Morphogenetic engineering weds bio self-organization to human-designed systems

René Doursat

Deeper understanding of complex biological and engineered systems should lead to the emergence of new hybrid designs that provide the best of both worlds.

Information and communication technology (ICT) systems are fundamentally different from natural complex systems (CS). ICT products—computer, control, and communication software and hardware—are assembled from unique, heterogeneous components in precise, often complicated ways. By contrast, self-organization in a physical, biological, ecological or social CS relies on myriad identical elementary agents behaving under relatively simple rules and stochastic (random) or indeterminant dynamics: see Figure 1(a). In CS, nontrivial collective behaviour can emerge and evolve from purely local interactions, properties often touted as the hallmark of complex systems. There are, however, major exceptions that come into the picture and blur the differences between ICT and CS, revealing a possible path toward what could become tomorrow’s ICT-controlled CS and CS-inspired ICT.

We propose to call this new field ‘morphogenetic engineering’. Just as other disciplines have worked toward new computing principles using neurons, ants and genes as models, morphogenetic engineering’s purpose is to abstract the behaviour of cells, termites and other natural agents into new principles of heterogeneous, controllable self-assembly.

Hybrid systems

ICT systems must function deterministically according to their designers’ specifications: see Figure 1(d). By contrast, most of the spontaneous order of CS falls into the pattern-formation category (spots, stripes, waves, flocks, trails, branches, clusters, hubs and so forth), that is, random, repetitive and statistically uniform phenomena. These CS motifs can be moulded into different shapes and variants through externally imposed boundary conditions. But they do not possess a genuine intrinsic architecture such as that exhibited by ICT systems on the hardware and software levels. From this viewpoint, the distance between a beehive comb and a processor’s architecture can seem vast indeed.

But consider ICT-controlled CS or the steering of CS that already resemble ICT systems. Certain biological CS display strong morphogenetic abilities, ones more sophisticated than texture-like pattern formation: see Figure 1(b). These systems are, literally, living proof that pure self-organization and elaborate architectures are mutually compatible. This is especially the case with embryogenesis, the self-assembly of cellular masses into detailed body plans. It is also seen in some insect colonies: the undirected coordination or stigmergic collaboration of swarms of insects building giant constructions such as termite mounds.

Multicellular organisms are composed of organs and appendages arranged in highly specific ways. Yet they entirely self-assemble through a decentralized, but finely synchronized, choreography of cell proliferation, migration and differentiation, under the guidance of genetic and epigenetic information spontaneously evolved over millions of years and stored in every cell. Similarly, at a higher scale, social insects such as termites, ants and wasps are also capable of collectively constructing complicated but extremely well-organized nests without any global plan or grand architect directing them from the outside.

In modern biotechnological endeavours such as synthetic biology, the instructions inside every agent or cell of such natural CS could then be modified by ICT techniques in targeted ways to steer the emergent collective behaviour of cellular populations towards technologically desirable outcomes, whether for biomedical or natural-computing applications.

Next, consider the opposite side: CS-inspired ICT, which can be viewed as emancipating ICT systems that already de facto resemble CS. Large-scale artificial ICT systems already exhibit complex-systems effects, albeit still mostly uncontrolled and undesired at this point. Segmentation and distribution of large computing systems over a multitude of smaller and relatively

Continued on next page
simpler components has become both a growing need and an inevitable reality in many domains of computer science and engineering, artificial intelligence and robotics.

Faced with an explosion in system size at all scales, whether hardware (integrated parts), software (program modules) or networks (applications and users), engineers will have to give up the rigid design of systems in every detail and rethink them in terms of CS. Instead of direct design, they should focus on ‘meta-design’: the generic conditions allowing the endogenous self-assembly, self-regulation and evolution of these systems. Several bio-inspired ICT methodologies already embody the idea of indirect design based on CS by deriving principles from the observation of neurons (abstracted into artificial neural networks), animal societies (abstracted into ant colony optimization and particle swarm optimization) or evolution (abstracted into genetic algorithms and genetic programming). Engineers then apply these principles to problem-solving tasks such as machine learning, classification, stochastic optimization, parameter search or automated design. Yet, these established bio-inspired ICT disciplines emphasize the preliminary, offline improvement of some computational ability or global blueprint before actual functioning takes place, rather than the emergence and re-emergence of various architectures, on the same fast timescale as the system’s dynamics. This is
where morphogenetic engineering, a new line of bio-
inspiration mostly originating from *biological development* comes on stage.

In sum, while complex systems already include natural sys-
tems that seemingly exhibit all the attributes of ICT systems, the latter are already becoming natural objects of study for CS researchers. Both of these cross-boundary examples and trends point to a new field that should explore the design and im-
plementation of autonomous systems capable of developing complex and *desired* functional architectures with little or no central planning. In other words, they are captivating examples of *programmable self-organization*: a hybrid concept insufficiently explored so far, whether in natural CS for the programmable part, or in traditional ICT engineering for the self-organization component.

Research in morphogenetic engineering is positioned at the in-
terface between the science and engineering of complex systems. It covers the computational modelling of self-organization phe-
nomena and the design of decentralized and adaptive artificial systems inspired by these phenomena. It aims to open new per-
spectives in software, robotic, electrical, mechanical or even civil engineering. It asks whether a sophisticated device or network architecture can construct itself from a reservoir of small compo-
ents: Can a robot rearrange its parts and evolve towards better performance without explicit instructions? Can a swarm of soft-
ware agents self-organize and collectively innovate in problem-
solving tasks?

**Artificial development**

Towards answering these kinds of questions, I initiated vari-
ous studies based on abstract, multi-agent models of biological development,\(^9\)\textsuperscript{−}\textsuperscript{11} with possible application to the field of collective robotics. Later, I extended these models to the self-organization of precise network topologies\(^12\),\textsuperscript{,}\textsuperscript{13} by ‘pro-
grammed attachment’ of nodes (in contrast to random ‘prefer-
tential attachment’\(^14\)). This work could contribute to the field of autonomic networks.

The spontaneous making of an entire organism from a sin-
gle cell is the epitome of a programmed, self-organizing CS. Through a precise interplay of genetic switches and chemical gradients in time and space, an elaborate form is created with-
out an explicit architectural plan or design intervention. On an abstract level, this phenomenon can best be understood as a combination of *self-assembly* and *pattern formation* under the control of non-random genetic regulation stored inside each agent. To take an artistic metaphor, this would be similar to the simultaneous self-sculpting and self-painting of a canvas.\(^\textsuperscript{15}\) The differential properties of cells, such as adhesion and division, are determined by the regions of gene expression to which they be-
long, while at the same time, these regions further expand and segment into subregions due to the self-assembly of differentiating cells.

To reflect this, the model I propose is computational, spatial and agent-based (see Figure 2). It can be equivalently construed from two different vantage points: either as pattern formation on *moving* cellular automata, in which the cells spatially rearrange under the influence of their activity pat-
tern, or as collective motion in a *heterogeneous* swarm, in which the agents gradually differentiate and modify their interactions according to their positions and the regions they form.

Each cell contains a genetic regulatory network (GRN), mod-
elled as multilayered switches that can settle in various on/off expression states. Local morphogen gradients \((X, Y, \text{‘grad’})\) in Figure 2), generated by a fully decentralized, peer-to-peer signalling system (a form of ‘gossiping’), provide positional information in input. This information is then integrated by each GRN to produce differential expression of identity genes \((I, J, \ldots)\) in output, which represent different-coloured cell types.

Similar to striping in the fruit-fly embryo, the lattice becomes segmented into spatial regions of homogeneous gene expression states (one for each identity gene\(^\textsuperscript{16}\)) that resemble stained glass motifs (‘patt’ in Figure 2). Meanwhile, it also expands by cell prolif-
eration (‘div’ in Figure 2), itself triggering new local gradients of positional information within former homogeneous regions, and so forth.

**Robotic evo-devo and network embryogeny**

This alternation of growth and patterning ultimately results in the creation of a reproducible multicellular anatomy.\(^9\)\textsuperscript{–}\textsuperscript{11} If the same genome is run again, the same macro and meso structures reappear, showing invariance by random fluctuations in the ex-
act number and location of cells at the micro level. If, however, mutated genomes are used, then new shapes emerge. This opens the way to original exploration methods by artificial evolution-
ary development or ‘evo-devo engineering’ (see Figure 2, right).

A direct application of this model can be found in collect-
ive robotics. New technological advances in hardware design have made smaller, lighter and cheaper robots possible, while retaining sufficient memory and computing capacity for a great diversity of behavioural programs. This opened the way to mass replication and the creation of robotic swarms able to display
various kinds of group behavior. Still, it remains difficult to program these units in a sufficiently generic way, based purely on local interactions with each other and external events, so as to make them self-organize and self-adapt adequately.

The goal of evo-devo is to find good solutions at the swarm level to different challenges posed by new and unknown environments. These are typically physical tasks, such as overcoming an obstacle, bump or gap, or moving an object, for which robots need to come together and form specific superstructures.

Another closely related project I have launched addresses the self-assembly of complex, precise network topologies by programmed attachment. It is an extension and generalization of the artificial development model described above, from 2D and 3D multicellular organisms to autonomous n-D network construction and dynamics. Here, nodes execute the same program in parallel and communicate and differentiate, while links are dynamically created and removed based on ‘ports’ \((X, X')\) and ‘gradient values’ \((x, x')\) that guide nodes to specific attachment locations. As the network expands, nodes switch different rules on and off, creating chains, lattices and other composite topologies.

More precisely, nodes initially have both ports open and gradients set to 0: see Figure 3(a). A node \(i\) can then create a link with another node \(j\) only through a pair of complementary open ports, \(X\) and \(X'\). As soon as a new link is established, gradients are updated by propagation of discrete increments \(x\) and \(x'\) in opposite directions. The purpose of the gradient counters is to keep track of the nodes’ positions in the chain. This allows, for example, the building of chains of a fixed length \(n\) (by closing ports as soon as \(x + x' = n - 1\)) and the creation of more complicated structures by switching on or off certain attachment rules when certain gradient values have been reached.

While an agent can connect to any available port at random, the final superstructure will always be the same, which is the whole purpose of programmable self-organization. All nodes carry the same program—their genotype or ‘DNA’, see Figure 3(g)—which consists of three main routines: gradient update \((G)\), port management \((P)\) and link creation \((L)\). \(G\) is the generic code that provides nodes with the positional information they need to make further decisions. \(P\) contains the heart of the logic specific to a target structure. \(G\) and \(P\) are executed by the nodes already involved in the network and prepare the way for newly arrived nodes to execute \(L\). Routine \(L\) provides the generic logic that prompts new nodes to pick one of the network’s open ports at random and make a new connection.

Finally, in addition to their endogenous morphogenetic abilities, self-organized network morphologies can also exhibit a high degree of adaptation to exogenous constraints imposed by the environment, such as spatial boundary conditions. Each network is based on the same node program (genotype), yet may grow differently (polymorphism of the phenotype) as it senses...
its environment, for example, via an anticollision rule between the nodes and certain obstacles (see Figure 3, right).

The study on 2D/3D swarm embryogenesis imagined self-assembling circuits, computers, vehicles and so forth, composed of a swarm of small components, parts and modules, either self-propelled or carried by mini robots, that would aggregate in an orderly fashion without following a central direction or global blueprint. In this section, the model of n-D network development envisions at a higher scale a self-reconfiguring manufacturing plant, a self-stabilizing energy grid, a self-deploying emergency team of first responders or a self-regulating social organization, all relying on myriad mobile devices, software agents and human users. Each would build their own dynamic and non-random adaptive network on the sole basis of local rules and peer-to-peer communication.

The road ahead
Whether in 2D/3D devices or n-D techno-social webs, decentralized automation giving rise to emergent architectures promises to be a fruitful new design paradigm. Traditionally, the role of an engineer is that of an active planner, who enforces hierarchical, top-down and linear thinking on passive materials, often in complicated (but not ‘complex’) ways. By contrast, to obtain innovative, unplanned emergent behaviour, engineers should only guide existing bottom-up interactions among a multitude of active components by endowing these components with certain parameters and rules or, better still, with some metaheuristic computational methods that give them the generic capability to acquire these parameters and rules by themselves.

Engineers need to design for emergence or ‘meta-design’ for systems that fundamentally and continually adapt and evolve. The hope is that it would bring many beneficial self-x’ properties to improve, complement or even replace current human-led efforts. For example, it could allow remote operations in hostile places, drive faster organization without the usual delays tied to a central command node, deliver greater robustness and reactivity to new events or environments, achieve better scalability if the system needs to grow and, possibly the greatest achievement of all, give the system ability to learn.

It is difficult to overstress the importance of fundamental laws of developmental variation on the dynamic timescale of artificial systems as a prerequisite to selection on the evolutionary timescale. Beside morphogenetic engineering, other innovative research avenues in complex design, such as artificial embryogeny, 17, 18 amorphous computing, 19, 20 spatial computing, 21 autonomic computing, 22 organic computing 23 and complex systems engineering, 6 recognize this as well. Note that this thesis mirrors the ‘structuralist’ paradigm in biology, 24, 25 which insists on the key role played by self-organized order in the spontaneous creation and proposal of innovative structures before any selection of these structures can take place. In evolutionary computational parlance, this means an indirect or implicit mapping (as opposed to direct or explicit) from genotype to phenotype.

In sum, all these new bio-inspired ICT trends bring to the traditional fields of control theory and systems engineering a fresh

Continued on next page

Figure 3. Left: Illustration of the self-assembly mechanisms of complex but precise network topologies by ‘programmed attachment’. Right: Example of simulation in an environment with obstacles, which alter an intrinsically programmed spider-shaped template. P: Port management. Xa−c, X′a−c: Different types of attachment ports. x, x′: Gradient values. (After Ulieru and Doursat.13 Copyright © 2004–2011 Inderscience Enterprises Limited.)
perspective that can be globally called ‘emergent engineering’, in which architectures are partially or entirely made without architects. Emergent engineering relies on the specification of basic agents and the mechanisms by which these agents together will be able to create reliable components and modules. In this sense, it radically departs from classical service-oriented architectures, in which modules are predefined in a top-down fashion. On the contrary, the objective here and in the future is to let architectures grow and evolve bottom-up from the many interactions among autonomous and social agents.

Author Information

René Doursat
Complex Systems Institute, Paris Ile-de-France
CREA, Ecole Polytechnique and CNRS
Paris, France

René Doursat is a researcher and lecturer in computer science, complex systems and neural networks. After a detour of several years through the software industry following his PhD and post-doctoral appointments, he returned to academia in 2004. He has held a position at the Complex Systems Institute in Paris since 2007 and was also its director for two years. His research focuses on the multi-agent modelling of complex systems, in particular biological systems, which can inspire novel principles in intelligent systems design.

References