Morphogenetic Engineering: Reconciling Self-Organization and Architecture

Article · December 2012

DOI: 10.1007/978-3-642-33902-8_1

In *Morphogenetic Engineering: Toward Programmable Complex Systems* (2012), R. Doursat, H. Sayama & O. Michel, eds.: pp. xx-xx. *"Understanding Complex Systems" Series*, Springer-Verlag. ***In press***.

Chapter 1: Introduction
Morphogenetic Engineering: Reconciling Self-Organization and Architecture

René Doursat, Hiroki Sayama and Olivier Michel

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? This book is the first initiative of its kind toward establishing a new field of research, *Morphogenetic Engineering*, to explore the modeling and implementation of "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 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 dynam-

R. Doursat: Complex Systems Institute, Paris Ile-de-France (ISC-PIF), CNRS & Ecole Polytechnique, 75005 Paris, France (rene.doursat@polytechnique.edu). H. Sayama: Collective Dynamics of Complex Systems (CoCo), Department of Bioengineering, Binghamton University SUNY, Binghamton, NY 13902, US (sayama@binghamton.edu). O. Michel: Algorithmic, Complexity and Logic Laboratory (LACL), CNRS, Department of Computer Science, Université de Créteil, 94010 Créteil, France (michel@u-pec.fr). This chapter is an extended version of references [16, 17].

ics. 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 [12, 11]. 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 [8] 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.

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* [45] 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,

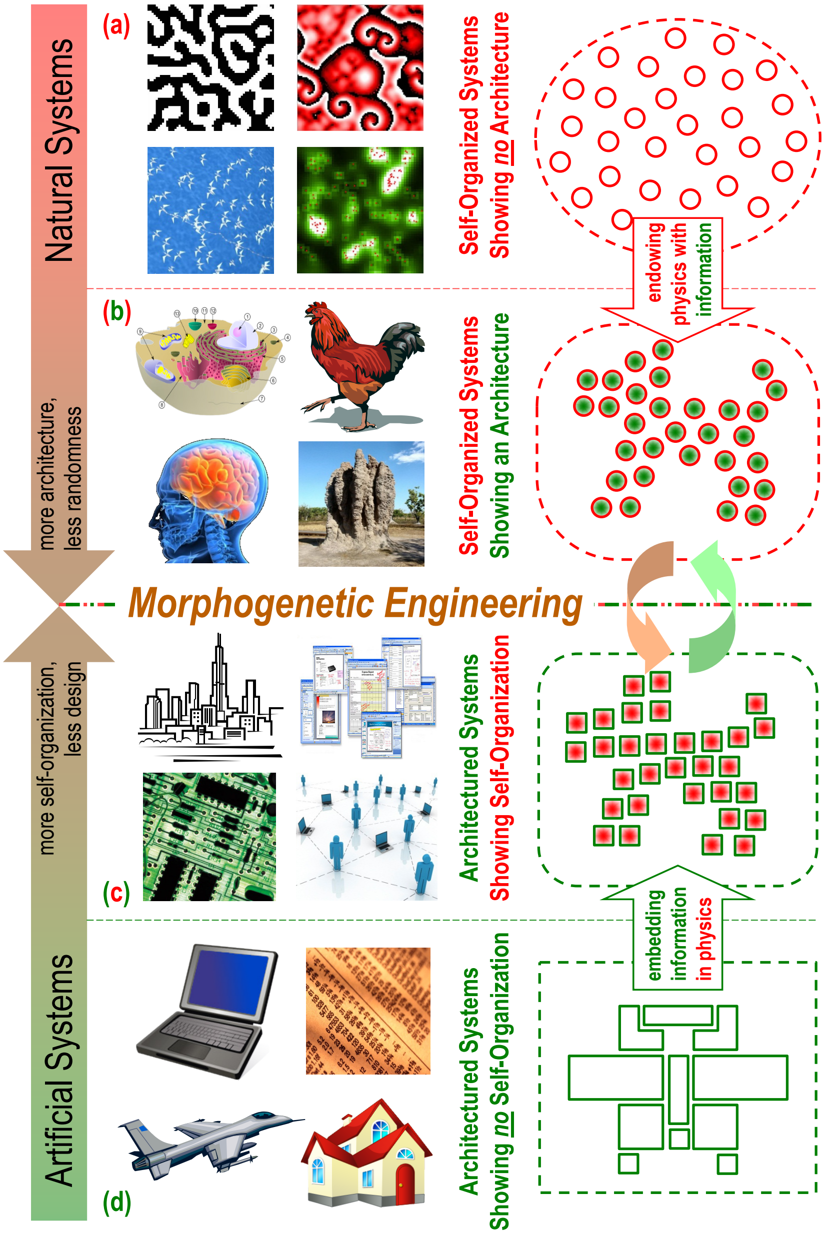


Fig. 1 Four families of systems representing various degrees of self-organization (vs. design), and architecture (vs. randomness). (a) Most natural complex systems are characterized by stochasticity, repetition and statistical uniformity: activator-inhibitor pattern formation (stripes and spots), traveling 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" [46] by inventing new systems that replace improvised with *programmed* complexification. See Fig. 2 for a zoom into the ME domain.

---

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). Its goal is to further explore the design and implementation of autonomous systems capable of developing complex, heterogeneous and *desired* functional architectures without relying on any central planning or external drive. In other words, while the above existing phenomena testify to the existence of *programmable self-organization*, the challenge of ME is to tap into this vast potential by proposing new "programmable" complex systems and "self-organized" engineered systems.

Keeping 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" [41]), can result in surprisingly complex and robust morphologies. Similarly, giving virtual wasps a pheromone that they can lay down and follow like termites ("waspmites" [10]) enhances their computation abilities, and transforms their usually repetitive nests into more elaborate constructions. In modern biotechnological endeavors such as synthetic biology [19, 28], 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" [44] (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 [1], spatial computing [20, 5, 6], organic computing [50], complex systems engineering [34] or emergent engineering [46]. 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 contributions to this book will show or hint at (see summary in 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 [34] 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 do it for you—and 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 [24] for that matter, in the sense that they are able to learn or evolve by themselves 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 through direct, internal learning or indirect, 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, and 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 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 Extended 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 [21, 38], uniform flocking [47], or undirected networking [3, 36, 4], but rather in virtual, extrapolated versions of these models, in which 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 of the elements and (ii) their ability to combine together in sufficiently various 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 can also be diversified, depending on the agent types and positions, introducing the required degree of heterogeneity for a system to exhibit a new type of behavior, more sophisticated than simple 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 [40], or embryonic development [18]. 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 nonsymbolic intelligence. This is the scope of reactive, behavior-based [9] or embodied robotics [39], exemplified by insect-like robots and evolving or reconfigurable mechanical morphologies [42, 30]. 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) [14, 8] or particle swarm optimization (PSO) [26] 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 non-developmental 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 [23] and Hopfield’s distributed attractor dynamics [25] 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 [24]. 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 and require great effort 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 revert to the latter type, to the detriment of the former, 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 states of order by evolving and recombining their local interactions. To imitate Ulam’s famous quip about nonlinear systems being the “non-elephant species” of physics, the pervasiveness of self-organized systems (vs. designed ones) make them, too, the non-elephant species of systems science—yet they remain the least familiar of them. Biological systems are not engineered and human-made systems could learn much more from them.

4 Contributions to this Book

The 18 contributed chapters of this book are all excellent illustrations of the above Morphogenetic Engineering manifesto. We chose to group them into four main parts according to the type of *dynamical process* characteristic of their morphogenetic models (Fig. 2). They can be summarized as follows:

- **Part I: Constructing** (Chapters 2-7): A relatively small number of mobile agents or components attach to each other or assemble blocks to build a precise, often "stick-figure" structure.
- **Part II: Coalescing** (Chapters 8-10): A great number of mobile agents flock and make together dense clusters, whose contours adopt certain shapes.
- **Part III: Developing** (Chapters 11-15): The system grows around a single initial agent or group by division or aggregation, forming biological-like patterns or organisms.
- **Part IV: Generating** (Chapters 16-19): The system grows by successive transformations of components in 3D space, based on a grammar of "rewrite" rules, creating various architectures.

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 chapters include several models belonging to more than one category, but chapters 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, and so on. Other terms such as "swarm" and "self-assembly" are used extensively in the majority of contributions, hence cannot constitute 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 Part I: Constructing

The first six chapters following this introduction describe ME systems in which relatively few robots or components (possibly originating from a larger, ambient pool) attach to each other or assemble blocks together, creating relatively precise formations. The built structures are "coarse-grained", and most of them stick figures, i.e., made of 1-unit wide segments. The space is the 2D plane, with occasional vertical elevation into 3D by stacking or folding. Chapters 2-6 feature physical and/or virtual morphogenetic robotic systems, while Chapter 7 distinguishes itself by a mechanical self-assembly system made of inert pieces that are shaken together. Chapter 3 also contains a developmental model typical of Part III contributions.

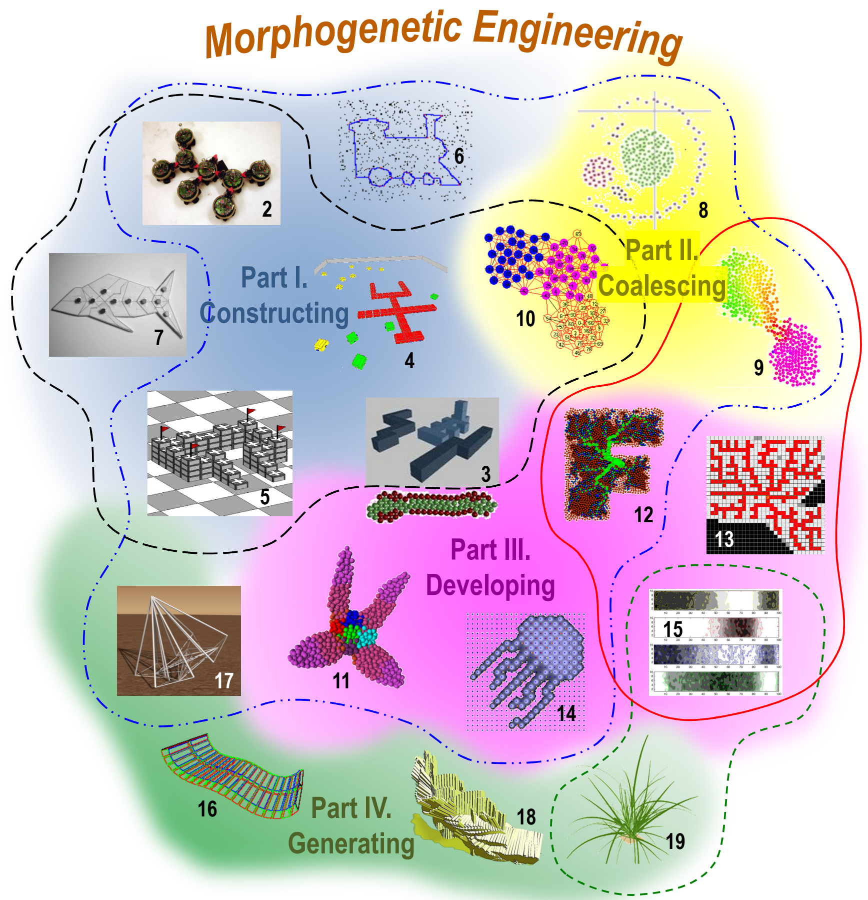
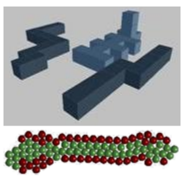
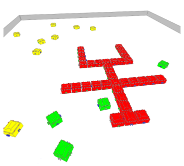


Fig. 2 The Morphogenetic Engineering continent represented in this book: 18 chapters were schematically grouped into four main parts according to their typical dynamics (see authors' names in Sections 4.1-4.4). **Part I: Constructing** (blue region, Chapters 2-7): A few agents attach to each other or 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.

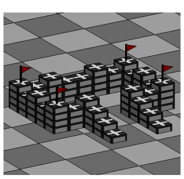
Chapter 2: SWARMORPH*: Morphogenesis with Self-Assembling Robots*, by O'Grady, Christensen & Dorigo: In the Swarm-bot platform, robots are given the capacity to assemble into appropriate morphologies and operate as a single entity when physically connected to one another. This chapter presents a low-level control logic to allow inter-robot connections to be formed at particular angles, and a higher-level control logic to dictate the sequence of these connections so as to form desired morphologies. The latter also allows robots to make appropriate collective responses to different tasks. The morphology generation capabilities of this framework are tested with real-world experiments (up to nine robots) and physics-based simulations to verify its scalability.



Chapter 3: *Morphogenetic Robotics: A New Paradigm for Designing Self-Organizing, Self-Reconfigurable and Self-Adaptive Robots*, by Jin & Meng: This chapter reviews under the term *morphogenetic robotics* the class of methodologies for designing self-organizing, self-reconfigurable and self-adaptive robots inspired by biological morphogenesis. It comprises three main areas: swarm robotic systems, modular robots and body-brain co-design. The relationships between morphogenetic robotics and *epigenetic robotics*, which focuses on cognitive development, and also *evolutionary robotics*, which is concerned with evolutionary design of controllers, are discussed. A few examples illustrate the main ideas underlying morphogenetic approaches to robotics.

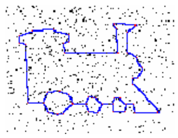


Chapter 4: *Distributed Autonomous Morphogenesis in a Self-Assembling Robotic System*, by Liu & Winfield: This chapter presents distributed morphogenesis control strategies in a swarm of robots able to autonomously disassemble and reassemble into different 3D symbiotic organisms. The idea is to combine the advantages of swarm and self-reconfigurable robotic systems in order to investigate and develop novel principles of evolution and adaptation for "robotic organisms" from bio-inspired and evolutionary perspectives. Robots here are independently mobile and can autonomously dock to each other. They initially form a 2D planar structure, then the aggregated organism must lift itself to a 3D morphology, move and function as a macroscopic whole.

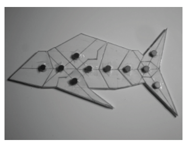


Chapter 5: *Collective Construction with Robot Swarms*, by Werfel: Social insects build large, complex structures, which emerge through the collective actions of many simple agents acting with no centralized control or preplanning. These natural systems inspire the research topic of collective construction, in which the goal is to engineer artificial systems that build in a similar way, with swarms of simple robots producing desired structures. This chapter reviews work on the design and implementation of such systems. Robots act independently, using only local information and no explicit communication; the sys-

tem takes only a high-level design as input, and is guaranteed to produce a structure matching that design.



Chapter 6: *Issues in Self-Repairing Robotic Self-Assembly*, by Arbuckle & Requicha: Robot swarms provide interesting and potentially very useful examples of self-organizing systems. This chapter focuses on a specific approach, dubbed "active self-assembly", for constructing arbitrary shapes with swarms of identical and identically programmed robots. Important, open issues are identified in the specific context of active self-assembly, but they are of more general applicability. Much of the 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.



Chapter 7: *Programming Self-Assembling Systems via Physically Encoded Information*, by Bhalla & Bentley: Natural self-assembly is dictated by the morphology and properties of the components and environmental conditions. The process of self-assembly is equivalent to a physical computation, through the interaction and transformation of physically and chemically encoded information. How does one program physical self-assembling systems? This chapter proposes a three-level design approach, which specifies a set of simple self-assembly rules, models and simulates these rules in software, then translates them to a physical system. The objective is to provide a bottom-up design methodology to create scalable self-assembling systems.

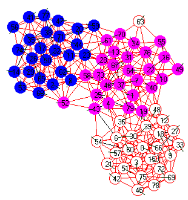
4.2 Part II: Coalescing

The next three chapters deal with a great number of mobile agents that exhibit *heterogeneous* flocking behavior. Without literally attaching, they aggregate and stay near each other while moving to maintain neighbor-to-neighbor communication (e.g., of a visual, chemical, or wireless type). Together, they tend to form dense, fine-grained clusters that assume certain "fluid" yet stable shapes. For understandable reasons, all contributions 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. Chapter 10, however, is directly motivated by a practical robotic application, and remains close to its source.

Chapter 8: *Swarm-Based Morphogenetic Artificial Life*, by Sayama: This chapter presents a "swarm chemistry" framework to design and implement morphogenetic artifacts that grow and self-organize in a fully decentralized manner. Swarms comprise multiple types of simple, interacting mobile particles with no elaborate connection or computation capabilities, which can still

produce 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 redifferentiation.

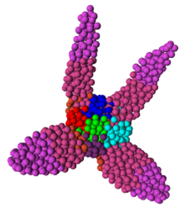
Chapter 9: *Chemotaxis-Inspired Cellular Primitives for Self-Organizing Shape Formation*, by Bai & Breen: Motivated by the ability of living cells to form specific structures, this chapter investigates chemotaxis-inspired cellular primitives for self-organizing shape formation. Cells emit a chemical into their environment, and move in the direction of the gradient of the cumulative chemical field from other cells. This behavior is modeled by 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.



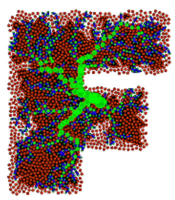
Chapter 10: *Emergent Swarm Morphology Control of Wireless Networked Mobile Robots*, by Nembrini & Winfield: This chapter describes decentralized control algorithms that link local wireless connectivity to low-level robot motion control for maintaining both swarm aggregation and connectivity, or "coherence". It investigates the potential of both first and second order connectivity information (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.

4.3 Part III: Developing

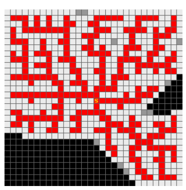
The inspiration for the five chapters of this part is situated closer to cell-based models of biological morphogenesis. Here, systems start from a single agent or small group and grow to a relatively large size by repeated, yet differential, division or aggregation. Mechanisms underlying this development 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.



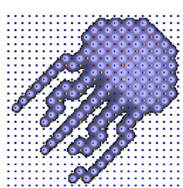
Chapter 11: *Embryomorphic Engineering: Emergent Innovation Through Evolutionary Development*, by Doursat, Sánchez, Dordea, Fourquet & Kowaliw: Embryomorphic Engineering combines three key principles of multicellular biological development: chemical gradient diffusion (providing positional information to the agents), gene regulatory networks (triggering their differentiation into types, thus patterning), and cell division (creating structural constraints, hence reshaping). It is illustrated in two different spaces: 2D/3D swarms with potential applications in collective robotics, and graph topologies with potential applications in peer-to-peer device/user networks. In all cases, the phenotype's architecture is programmable in the genotype shared by the agents.



Chapter 12: *Functional Blueprints: An Approach to Modularity in Grown Systems*, by Beal: The engineering of grown systems poses fundamentally different challenges than static designs. Growth offers much greater potential for adaptation to changes in the environment, but a grown system must also be capable of surviving every intermediate stage, despite internal stresses due to the uneven scaling of its subsystems. This chapter 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.

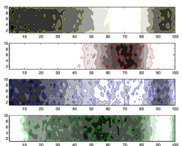


Chapter 13: *Mechanisms for Complex Systems Engineering Through Artificial Development*, by Kowaliw & Banzhaf: This chapter considers the means by which morphogenetic growth can lead to complex systems design. The evolvability of a difficult design space can be enhanced through mechanisms such as *regularities* and *adaptive feedback*. A concrete example is presented and applied 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. This "local fitness" is used as a drive during the development of the final design. This approach is shown to improve the efficacy of the learner, and to eliminate the problem's hardness associated with the complexity of the environment.



Chapter 14: *A Synthesis of the* Cell2Organ *Developmental Model*, by Cussat-Blanc, Pascalie, Mazac, Luga & Duthen: Developmental mechanisms of living beings have inspired *artificial embryogeny*, i.e., the generation of small creatures composed of a few hundred cells starting from a single cell. To create complete organisms containing different organs and high-level functionalities, this chapter proposes a three-layer developmental framework: a *chemical* layer, where cells can divide and metabolize substrates, a *hydrodynamic* layer sim-

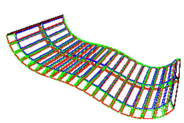
ulating substrate flows, and a *physics* layer that allows cells to change shape and organisms to move. Additionally, a new method based on L-systems without molecular morphogens is also introduced.



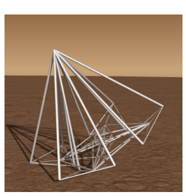
Chapter 15: *A Computational Framework for Multilevel Morphologies*, by Montagna & Viroli: The hierarchical organization of biological systems plays a crucial role in pattern formation regulated by gene expression, and morphogenesis in general. Modeling and simulating the developmental dynamics of living organisms at multiple scales might prove useful in the design of engineered products that manifest spatial self-organizing properties. This chapter describes 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 scenarios.

4.4 Part IV: Generating

In the last four chapters, morphogenetic systems are generated by successive transformations of components in 3D space, based on "rewrite" rules. These rules are formally expressed as "grammars", which can be designed by hand or evolved. The resulting architectures have potential applications as diverse as natural computing, robotics, computer graphics or plant biology.



Chapter 16: *Interaction-Based Modeling of Morphogenesis in MGS*, by Spicher, Michel & Giavitto: This chapter advocates a domain specific language (DSL) approach to overcome the difficulties of modeling and simulating morphogenetic processes. To this aim, it presents an experimental programming language called MGS. The declarative approach of MGS is based on the notion of *topological collection*, which arises naturally when trying to model "dynamical systems with a dynamic structure". The evolution function of such systems is specified by *transformations* made of sets of rewriting rules, where each rule defines a local interaction. The MGS framework is illustrated here by various models of the same T-shape growth.



Chapter 17: *Behavior-Finding: Morphogenetic Designs Shaped by Function*, by Lobo, Fernández & Vico: Evolution has shaped an incredible diversity of multicellular organisms, whose complex forms are self-made by robust developmental processes. This fundamental "evo-devo" combination can inspire novel computational methodologies to overcome the scalability problems of classical top-down design. This chapter describes evolutionary morphogenetic algorithms automating the design of "organic" morphologies and controllers

intended to solve certain functional problems. Performance is tested on "behavior-finding" optimization challenges not based explicitly on structural constraints, but only on solving abilities and functional efficacy.

Chapter 18: *Swarm-Based Computational Development*, by von Mammen, Phillips, Davison, Jamniczky, Hallgrímsson & Jacob: In swarms, as in complex systems, large numbers of individuals locally interact and form non-linear, dynamic interaction networks. Ants, wasps and termites, for instance, are natural swarms whose individual and group behaviors have been evolving over millions of years. In their intricate nest constructions, the emergent effectiveness of their behaviors becomes apparent. Swarm-based computational simulations capture the corresponding principles of agent-based, decentralized, self-organizing models. This chapter presents ideas around swarm-based generative and developmental systems, in particular *swarm grammars*.

Chapter 19: *Programmable and Self-Organized Processes in Plant Morphogenesis: The Architectural Development of Ryegrass*, by Verdenal, Combes & Escobar-Gutiérrez: Forage grass morphology 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. This chapter presents 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.

5 Perspectives

Biological organisms are the pure products of undesigned evolution (UDE) by random variations (expressed through 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 this link should become less and less clear or direct. Traditional engineering has always followed a "directly designed construction" (DDC) paradigm, in which architects plan and build entire systems top-down. Morphogenetic Engineering proposes a gradual shift toward biology, 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 these local mechanisms, prepare a few primitive ancestor systems, then step back to let evolution and development invent everything else.

Morphogenetic Engineering endeavors are related to those of other innovative fields that have emerged during the last decade, mostly during the 2000's: artificial embryogeny (AE) [7, 33, 43, 29, 15], amorphous computing [1, 13, 35, 49], spatial computing [20, 5, 6], programmable matter [22], autonomic computing [27], organic computing [31, 50], natural/unconventional computing [44, 37], complex systems engineering [34], ambient intelligence [32], and pervasive/ubiquitous computing [48]. ME, 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.

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. They need to design for *emergence*, i.e., for systems that fundamentally and continually adapt and evolve. The appeal of Morphogenetic Engineering 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.

References

1. Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight Jr, T.F., Nagpal, R., Rauch, E., Sussman, G.J., Weiss, R.: Amorphous computing. Communications of the ACM **43**(5), 74–82 (2000)
2. Ball, P.: The Self-Made Tapestry. Oxford University Press (1999)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
4. Barrat, A., Barthélemy, M., Vespignani, A.: Dynamical processes on complex networks. Cambridge University Press (2008)
5. Beal, J., Bachrach, J.: Infrastructure for engineered emergence on sensor/actuator networks. Intelligent Systems, IEEE **21**(2), 10–19 (2006)
6. Beal, J., Dulman, S., Usbeck, K., Viroli, M., Correll, N.: Organizing the aggregate: Languages for spatial computing. Computing Research Repository **abs/1202.5509** (2012)
7. Bentley, P., Kumar, S.: 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 (1999)
8. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press (1999)

9. Brooks, R.: A robust layered control system for a mobile robot. Robotics and Automation, IEEE Journal of **2**(1), 14–23 (1986)
10. Bullock, S., Ladley, D., Kerby, M.: Wasps, termites and waspmites: Distinguishing competence from performance in collective construction. Artificial Life **18**(3), [forthcoming] (2012)
11. Carroll, S.B.: Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom. W. W. Norton (2005)
12. Coen, E.: The Art of Genes. Oxford University Press (2000)
13. Coore, D.N.: Botanical computing: a developmental approach to generating interconnect topologies on an amorphous computer. Ph.D. thesis, MIT (1999)
14. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press (2004)
15. Doursat, R.: Organically grown architectures: creating decentralized, autonomous systems by embryomorphic engineering. In: R.P. Würtz (ed.) Organic computing, Understanding complex systems, pp. 167–199. Springer (2008)
16. Doursat, R.: Morphogenetic engineering weds bio selforganization to human-designed systems. PerAda Magazine (2011)
17. Doursat, R.: The myriads of alife: Importing complex systems and self-organization into engineering. In: Proceedings of the 3rd IEEE Symposium on Artificial Life (IEEE-ALIFE 2011), pp. xii–xix. IEEE (2011)
18. Eggenberger, P.: Evolving morphologies of simulated 3d organisms based on differential gene expression. In: Proceedings of the Fourth European Conference on Artificial Life, pp. 205–213 (1997)
19. Endy, D.: Foundations for engineering biology. Nature **438**(7067), 449–453 (2005)
20. Giavitto, J.L., Michel, O.: The topological structures of membrane computing. Fundamenta Informaticae **49**(1-3), 123–145 (2002)
21. Gierer, A., Meinhardt, H.: A theory of biological pattern formation. Biological Cybernetics **12**(1), 30–39 (1972)
22. Goldstein, S., Campbell, J., Mowry, T.: Programmable matter. Computer **38**(6), 99–101 (2005)
23. Hebb, D.O.: The organization of behavior. John Wiley and Sons (1949)
24. Holland, J.H.: Adaptation in natural and artificial systems. 53. University of Michigan Press (1975)
25. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences **79**(8), 2554 (1982)
26. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings, IEEE International Conference on, vol. 4, pp. 1942–1948. IEEE (1995)
27. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
28. Knight, T.: Idempotent vector design for standard assembly of biobricks. Tech. rep., DTIC Document (2003)
29. Kowaliw, T., Grogono, P., Kharma, N.: Bluenome: A novel developmental model of artificial morphogenesis. In: Genetic and Evolutionary Computation GECCO '04, pp. 93–104. Springer (2004)
30. Lipson, H., Pollack, J.B.: Automatic design and manufacture of robotic lifeforms. Nature **406**(6799), 974–978 (2000)
31. Malsburg, C.: Organic computing. In: R.P. Würtz (ed.) Organic computing, Understanding complex systems, chap. The organic future of information technology, pp. 7–24. Springer (2008)
32. Marzano, S., Aarts, E.: The New Everyday View on Ambient Intelligence. Uitgeverij 010 Publishers (2003)
33. Miller, J.F., Banzhaf, W.: Evolving the program for a cell: from french flags to boolean circuits. On Growth, Form and Computers pp. 278–301 (2003)
34. Minai, A.A., Braha, D., Bar-Yam, Y.: Complex engineered systems: Science meets technology. In: D. Braha, Y. Bar-Yam, A.A. Minai (eds.) Complex Engineered Systems: Science Meets Technology, chap. Complex engineered systems: A new paradigm, pp. 1–21. Springer (2006)

35. Nagpal, R.: 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 (2002)
36. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006)
37. Nunes de Castro, L.N.: Fundamentals of natural computing: basic concepts, algorithms, and applications. CRC Press (2006)
38. Pearson, J.E.: Complex patterns in a simple system. Science **261**(5118), 189–192 (1993)
39. Pfeifer, R., Bongard, J., Grand, S.: How the body shapes the way we think: a new view of intelligence. The MIT Press (2006)
40. Rothemund, P.W.K.: Folding dna to create nanoscale shapes and patterns. Nature **440**(7082), 297–302 (2006)
41. Sayama, H.: Swarm chemistry. Artificial Life **15**(1), 105–114 (2009)
42. Sims, K.: Evolving 3d morphology and behavior by competition. Artificial life **1**(4), 353–372 (1994)
43. Stanley, K.O., Miikkulainen, R.: A taxonomy for artificial embryogeny. Artificial Life **9**(2), 93–130 (2003)
44. 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., Partridge, D.: Journeys in non-classical computation i: A grand challenge for computing research. International Journal of Parallel, Emergent and Distributed Systems **20**(1), 5–19 (2005)
45. Tanenbaum, A.S., van Steen, M.: Distributed Systems: Principles and Paradigms. Prentice Hall (2002)
46. Ulieru, M., Doursat, R.: Emergent engineering: a radical paradigm shift. International Journal of Autonomous and Adaptive Communications Systems **4**(1), 39–60 (2011)
47. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Physical Review Letters **75**(6), 1226–1229 (1995)
48. Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM **36**(7), 75–84 (1993)
49. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. Intelligent Systems, IEEE **21**(2), 20–28 (2006)
50. Würtz, R.P.: Organic computing. Springer (2008)