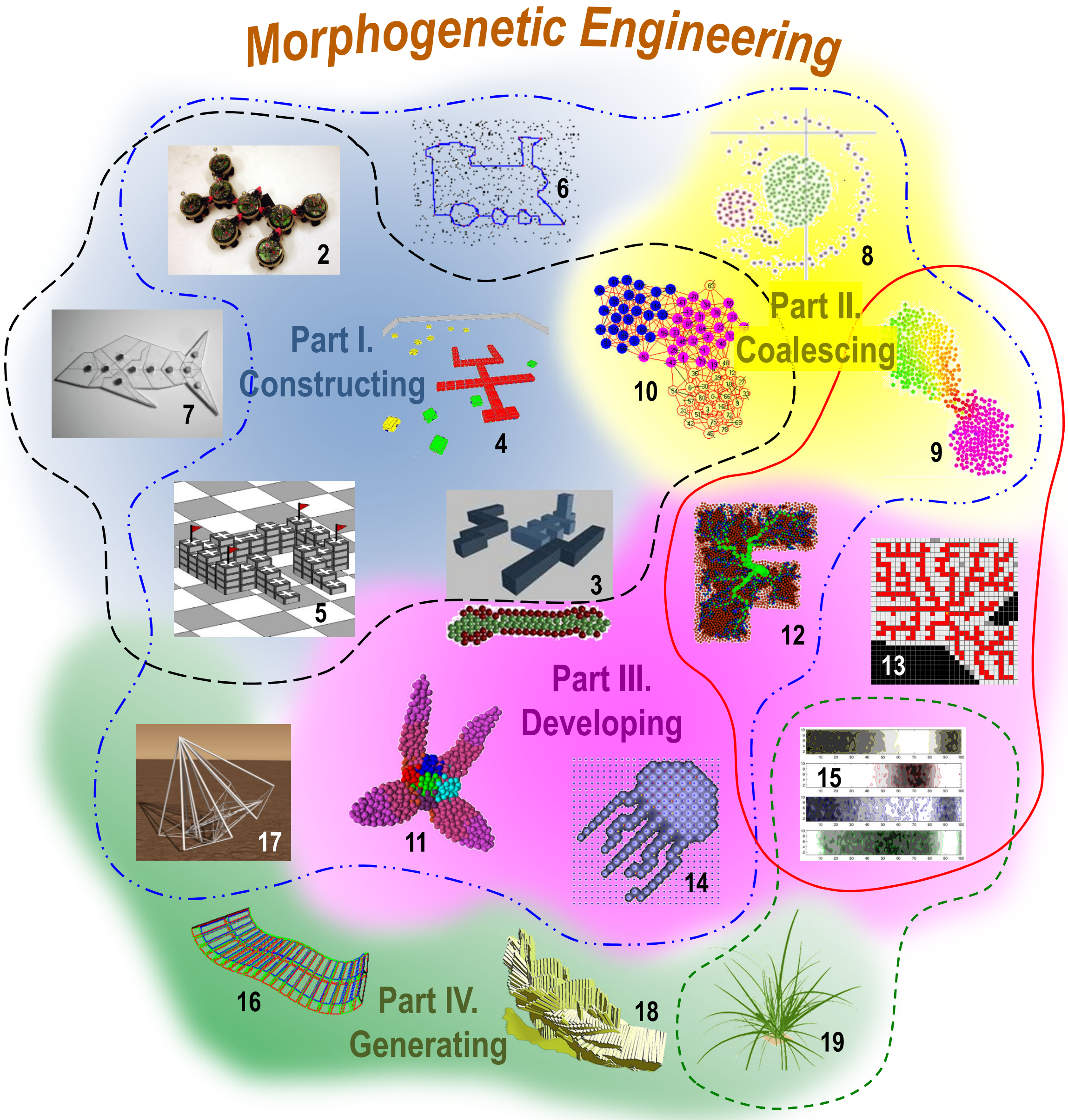
Morphogenetic Engineering
Part I. Constructing
Part II. Coalescing
Part III. Developing
Part IV. Generating
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

Figure 2: The Morphogenetic Engineering continent: We show here the 18 chapters of our book [40], after its introduction, taken as a sample illustrating the landscape of ME-minded or ME-contributing publications. They were schematically grouped into four main parts according to their typical dynamics. **Part I: Constructing** (*blue region*, Chapters 2-7): A few agents attach to each other or assemble blocks to build a precise structure. **Part II: Coalescing** (*yellow region*, Chapters 8-10): Many agents flock together into dense clusters creating shapes. **Part III: Developing** (*pink region*, Chapters 11-15 and part of Chapter 3): Agents divide or aggregate around an initially small core, forming patterns or organisms. **Part IV: Generating** (*green region*, Chapters 16-19): Components are iteratively transformed into architectures by repeated application of a grammar. Other, less distinguishing features that were not retained: (*dot-dashed blue line*) robotic models or applications, virtual or physical; (*big-dashed black line*) actual physical implementations with robots or mechanical components; (*solid red line*) emphasis on various patterns, textures, or symmetrical shapes, rather than complicated morphologies; (*small-dashed green line*) biological models based on real data, such as fruit fly and rye grass (2: O'Grady, Christensen & Dorigo [41], 3: Jin & Meng [36], 4: Liu & Winfield [42], 5: Werfel [43], 6: Arbuckle & Requicha [44], 7: Bhalla & Bentley [45], 8: Sayama [46], 9: Bai & Breen [47], 10: Nembrini & Winfield [48], 11: Doursat, Delile, Sánchez, Dordea, Fourquet & Kowaliw [49], 12: Beal [50], 13: Kowaliw & Banzhaf [51], 14: Cussat-Blanc, Pascalie, Mazac, Luga & Duthen [52], 15: Montagna & Viroli [53], 16: Spicher, Michel & Giavitto [54], 17: Lobo, Fernández & Vico [55], 18: von Mammen, Phillips, Davison, Jamniczky, Hallgrímsson & Jacob [56], 19: Verdenal, Combes & Escobar-Gutiérrez [57]).

Self-rearranging robotic parts In *self-reconfigurable modular robotics*, the issue is how an overall structure made of interconnected components can change its morphology. Components are relatively autonomous but not completely independent as in swarms (reviewed below). Rus and Vona [38] proposed one of the first self-reconfigurable modular systems. Their robotic unit was called a “crystalline atom”, a square or cubic device whose side faces could expand or contract, changing its effective volume. An interesting feature of their architecture was that it did not involve any rotational movement. They also developed a physical implementation of those robots, demonstrating simple self-reconfiguration and movement. Murata et al.'s M-TRAN system [39] is a well-known self-reconfigurable modular robot that can show very complex self-reconfiguration and locomotion tasks. The shape of their robotic unit is anisotropic with two rotational degrees of freedom, but its mechanical design was simpler and yet capable of more complex movements than other models such as crystalline atoms. The authors fully implemented the physical robots and successfully demonstrated intriguing behaviors, including snake-like locomotion, caterpillar track, quadruped walk, obstacle avoidance, and dynamic self-transformation among those shapes/behaviors.

Lipson and his lab [58, 59] have developed modular robots that can demonstrate complex behaviors including self-reconfiguration and self-reproduction. Their robotic unit, called "molecube", is a cubic module with only one rotational degree of freedom that swivels two halves of the cube around a diagonal axis. This makes the architecture of the system even simpler and more programmable than earlier models. Using this framework, they demonstrated self-reproduction and self-repair of simple morphologies, as well as evolutionary acquisition of novel forms in a simulated universe. Detweiler et al. [60] designed self-reconfigurable modular robot systems that consisted of a mixture of simple, passive structural modules and active robotic modules. While the robotic modules were still complicated and centrally controlled, the distinction between active constructors and passive structural materials is a promising approach that can be incorporated into other morphogenetic engineering systems. It also has some interesting similarities with von Neumann-style construction models, in which a small number of active parts can construct a much larger passive structure.

Self-assembling mobile robots In *swarm robotics*, the emphasis is shifted from the ensemble to the individual unit. Robots are self-propelled, run separately, and have the ability to autonomously dock to (and undock from) each other to form a larger robotic system. Baldassarre et al. [61] explored behavioral coordination mechanisms for a group of "swarm-bots" that were physically linked to each other. They elaborated robust, distributed coordination mechanisms using evolutionary search and neural networks. This is in contrast to the coordination mechanisms adopted in the modular robot models reviewed above (in which control mechanisms are generally designed and sent to each module in a top-down manner), making an important step toward truly bottom-up, self-organizing morphogenetic systems.

The "s-bots" designed by Dorigo and colleagues for their Swarmorph and Swarmanoid projects [62, 41] (Fig. 2:2) have the capacity to assemble into appropriate morphologies and operate as a single entity when physically connected to one another (using clamps). A low-level control logic allows inter-robot connections to be formed at particular angles, and a higher-level control logic dictates the sequence of these connections so as to form desired morphologies and make appropriate collective responses to different tasks. Arbuckle and Requicha [44] (Fig. 2:6) propose *active* *self-assembly*, a specific approach for constructing arbitrary shapes with swarms of identical and identically programmed robots. They identify important, open questions toward achieving full-fledged active self-assembly. Much of their discussion is centered on the fundamental problems of how to control a swarm to ensure that the structures it builds are self-repairing, and how to assess the performance of self-assembling swarms. Liu and Winfield [42] (Fig. 2:4) present distributed morphogenesis control strategies in a swarm of mobile robots able to autonomously disassemble and reassemble into different 3D symbiotic organisms. Their idea is to combine the advantages of swarm and self-reconfigurable robotic systems in order to investigate novel principles of evolution for "robotic organisms" from bio-inspired and evolutionary perspectives. Robots initially form a 2D planar structure, then the aggregated organism must lift itself to a 3D morphology, move and function as a macroscopic whole.

The grand challenge of engineering a self-organized morphogenetic robotic collective, whether considered "modular" or "swarm" (the distinction between the two tending to fade), was epitomized by the 5-year European projects Symbrion and Replicator [63], each involving a dozen institutional partners. Their ambition was to design new collective robotic systems using principles derived from natural symbiosis, such as demonstrated by bacteria and fungi. When it is advantageous to do so, these swarm robots would dynamically aggregate into one or many symbiotic organisms and collectively interact with the physical world, sharing energy and computational resources as a single "artificial-life-form". Considerable effort was dedicated to the design and manufacturing of novel robotic hardware, however, which eventually assumed the shape of large cubes equipped with sophisticated sensors and actuators, in particular for precise docking. Thus, as the other works described in this section, these platforms permit so far only a relatively small number of units arranged in exact formations such as chains and crosses.

Block constructions Social insects build large, complex structures, which emerge through the collective actions of many simple agents acting with no centralized control or preplanning. These natural systems inspire the research topic of *collective construction*, in which the goal is to engineer artificial systems that build in a similar way, characterized by swarms of simple robots producing desired structures. The work of Werfel [64, 43] (Fig. 2:5) investigates the design and implementation of such systems, for example based on LEGO Mindstorms. There, robots act independently, using only local information and no explicit communication; the system takes only a high-level design as input and is guaranteed to produce a structure matching that design.

Bhalla and Bentley [65, 45] (Fig. 2:7) examine how natural, nonrobotic self-assembly can be directly dictated by the morphology and properties of the components, and by environmental conditions. The process of self-assembly is equivalent to a physical computation, through the interaction and transformation of physically and chemically encoded information. They propose a three-level design approach, which specifies a set of simple self-assembly rules, then models and simulates these rules in software, then translates them to a physical system of plastic pieces with magnets. Their objective is to provide a bottom-up design methodology to create scalable self-assembling systems. In a similar vein, Virgo et al. [66] developed a physical model of self-replicating patterns made of simple passive plastic pieces with embedded magnets that float on an air-hockey table and agitated by wind coming from external fans. They created a very intricate design of the shape of plastic pieces, which allowed template-based self-replication of chains made of those pieces. Moreover, the inevitable stochasticity in the physical world naturally introduced "mutation" in those chains, demonstrating one of the first instances of evolutionary processes using macroscopic physical objects.

4.2 Category II: Coalescing

This second category of ME systems, which could also be subsumed under "Synchronizing" or "Swarming" (were it not for the ambiguity about the swarm's size and precision; see previous category), deals with a great number of mobile agents that are all simultaneously present (as opposed to being added one by one as in developing systems; see next category) and form a dense mass or network. Without literally attaching, they exhibit *heterogeneous* flocking behavior by staying near each other as they continuously move to maintain neighbor-to-neighbor communication, whether of the visual/infrared or wireless/radio type. Motion dynamics is often inspired by chemical concepts such as "pheromones" (as in ant colonies) or "morphogens" (as in multicellular tissues), which create concentration gradients that individual agents use to guide themselves by "chemotaxis". Together, they arrange themselves into fine-grained clusters that assume certain "fluid" yet stable shapes. For understandable reasons, most contributions in this category show simulated systems, as large-scale robotic swarms are still too costly to build with today's technology, and programmable flocking nano-particles are still unheard of. Some studies, however, are directly motivated by practical robotic applications, and remain close to their source.

Robotic agents Mamei et al. [67] proposed one of the first morphogenetic swarm models based on "coalescing" mechanisms. They assumed a finite number of autonomous mobile robots that could communicate indirectly by sensing morphogen gradients, and demonstrated in computer simulation that various spatial structures, such as circles, rings, lobes or polygons, could be produced in their framework. Their model uses only simple robots with minimal computational capabilities, which nicely illustrate the strengths of design principles based on crowds. Winfield and Nembrini [68, 48] (Fig. 2:10) describe decentralized control algorithms that link local wireless connectivity to low-level robot motion control for maintaining both swarm aggregation and connectivity, or "coherence". They investigate the potential of both first- and second-order connectivity information (i.e. around a node and this node's neighbors), showing that the number of *shared* neighbors acts as an adhesion parameter controlling the area coverage of the swarm, its taxis behavior toward a beacon, and its obstacle avoidance abilites. Adding more heterogeneity can also lead to an emergent segregation of sub-groups and the formation of specific axial morphologies.

Alonso-Mora et al. [69] present Display Swarm, another application of coalescing morphogenetic systems with a different flavor—aesthetics. In their model, each robotic module is given a predetermined goal location, and its trajectory is calculated a priori to be smooth and oscillation-free for visually pleasing reasons. Although not as bottom-up as other ME models, the fact that collective robots' trajectories are an objective in themselves can find important applications in the animation industry. Finally, in his Kilobot project, Rubenstein [70] precisely addresses the issue of system size. He notes that for technical reasons of hadware cost and complexity, most so-called robot "swarms" or "collectives" in fact contain at best a few tens of robots. To address this issue, he designed robot units that are made of cheap parts and are quick to assemble. His platform enables testing algorithms and control methods for groups of decentralized cooperating robots at least an order of magnitude larger than what exists today. His own research interest is also oriented toward morphological formations.

Software particles Sayama [8, 46] (Fig. 2:8) created an original *swarm chemistry* framework for morphogenetic artifacts that grow and self-organize in a fully decentralized manner. His swarms comprise multiple types of simple, interacting mobile particles with no elaborate connection or computation capabilities. They are essentially akin to flocks of birds ("boids") that were made heterogeneous and, from there, can produce surprisingly complex dynamic structures and behaviors. Features of emergent patterns are implicitly encoded, via interactive evolutionary design, into a set of kinetic parameter values, or "recipe". Diverse and robust morphological patterns can self-assemble by local information transmission, and self-repair by particle re-differentiation. The swarm chemistry model was also recently extended to evolutionary and 3D versions [71, 72]. Motivated by the ability of living cells to form specific structures, Bai and Breen [47] (Fig. 2:9) also investigated chemotaxis-inspired cellular primitives for self-organizing shape formation. In their case, cells emit a chemical into their environment and move in the direction of the gradient of the cumulative chemical field from other cells. This behavior is modeled by so-called "morphogenetic primitives" (MPs), software agents that may be programmed to self-organize into user-specified 2D shapes. Genetic programming is used to discover the particular chemical fields of individual MPs that are needed to produce macroscopic shapes from simple aggregation behaviors.

Programming matter, computing in space *Programmable matter* refers to various concepts and potential technologies that revolve around the physical instantiation of computing environments into a fine-grained substrate made of computing elements, also called "artificial atoms" (e.g. of a nanotechnological or biochemical sort), which could communicate by nearest neighbor interactions. It means creating "intelligent" materials with sensing/actuating and information-processing capabilities that would allow them to change their physical properties, e.g. shape, conductivity or reflection, as a function of external input. Toward the goals of programmable matter, Goldstein [73] introduced Claytronics, a 3D hardware technology that is intended to bring the objects of virtual reality into the real world to create a "synthetic reality". His system is composed of autonomous units containing a CPU and equipped with networking, locomotion, and docking mechanisms, so that they can organize themselves into the shape of any object and reproduce its surface and overall visual appearance. The endeavors of Claytronics are typically ME, as a major goal is morphology control without global motion planning but, instead, by local motion via displacement of holes in the 3D atom packing.

Similar to programmable matter, but from another perspective, "spatial computing" relies on a multitude of devices arranged or scattered in space, where communication between devices is strongly dependent on their geometric distance. Examples include sensor/actuator networks, mobile ad-hoc networks, or robotic swarms. In this context, Beal and Bachrach designed Proto [15], a domain-specific language that uses a continuous space abstraction called the *amorphous medium* [74] to view any spatial computer as an approximation of a space-time manifold with a computing device at every point. Information flows through this manifold with a bounded velocity, and each device has access to the recent past state of other devices within a nearby neighborhood. Proto primitives are operations on fields [75] and come in four types: pointwise computations, neighborhood operations, feedback operations, and restriction operations. Although not dedicated to ME, spatial computing is certainly conducive to morphogenetic tasks. The same medium can also be the site of processes more akin to growth than synchronization. This was the origin of "amorphous computing" [13], which is reviewed in the next category. Other spatial computing languages such as MGS [76] rather follow the paradigm of grammar-based, generative systems, which will be seen in the last category.

4.3 Category III: Developing

The inspiration for this category, equivalent to "Growing" or "Aggregating", is situated closer to cell-based models of biological morphogenesis. Here, systems start from a single agent or small group of agents, and grow to a relatively large size by repeated, yet differential, division or aggregation. Mechanisms underlying this development involve one or several biological features such as gene regulation, molecular signalling and chemotaxis. As a consequence, the resulting structures exhibit properties of biotic patterns or tissues, such as vascularization and segmentation, or entire organisms, such as arthropods or branched creatures. Potential applications range from synthetic biology and collective robotics to computer networks.

Evo-devo: from biology to computation One of the greatest challenges of biology is to create a generic model of multicellular development, in order to unify what Darwin called nature's "endless forms most beautiful" [4], and construe them as variants around a common theme. The variants are the unique genetic (and epigenetic) information of each species; the common theme is the developmental dynamics that this information guides and "parametrizes". While the Modern Synthesis focused most of the attention on selection, it is only during the past decade that analyzing and understanding variation by comparing the developmental processes of different species, at both embryonic and genomic levels, became a major concern of *evolutionary development*, or "evo-devo". To what extent are organisms also the product of self-organized physicochemical developmental processes not necessarily or always controlled by complex underlying genetics? Before and during the advent of genetics, the study of developmental structures had been pioneered by the "structuralist" school of theoretical biology, which can be traced back to Goethe, D'Arcy Thompson, and Waddington. Later, it was most actively pursued and defended by Kauffman [77, 78] and Goodwin [79, 80] under the banner of "self-organization", argued to be an even greater force than natural selection in the production of viable diversity. In particular, the strong morphological properties of biological organisms can be effectively captured by the paradigm of *positional information* introduced by Wolpert in the 1960's [81]. At an abstract level, the key idea is simply that cells must establish a long-range communication system that allows them to create different parts of the organism in different locations. It is inevitable that some form of positional information should be at work in multicellular organism development, embodied in various ways, be it through passive diffusion of morphogens spreading throughout the tissue and/or cell-to-cell intermediate-messenger signaling.

Now, looking at the full evolutionary *and* developmental picture should also be a primary concern of systems engineering and computer science when venturing into the new arena of autonomous, distributed architectures. Evolutionary computation (EC) techniques such as genetic algorithms or genetic programming, have just like their modern-synthesis model principally focused on selection and relied for the most part on "direct representations", i.e. parametric mappings from genotypes to phenotypes. One important goal of a new field of "artificial-life evo-devo" is to provide the computational foundation for a virtual re-engineering of the strongly morphogenetic complex systems spontaneously produced by nature, such as biological development. New EC avenues need to stress the importance of fundamental laws of developmental variations as a prerequisite to selection on the evolutionary time scale of artificial systems. Because indirect mappings promote compact encodings, modularity and combinatorial reuse, fine-grained multi-agent architectures similar to multicellular organisms might be in a unique position to provide the "solution-rich" space needed for successful selection and emergent innovation—through developmental modularity and composition.

Artificial development Putting in practice this theoretical ambition, a new class of systems collectively called *artificial embryogeny* (AE, sometimes "artificial ontogeny" or simply "artificial development") appeared recently, among others through the works and reviews of Bentley [82], Miller and Banzhaf [83], or Stanley and Miikkulainen [84]. Inspired by the biological development of multicellular organisms, AE systems realize the indirect mapping from genotype to phenotype via more or less complex developmental stages. Instead of coding directly for macroscopic features of the phenotype (the system), the parameters of the genotype code for microscopic features of the cells (the agents), i.e. their abilities to communicate, their propensity for motion, and their affinities for assembly with other cells. Like biological cells, the agents of an AE system generally share the same genotype, i.e. the same set of developmental and behavioral rules. Imitating cell division, differentiation and self-positioning, an agent spawns other agents, follows its own execution path within the common program—which may diverge from its neighbors' path depending on its position—and creates specific links with other nodes according to its fate.

Created independently, the *embryomorphic engineering* framework of Doursat et al. [85, 86, 49] (Fig. 2:11) also explores the causal and programmable link from genotype to phenotype at these two levels simultaneously: it consists of genetic engineering and functional shape engineering, based on a common playground made of a multitude of small agents capable of self-assembling into a particular organism. An embryomorphic model combines three key principles of multicellular biological development: chemical gradient diffusion (providing positional information to the agents), gene regulatory networks (triggering their differentiation into types, thus patterning), and cell division (creating structural constraints, hence reshaping). Therefore, agents are guided by the genetic instructions they carry, which parametrize and modulate the fundamental laws of mechanical-like assembly and biochemical-like signaling that they obey. Embryomorphic engineering is illustrated in various spaces: 2D/3D swarms with potential applications in collective robotics, and graph topologies with potential applications in peer-to-peer device/user networks.

Developmental animats In robotics (virtual or real), the integration of development with animation shows great promise toward faster and more innovative methodologies of automated design. Under the name of *brain-body co-evolution*, a recently emerging trend of evolutionary computation, it is argued that structure and function should not be predefined and optimized separately, but rather simultaneously as a whole. Like in animals, both the morphology *and* control of the organism should be encoded by the same genome, thus evolve and function together. From there, it is only a short and natural step to adding development to the equation and calling it "brain-body co-evo-devo". In this spirit, Sano and Sayama [87] designed the Wriggraph system, a model of kinetic graph-based virtual creatures whose behaviors are controlled by local rules that combine execution of program codes and simulation of reaction-diffusion systems. A unique feature of this model was that development and movement of the organism were uniformly represented without clear-cut separation. The authors created several illustrative examples, including the growth of a truss, simple self-replication, movement, target chasing and self-repair behaviors.

Taking after an earlier developmental model by Eggenberger [27], Joachimczak and Wrobel [88, 89] also elaborated a model of parallel co-evolution of development and motion control in multicellular, *soft-bodied animats.* Development is first guided by an artificial gene regulatory network (GRN), then embryos are converted into animat structures by connecting neighboring cells through elastic springs. Outer cells, which form the external envelope, are affected by drag forces in a fluid-like environment. Both the developmental program and locomotion controller are encoded into a single genomic sequence, which consists of regulatory regions and genes expressed into transcription factors and morphogens. A genetic algorithm is applied to evolve individuals able to swim in the simulated fluid, where the fitness depends on distance traveled during the evaluation phase. A similar work has been realized by Schramm et al. with worm-like 2D spring-mass animats [90]. Arguing that artificial developmental creatures are generally limited to a few hundred cells, Cussat-Blanc et al. [52] (Fig. 2:14) want to create larger and more complete organisms containing different organs and high-level functionalities. To this end, they propose a three-layer developmental framework called Cell2Organ. It is made of a chemical layer, where cells can divide and metabolize substrates, a hydrodynamics layer simulating substrate flows, and a physics layer that allows cells to change shape and organisms to move. Additionally, a new method based on L-systems without molecular morphogens is also introduced (see Section 4.4).

Morphogenetic patterning Other developmentally inspired models have essentially focused on *pattern formation*, with little or no motion of the agents. In physics, as mentioned in the introduction, patterns generally refer to statistically regular but stochastic motifs emerging in a quasi-continuous and initially homogeneous 2D or 3D substrate. By contrast, pattern formation of the morphogenetic type must also exhibit specific and *reproducible* spatial arrangements—even though it is still "painting" a fixed backdrop and not "sculpting" its own space, in the absence of mechanical interactions. For example, Montagna and Viroli [53] (Fig. 2:15) have examined how pattern formation in biological morphogenesis is regulated by gene expression. For them, modeling and simulating the developmental dynamics of living organisms at multiple scales can prove useful in the design of engineered products that manifest spatial self-organizing properties. To this aim, they describe a computational framework capable of supporting both the study of biological systems, such as patterning in the *Drosophila* morphogenesis, and the design of artificial systems that can autonomously develop a spatial structure, such as in *pervasive computing and service* scenarios. These involve sensing and actuating devices (RFID tags, PDAs, GPS, etc.) that densely populate our everyday environment with digital information about users and locations, in tightly integration with the Web. A pervasive service can then be "injected" like a morphogen in one location and "diffuse" around, interacting and competing with other services in a context-dependent manner inside each node of the network.

The goal of the *amorphous computing* initiative launched by Abelson and colleagues [13] is also to identify organizational principles and create programming technologies for obtaining intentional, pre-specified behavior from the cooperation of a myriad of unreliable parts arranged in unknown, irregular, and time-varying ways. Under this umbrella, the Growing Point Language (GPL) designed by Coore [91] from a botanical metaphor expresses topological structures in terms of points that build a pattern by incrementally passing the locus of activity through space. "Tropisms" attract or repel the motion of growing points through simulated chemical signals. The Origami Shape Language (OSL) by Nagpal [92, 93] is another amorphous computing language for constructively describing global shapes as a sequence of straight folds performed on a square sheet of paper, with no cuts or glue (the folds are always flat and 3D structures are created by opening the flat structure in the end). Macroscopic fold lines are compiled into microscopic particle-based programs such that, given an initial identification of edges, the local interactions will compute the desired folds, eventually producing the specified shape. Both GPL and OSL are robust and tolerant of changes in their conditions of execution, with distorted initial conditions producing a similarly distorted final pattern.

Another subclass of morphogenetic patterning concerns *branching growth* systems, among which vasculogenesis is a prime example. For Beal [50] (Fig. 2:12), the engineering of development poses fundamentally different challenges than static designs. Growth certainly offers much greater potential for adaptation to changes in the environment, but a growing system must also be capable of surviving every intermediate stage, despite internal stresses due to the uneven scaling of its subsystems (e.g. muscles and bones must grow together). His work considers a new engineering approach based on *functional blueprints*, under which a system is specified in terms of desired performance and means of incrementally correcting deficiencies by following trajectories in a viability space. It is demonstrated on models of tissue growth and vasculogenesis. Kowaliw [94, 51] (Fig. 2:13) also considers the means by which morphogenetic growth can lead to complex systems design. The evolvability of a difficult design space can be enhanced through mechanisms such as regularities and *adaptive feedback*. He adapts a concrete example to a simplified simulation of vasculogenesis. Here, the focus is on the feedback mechanism: by requiring viability during growth, the organism gains awareness of its interim success. A "local fitness" is used as a drive during the development of the final design. This approach is shown to improve the efficiency of the learner, and to eliminate the problem's hardness associated with the complexity of the environment.

4.4 Category IV: Generating

In this last category, in effect very similar to development but with a different origin and formalism, morphogenetic systems are *generated* by successive transformations of components in 2D or 3D space, based on on substitution-based rewrite rules, which have the effect of inserting (or deleting) components. These rules are formally expressed as "grammars" designed by hand or evolved. The resulting architectures have potential applications as diverse as natural computing, robotics, computer graphics or plant biology.

Biologically inspired grammars Probably the earliest and most famous class of geometric generative models, called *L-systems*, was designed by the biologist A. Lindenmayer to give a formal description of the cellular development of filamentous plants [95]. It is a powerful formalism that represents systems by strings of symbols with a hierarchical structure, in which bracketed groups of symbols corresponds to parts, embedded groups to subparts, and contiguous groups to neighboring parts. The morphogenetic development of the system is carried out by parallel rewrite rules that select symbols to be replaced by new groups. Driven in part by the work of Prusinkiewicz [96], this framework has become popular in the modeling and simulation of growth processes, especially in botany. As a consequence, the graphical interpretation of L-systems generally (and quite literally) produces tree-like, branching structures, which also make them particularly suited to the vascular and respiratory systems of animal models. To diversify the morphogenetic abilities of L-systems beyond self-similar fractal topologies, rewrite rules can also be made probabilistic and/or context-dependent. Verdenal et al. [57] (Fig. 2:19) have applied L-systems to forage grass morphology, which emerges from the combination of many interrelated dynamical processes. Although a plant's architecture contains an intrinsic, genetically determined part, its morphogenesis also exhibits very high plasticity with respect to environmental conditions. This could be mediated by a self-regulatory process, e.g. where leaf length is affected by preceding leaves. The authors present a functional-structural 3D model of ryegrass based on this hypothesis, showing that architectural development can result from a collaboration between genetic programmability and self-organization, instead of a centralized control of each trait.

Spicher et al. [54] (Fig. 2:16) advocate a *domain-specific language* approach to overcome the difficulties of modeling and simulating morphogenetic processes. To this aim, they present an experimental programming language called MGS. The declarative approach of MGS is based on the notion of *topological collection*, which originates from algebraic topology and arises naturally when trying to model "dynamical systems with a dynamic structure" (DS^2) [97]. The evolution function of such systems is specified by *transformations* made of sets of rewrite rules, where each rule defines a local interaction. These notions allow to easily describe a variety of models in a uniform setting [98] focusing on the interacting parts of the model. In this sense, it also belongs to the spatial computing paradigm mentioned at the end of Section 4.3.

Graph and swarm grammars With *graph automata*, Tomita et al. [99] proposed a new class of spatio-temporal developmental processes using rewrite rulesets applied to 3-regular graphs. They demonstrated that complex developmental processes such as self-reproduction of Turing machines can be expressed by simple, straightforward formalism. It was among the first models to extend discrete lattice-based dynamical systems (such as cellular automata) to a more generative, topologically flexible domain. Sayama [100] proposed *generative network automata* (GNA), another graph rewrite framework for modeling generative processes, similar to graph automata. GNA is unique in that it can model any kind of network topologies, and also that it formulates dynamical processes in two separate algorithmic phases: extraction and replacement. He showed that both traditional dynamical networks and more recent complex network growth models could be represented in this framework. Computational exploration of different classes of model behaviors were also conducted.

Von Mammen et al. [101, 56] (Fig. 2:18) have introduced *swarm grammars* to model swarm-based generative and developmental systems. In swarms, as in complex systems, large numbers of individuals locally interact and form nonlinear, dynamic interaction networks. Ants, wasps and termites, for instance, are natural swarms whose individual and group behaviors have been evolving over millions of years. The emergent effectiveness of their behaviors is most striking in their intricate nest constructions. Swarm-grammar-based computational simulations capture the corresponding principles of agent-based, decentralized, self-organizing models. Rieffel et al. [102] proposed a framework for automatically creating *tensegrity structures*, where the morphological design of a structure was represented in a generative form so that evolutionary search could easily discover and evolve physically meaningful patterns. Tensegrity is suitable for designing "soft robots", i.e. robots that exhibit flexibility and adaptive balance in their morphology. The authors demonstrated physical realization of automatically generated designs, indicating the possible connections between simulated and physical worlds.

Evolutionary grammars The combination of morphogenetic generative systems and evolution was best exemplified by the *automated generative design* system of Hornby and Pollack [103]. From the viewpoint of evolutionary computation, a generative transformation is considered an "indirect representation", as it creates a mapping from the genotype to the phenotype. Despite the apparent complexification that they bring to evolutionary algorithms, indirect representations have recently come under close scrutiny as they are also thought to allow long-term evolvability via accumulated elaborations and emergent, unplanned features—hence open-ended and more scalable genotypes. In contrast, “direct representations” are not capable of open-ended elaboration because they restrict genotype space to predefined features. Accordingly, in [103], it is not the objects (here, tables) but their grammar-based designs which are evolved, creating an order of magnitude more parts than previous generative systems. The authors show that a generative representation can achieve better fitness (here, the tables’ proportions and physical robustness) by reusing successful elements from the design space and allowing large-scale, yet viable mutations in the phenotype.

Lobo and Vico [104, 55] (Fig. 2:17) have also worked on evolutionary morphogenetic algorithms of the generative kind to automate the design of “organic” morphologies and controllers with the goal of solving certain functional problems. They point out that evolution has shaped a remarkable diversity of multicellular organisms, whose complex forms are self-made by robust developmental processes, and that this “evo-devo” combination can inspire novel computational methodologies to overcome the scalability problems of classical top-down design. In their systems, performance is tested on optimization challenges not based explicitly on morphological and structural constraints, however, but on solving abilities and functional efficiency, such as *behavior-finding* in animated structures (e.g. moving along a path).

5 Perspectives

Biological organisms are the pure products of “undesigned evolution” (UDE) through random variations (expressed via the physical constraints of self-organization) and nonrandom natural selection. By contrast, artificial structures will (hopefully) always possess a causal link originating from their human makers, while at the same time this link should become less and less visible. Traditional engineering has always followed a “directly designed construction” paradigm, in which architects plan and build entire systems top-down. Morphogenetic engineering proposes a gradual shift toward the biological paradigm, via stages that could be called “meta-designed development” (MDD) and “meta-designed evolution” (MDE)—stopping short of pure UDE. In MDD, meta-designers will focus on creating local mechanisms that allow small agents or components to assemble, coalesce, grow, or generate architectures by themselves. In MDE, even more “disengaged” meta-designers will only create laws of variation and selection of the parameters of these local mechanisms (or the mechanisms themselves), prepare a few primitive ancestor systems, then step back to let evolution and development invent the rest.

In sum, ME consists of designing a *programmable* and *reproducible* two-way indirect mapping between the local rules of self-assembly followed by the elementary agents at the microscopic level, on the one hand (the genotype G), and the collective structure and function of the system at the macroscopic level, on the other hand (the phenotype P). Calculating the transformation from G to P corresponds to developing an organism, while solving the inverse problem of finding an appropriate G given a desired P (or family of P's) is the challenge of MDD and MDE, which can itself be goal-oriented or open-ended, or a mix of the two.

Whether in 2D/3D physical devices, in software or in techno-social networks, emergent architectures and decentralized automation create exciting new perspectives. To obtain novel and unplanned behavior, engineers, the old enforcers of passive matter, should now *guide* bottom-up interactions among a multitude of active components by endowing them with certain rules and parameters—or, better still, with some metaheuristic (such as learning or evolution) giving them the generic capability to fine-tune or acquire these rules and parameters by themselves. Morphogenetic engineers need to design for *emergence*, i.e. for systems that fundamentally and continually adapt and evolve. The appeal of ME resides in the many beneficial "self-x" properties that this new attitude could bring, by improving, complementing or even replacing human-led design and planning efforts. For example, it could allow remote operations in hostile places, faster organization without the usual delays tied to a central command node, greater robustness and reactivity to new events or environments, and better scalability if the system needs to grow.

More broadly, as we have shown in the review above, ME endeavors are closely related and intersecting with those of many other innovative fields that have emerged during the last decade, essentially during the 2000's, and could be subsumed altogether under the overarching concept of *Engineering and Control of Self-Organization* (ECSO). With morphogenetic engineering, we hope to be contributing to a larger movement of emerging cross-disciplinary topics that all attempt to solve, at their core, the paradoxes of "organizing self-organization" and "planning autonomy" in various ways—and not only toward shapes or architectures. In recent years, it has become increasingly apparent that a whole new generation of innovative researchers and projects cannot find an appropriate home in traditional venues—established conferences or journals, even university departments or schools—as they are torn between *scientific domains* focused on the observation and models of natural complex systems (in which they appear too "artificial" and disconnected from "real data") and *engineering domains* more interested in top-down design and optimization of perfectly reliable "complicated systems" than complexity per se (in which these meta-designs appear too "soft" or "bio-inspired" and not sufficiently "proven"). ECSO should therefore be a unifying pole between two big families: complex systems and engineeering, which are traditionally not communicating, yet have witnessed the rise of a new, unofficial intersection populated with numerous original ideas (references only indicative, not exhaustive):

- Amorphous Computing [13, 91, 93, 64], Spatial Computing [14, 15, 16]
- Artificial Embryogeny, Artificial Ontogeny, Artificial Development [27, 82, 83, 84, 94]
- Complex Systems Engineering [18]
- Embryomorphic Engineering [85, 86, 105]
- Emergent Neural Dynamics, Neuroevolution
- Generative Systems, Grammars, Rewriting Systems [95, 96, 103]
- **Morphogenetic Engineering** [40]
- Morphogenetic Robotics, Evo-Devo Robotics, Generative Robotics [63, 37]
- Multi-Agent Systems, Agent-Based Modeling
- Organic Computing [106, 17]
- Pervasive Adaptation, Autonomic Computing [107], Autonomic Networking, Emergent Engineering
- Pervasive Computing, Ubiquitous Computing [108], Ambient Intelligence [109]
- Programmable Matter [73]
- Swarm Robotics, Collective Robotics, Swarm Intelligence [32, 67, 62, 41], Swarm Chemistry [8, 46]
- Synthetic Biology, Biological and Chemical IT, Artificial Chemistry
- Unconventional Computing, Natural Computing [12, 110]

Within the larger ECSO federation, morphogenetic engineering, for its part, focuses on the strong *architectural* and complex functional properties of systems, and how these properties can be influenced or programmed at the microlevel.

References

[1] Doursat, R., Sayama, H., and Michel, O. (2012), Morphogenetic engineering: Reconciling self-organization and architecture. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 1–25, Springer-Verlag.

[2] Ball, P. (1999), The Self-Made Tapestry. Oxford University Press.

[3] Coen, E. (2000), The Art of Genes. Oxford University Press.

[4] Carroll, S. B. (2005), Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom. W. W. Norton.

[5] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999), Swarm intelligence: from natural to artificial systems. Oxford University Press.

[6] Ulieru, M. and Doursat, R. (2011), Emergent engineering: a radical paradigm shift. *International Journal of Autonomous and Adaptive Communications Systems* **4**(1), 39–60.

[7] Tanenbaum, A. S. and van Steen, M. (2002), Distributed Systems: Principles and Paradigms. Prentice Hall.

[8] Sayama, H. (2009), Swarm chemistry. *Artificial Life* **15**(1), 105–114.

[9] Bullock, S., Ladley, D., and Kerby, M. (2012), Wasps, termites, and waspmites: Distinguishing competence from performance in collective construction. *Artificial Life* **18**(3), 267–290.

[10] Endy, D. (2005), Foundations for engineering biology. *Nature* **438**(7067), 449–453.

[11] Knight, T. (2003), Idempotent vector design for standard assembly of biobricks. Tech. rep., DTIC Document.

[12] Stepney, S., Braunstein, S. L., Clark, J. A., Tyrrell, A., Adamatzky, A., Smith, R. E., Addis, T., Johnson, C., Timmis, J., Welch, P., Milner, R., and Partridge, D. (2005), Journeys in non-classical computation i: A grand challenge for computing research. *International Journal of Parallel, Emergent and Distributed Systems* **20**(1), 5–19.

[13] Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight Jr, T. F., Nagpal, R., Rauch, E., Sussman, G. J., and Weiss, R. (2000), Amorphous computing. *Communications of the ACM* **43**(5), 74–82.

[14] Giavitto, J. L. and Michel, O. (2002), The topological structures of membrane computing. *Fundamenta Informaticae* **49**(1-3), 123–145.

[15] Beal, J. and Bachrach, J. (2006), Infrastructure for engineered emergence on sensor/actuator networks. *Intelligent Systems, IEEE* **21**(2), 10–19.

[16] Beal, J., Dulman, S., Usbeck, K., Viroli, M., and Correll, N. (2012), Organizing the aggregate: Languages for spatial computing. *Computing Research Repository* **abs/1202.5509**.

[17] Würtz, R. P. (2008), Organic computing. Springer.

[18] Minai, A. A., Braha, D., and Bar-Yam, Y. (2006), Complex engineered systems: Science meets technology. In Complex Engineered Systems: Science Meets Technology (D. Braha, Y. Bar-Yam, and A. A. Minai, eds.), chap. Complex engineered systems: A new paradigm, pp. 1–21, Springer.

[19] Holland, J. H. (1992), Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

[20] Gierer, A. and Meinhardt, H. (1972), A theory of biological pattern formation. *Biological Cybernetics* **12**(1), 30–39.

[21] Pearson, J. E. (1993), Complex patterns in a simple system. *Science* **261**(5118), 189–192.

[22] 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**(6), 1226–1229.

[23] Barabási, A. L. and Albert, R. (1999), Emergence of scaling in random networks. *Science* **286**(5439), 509–512.

[24] Newman, M. E. J. (2006), Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* **103**(23), 8577–8582.

[25] Barrat, A., Barthélemy, M., and Vespignani, A. (2008), Dynamical processes on complex networks. Cambridge University Press.

[26] Rothemund, P. W. K. (2006), Folding dna to create nanoscale shapes and patterns. *Nature* **440**(7082), 297–302.

[27] Eggenberger, P. (1997), Evolving morphologies of simulated 3d organisms based on differential gene expression. In Proceedings of the Fourth European Conference on Artificial Life, pp. 205–213.

[28] Brooks, R. (1986), A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of* **2**(1), 14–23.

[29] Pfeifer, R., Bongard, J., and Grand, S. (2006), How the body shapes the way we think: a new view of intelligence. The MIT Press.

[30] Sims, K. (1994), Evolving 3d morphology and behavior by competition. *Artificial life* **1**(4), 353–372.

[31] Lipson, H. and Pollack, J. B. (2000), Automatic design and manufacture of robotic lifeforms. *Nature* **406**(6799), 974–978.

[32] Dorigo, M. and Stützle, T. (2004), Ant Colony Optimization. The MIT Press.

[33] Kennedy, J. and Eberhart, R. (1995), Particle swarm optimization. In Neural Networks, 1995. Proceedings, IEEE International Conference on, vol. 4, pp. 1942–1948, IEEE.

[34] Hebb, D. O. (1949), The organization of behavior. John Wiley and Sons.

[35] Hopfield, J. J. (1982), Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* **79**(8), 2554.

[36] Jin, Y. and Meng, Y. (2012), Morphogenetic robotics: A new paradigm for designing self-organizing, self-reconfigurable and self-adaptive robots. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 61–87, Springer-Verlag.

[37] Jin, Y. and Meng, Y. (2011), Morphogenetic robotics: An emerging new field in developmental robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **41**(2), 145–160.

[38] Rus, D. and Vona, M. (2000), A physical implementation of the self-reconfiguring crystalline robot. In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, vol. 2, pp. 1726–1733, IEEE.

[39] Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S. (2002), M-tran: Self-reconfigurable modular robotic system. *Mechatronics, IEEE/ASME Transactions on* **7**(4), 431–441.

[40] Doursat, R., Sayama, H., and Michel, O., eds. (2012), Morphogenetic Engineering: Toward Programmable Complex Systems. Understanding Complex Systems, Springer-Verlag.

[41] O'Grady, R., Christensen, A. L., and Dorigo, M. (2012), Swarmorph: Morphogenesis with self-assembling robots. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 27–60, Springer-Verlag.

[42] Liu, W. and Winfield, A. F. T. (2012), Distributed autonomous morphogenesis in a self-assembling robotic system. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 89–113, Springer-Verlag.

[43] Werfel, J. (2012), Collective construction with robot swarms. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 115–140, Springer-Verlag.

[44] Arbuckle, D. J. and Requicha, A. A. G. (2012), Issues in self-repairing robotic self-assembly. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 141–155, Springer-Verlag.

[45] Bhalla, N. and Bentley, P. J. (2012), Programming self-assembling systems via physically encoded information. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 157–188, Springer-Verlag.

[46] Sayama, H. (2012), Swarm-based morphogenetic artificial life. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 191–208, Springer-Verlag.

[47] Bai, L. and Breen, D. (2012), Chemotaxis-inspired cellular primitives for self-organizing shape formation. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 209–237, Springer-Verlag.

[48] Winfield, A. F. T. and Nembrini, J. (2012), Emergent swarm morphology control of wireless networked mobile robots. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 239–271, Springer-Verlag.

[49] Doursat, R., Snchez, C., Dordea, R., Fourquet, D., and Kowaliw, T. (2012), Embryomorphic engineering: Emergent innovation through evolutionary development. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 275–311, Springer-Verlag.

[50] Beal, J. (2012), Functional blueprints: An approach to modularity in grown systems. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 313–329, Springer-Verlag.

[51] Kowaliw, T. and Banzhaf, W. (2012), Mechanisms for complex systems engineering through artificial development. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 331–351, Springer-Verlag.

[52] Cussat-Blanc, S., Pascalie, J., Mazac, S., Luga, H., and Duthen, Y. (2012), A synthesis of the cell2organ developmental model. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 353–381, Springer-Verlag.

[53] Montagna, S. and Viroli, M. (2012), A computational framework for multilevel morphologies. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 383–405, Springer-Verlag.

[54] Spicher, A., Michel, O., and Giavitto, J.-L. (2012), Interaction-based modeling of morphogenesis in mgs. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 409–440, Springer-Verlag.

[55] Lobo, D., Fernndez, J. D., and Vico, F. J. (2012), Behavior-finding: Morphogenetic designs shaped by function. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 441–472, Springer-Verlag.

[56] von Mammen, S., Phillips, D., Davison, T., Jamniczky, H., Hallgrmsson, B., and Jacob, C. (2012), Swarm-based computational development. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 473–499, Springer-Verlag.

[57] Verdenal, A., Combes, D., and Escobar-Gutirrez, A. (2012), Programmable and self-organised processes in plant morphogenesis: The architectural development of ryegrass. In Morphogenetic Engineering: Toward Programmable Complex Systems (R. Doursat, H. Sayama, and O. Michel, eds.), Understanding Complex Systems, pp. 501–517, Springer-Verlag.

[58] Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005), Self-reproducing machines. *Nature* **435**(7039), 163–164.

[59] Grouchy, P. and Lipson, H. (2012), Evolution of self-replicating cube conglomerations in a simulated 3d environment. In Artificial Life, vol. 13, pp. 59–66.

[60] Detweiler, C., Vona, M., Yoon, Y., Yun, S., and Rus, D. (2007), Self-assembling mobile linkages. *Robotics & Automation Magazine, IEEE* **14**(4), 45–55.

[61] Baldassarre, G., Parisi, D., and Nolfi, S. (2006), Distributed coordination of simulated robots based on self-organization. *Artificial Life* **12**(3), 289–311.

[62] Christensen, A. L., O'Grady, R., and Dorigo, M. (2007), Morphology control in a multirobot system. *Robotics & Automation Magazine, IEEE* **14**(4), 18–25.

[63] Kernbach, S., Meister, E., Schlachter, F., Jebens, K., Szymanski, M., Liedke, J., Laneri, D., Winkler, L., Schmickl, T., Thenius, R., et al. (2008), Symbiotic robot organisms: Replicator and symbrion projects. In Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 62–69, ACM.

[64] Werfel, J. and Nagpal, R. (2006), Extended stigmergy in collective construction. *Intelligent Systems, IEEE* **21**(2), 20–28.

[65] Bhalla, N., Bentley, P. J., and Jacob, C. (2007), Mapping virtual self-assembly rules to physical systems. *Unconventional Computing 2007* p. 167.

[66] Virgo, N., Fernando, C., Bigge, B., and Husbands, P. (2012), Evolvable physical self-replicators. *Artificial life* **18**(2), 129–142.

[67] Mamei, M., Vasirani, M., and Zambonelli, F. (2004), Experiments of morphogenesis in swarms of simple mobile robots. *Applied Artificial Intelligence* **18**(9-10), 903–919.

[68] Winfield, A., Harper, C., and Nembrini, J. (2005), Towards dependable swarms and a new discipline of swarm engineering. *Swarm robotics* pp. 126–142.

[69] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., and Beardsley, P. (2011), Multi-robot system for artistic pattern formation. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 4512–4517, IEEE.

[70] Rubenstein, M., Ahler, C., and Nagpal, R. (2012), Kilobot: A low cost scalable robot system for collective behaviors. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 3293–3298, IEEE.

[71] Sayama, H. (2011), Seeking open-ended evolution in swarm chemistry. In Artificial Life (ALIFE), 2011 IEEE Symposium on, pp. 186–193, IEEE.

[72] Sayama, H. (2012), Morphologies of self-organizing swarms in 3d swarm chemistry. In Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference, pp. 577–584, ACM.

[73] Goldstein, S., Campbell, J., and Mowry, T. (2005), Programmable matter. *Computer* **38**(6), 99–101.

[74] Beal, J. (2005), Programming an amorphous computational medium. *Unconventional Programming Paradigms* pp. 97–97.

[75] Hammarlund, P. and Lisper, B. (1993), On the relation between functional and data parallel programming languages. In Proceedings of the conference on Functional programming languages and computer architecture, pp. 210–219, ACM.

[76] Spicher, A., Michel, O., and Giavitto, J. (2010), Declarative mesh subdivision using topological rewriting in mgs. *Graph Transformations* pp. 298–313.

[77] Kauffman, S. A. (1993), The origins of order: Self organization and selection in evolution. Oxford University Press.

[78] Kauffman, S. A. (2008), Reinventing the sacred: A new view of science, reason, and religion. Basic Books.

[79] Goodwin, B. C. (1994), How the leopard changed its spots: the evolution of complexity. Scribner.

[80] Webster, G. and Goodwin, B. C. (1996), Form and transformation: generative and relational principles in biology. Cambridge Univ Press.

[81] Wolpert, L. (1969), Positional information and the spatial pattern of cellular differentiation. *Journal of theoretical biology* **25**(1), 1–47.

[82] Bentley, P. and Kumar, S. (1999), Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, pp. 35–43, Morgan Kaufmann.

[83] Miller, J. F. and Banzhaf, W. (2003), Evolving the program for a cell: from french flags to boolean circuits. *On Growth, Form and Computers* pp. 278–301.

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

[85] Doursat, R. (2006), The growing canavas of biological development: multiscale pattern generation on an expanding lattice of gene regulatory networks. *InterJournal: Complex Systems* **1809**.

[86] Doursat, R. (2008), Organically grown architectures: creating decentralized, autonomous systems by embryomorphic engineering. In Organic computing (R. P. Würtz, ed.), Understanding complex systems, pp. 167–199, Springer.

[87] Sano, K. and Sayama, H. (2006), Wriggraph: A kinetic graph model that uniformly describes ontogeny and motility of artificial creatures. In Artificial life X: proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, vol. 10, p. 77, The MIT Press.

[88] Joachimczak, M. and Wróbel, B. (2008), Evo-devo in silico: a model of a gene network regulating multicellular development in 3d space with artificial physics. In Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 297–304.

[89] Joachimczak, M., Kowaliw, T., Doursat, R., and Wrobel, B. (2012), Brainless bodies: Controlling the development and behavior of multicellular animats by gene regulation and diffusive signals. In Artificial Life 13: Proceedings of the Thirteenth International Conference on the Simulation and Synthesis of Living Systems, pp. 349–356.

[90] Schramm, L., Jin, Y., and Sendhoff, B. (2011), Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. *Advances in Artificial Life. Darwin Meets von Neumann* pp. 27–34.

[91] Coore, D. N. (1999), Botanical computing: a developmental approach to generating interconnect topologies on an amorphous computer. Ph.D. thesis, MIT.

[92] Nagpal, R. (2001), Programmable self-assembly: constructing global shape using biologically-inspired local interactions and origami mathematics. Ph.D. thesis, MIT.

[93] Nagpal, R. (2002), Programmable self-assembly using biologically-inspired multiagent control. In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, pp. 418–425, ACM.

[94] Kowaliw, T., Grogono, P., and Kharma, N. (2004), Bluenome: A novel developmental model of artificial morphogenesis. In Genetic and Evolutionary Computation GECCO '04, pp. 93–104, Springer.

[95] Prusinkiewicz, P. and Lindenmayer, A. (1991), The algorithmic beauty of plants. Springer-Verlag.

[96] Coen, E., Rolland-Lagan, A. G., Matthews, M., Bangham, J. A., and Prusinkiewicz, P. (2004), The genetics of geometry. *Proceedings of the National Academy of Sciences of the United States of America* **101**(14), 4728–4735.

[97] Giavitto, J.-L., Godin, C., Michel, O., and Prusinkiewicz, P. (2002), Modelling and Simulation of biological processes in the context of genomics, chap. "Computational Models for Integrative and Developmental Biology". Hermes.

[98] Spicher, A., Michel, O., and Giavitto, J. (2011), Interaction-based simulations for integrative spatial systems biology. *Understanding the Dynamics of Biological Systems* pp. 195–231.

[99] Tomita, K., Kurokawa, H., and Murata, S. (2002), Graph automata: natural expression of self-reproduction. *Physica D: Nonlinear Phenomena* **171**(4), 197–210.

[100] Sayama, H. (2007), Generative network automata: A generalized framework for modeling complex dynamical systems with autonomously varying topologies. In Artificial Life, 2007. ALIFE'07. IEEE Symposium on, pp. 214–221, IEEE.

[101] von Mammen, S. and Jacob, C. (2007), Genetic swarm grammar programming: Ecological breeding like a gardener. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 851–858, IEEE.

[102] Rieffel, J., Valero-Cuevas, F., and Lipson, H. (2009), Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures* **87**(5), 368–379.

[103] Hornby, G. S. and Pollack, J. B. (2001), The advantages of generative grammatical encodings for physical design. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, vol. 1, pp. 600–607, IEEE.

[104] Lobo, D. and Vico, F. J. (2010), Evolutionary development of tensegrity structures. *Biosystems* **101**(3), 167–176.

[105] Doursat, R. (2009), Facilitating evolutionary innovation by developmental modularity and variability. In Proceedings of the 11th annual conference on genetic and evolutionary computation (GECCO), pp. 683–690, ACM.

[106] Malsburg, C. (2008), Organic computing. In Organic computing (R. P. Würtz, ed.), Understanding complex systems, chap. The organic future of information technology, pp. 7–24, Springer.

[107] Kephart, J. O. and Chess, D. M. (2003), The vision of autonomic computing. *Computer* **36**(1), 41–50.

[108] Weiser, M. (1993), Some computer science issues in ubiquitous computing. *Communications of the ACM* **36**(7), 75–84.

[109] Marzano, S. and Aarts, E. (2003), The New Everyday View on Ambient Intelligence. Uitgeverij 010 Publishers.

[110] Nunes de Castro, L. N. (2006), Fundamentals of natural computing: basic concepts, algorithms, and applications. CRC Press.