The Self-Construction and -Repair of a Foraging Organism by Explicitly Specified Development from a Single Cell

Fabian Roth*
Institute of Neuroinformatics
University/ETH Zürich
Winterthurerstrasse 190
8057 Zürich, Switzerland
fin@ini.phys.ethz.ch
and
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
43 Vassar Street
Cambridge, MA 02139
faroth@mit.edu

Hava Siegelmann
Department of Computer Science
University of Massachusetts
140 Governors Drive
Amherst, MA 01003
hava@cs.umass.edu

Rodney J. Douglas
Institute of Neuroinformatics
University/ETH Zürich
Winterthurerstrasse 190
8057 Zürich, Switzerland
rjd@ini.phys.ethz.ch

Abstract As man-made systems become more complex and autonomous, there is a growing need for novel engineering methods that offer self-construction, adaptation to the environment, and self-repair. In a step towards developing such methods, we demonstrate how a simple model multicellular organism can assemble itself by replication from a single cell and finally express a fundamental behavior: foraging. Previous studies have employed evolutionary approaches to this problem. Instead, we aim at explicit design of self-constructing and -repairing systems by hierarchical specification of elementary intracellular mechanisms via a kind of genetic code. The interplay between individual cells and the gradually increasing self-created complexity of the local structure that surrounds them causes the serial unfolding of the final functional organism. The developed structure continuously feeds back to the development process, and so the system is also capable of self-repair.

Keywords
Self-construction, self-repair, behaving organism, stigmergy, cellular development

1 Introduction

Machines and engineered systems are usually designed and constructed by a process that transforms an explicit, abstract description of the target into a physical instance. Both the designer and the constructor are external to the target instance, which is passively assembled to conform to its blueprint. When the target is fully assembled, it becomes functional, and only then can its performance be verified exhaustively. As engineered systems become more complex, this conventional feedforward

* Corresponding author.

method of construction is becoming inadequate, because the cost and complexity of precise design, assembly, and testing are becoming prohibitive. Also, it is difficult to specify a priori the full range of configurations that the target system may experience during operation, and some of these unknown states may lead to critical failures.

By contrast, nature constructs extremely complex organisms by autonomous construction from a single precursor cell. Both the developing and the final organism are very robust against perturbation. There are astonishingly few defects in construction, or failures of performance, despite large variations in operating conditions. Moreover, the organism is able to compensate for injury and infection by self-repair. An understanding of these natural principles of self-construction would permit the development of scalable self-construction and self-repair in artificial systems—properties that become ever more relevant as we confront the challenges of manufacturing and foraging in hostile [15] or nanoscale [13, 11, 28, 40] environments.

In his pioneering work, von Neumann [38] proposed minimal requirements for self-replication: A universal *constructor*, and an explicit *description* of the target system. The constructor reads the description, and translates that information into the construction of the target, which in turn is also capable of self-replication. von Neumann used an explicit one-to-one mapping between the description (a blueprint) and the target system. While his work is a remarkable theoretical achievement, systems of his kind remain subject to the same problems that traditionally engineered systems have: They are not robust against structural defects. Their construction process is purely feedforward and therefore insensitive to their own structure during assembly. Hence, they are unable to compensate for deficiencies and perturbations. A further difference from the von Neumann concept is that nature does not encode the entire target system explicitly in the description. Instead, nature encodes modular rules that provide an indirect mapping between genotype and phenotype. One obvious advantage of this indirect rule-based mapping is that it naturally compresses information about the explicit structure, as in algorithmic information theory, where the specification of an (algorithmically) nonrandom structure can be compressed into an algorithm [8]. Modularity also promotes the automatic emergence of complexity through iterative application of simple rules [39], and the ability to adapt to environmental signals [23].

There has been considerable previous research on self-construction and repair (see Section 5). In general, however, that work has relied heavily on genetic algorithms to discover suitable assembly instructions. Genetic algorithms are useful for finding solutions in a large search space, but we consider that these algorithms are an inappropriate foundation for an effective technology of self-construction, for the following reasons.

Firstly, the use of genetic algorithms is only feasible when fitness can be evaluated quickly. They are less suitable for configuring self-constructing systems where a large fraction of the time is required for the development of the system before it can be evaluated. In particular, a trial and selection scheme with real hardware in a real environment would also be economically infeasible. The search process could be sped up by simulation [24], but that approach raises the additional problem of the *reality gap*, the imprecisions in the mapping between the simulated universe and reality.

Secondly, a technology is usually task orientated. That is, we require that the self-organized organism possess some specific physical characteristics and functional competences, which will enable the phenotype to perform (economically) some target task. Ideally, we would like to specify the characteristics, competences, and tasks explicitly, using a high-level design language that is able to generate the description that will be inserted into the progenitor cell (or *stem cell*) of the self-constructing system. In this article we make some first steps toward such a process. Our overall approach is to constrain the developmental process to unfold in such a way that the earliest cell populations interact with one another to provide a physical infrastructure, within which the later, more task-related populations can be configured.

Our description describes a set of elementary intracellular mechanisms (factories) that support the self-construction of simple structures such as cell aggregations of specific sizes or segmentation of populations into two regions. These mechanisms can be used in a hierarchical ruleset to specify finer-grained structures. Branches of the ruleset tree represent substructures, which can be designed independently of the rest of the system. Through incremental application of the rules, a high degree of organismal complexity can theoretically be achieved [39].

Our multicellular construction process depends on local feedback of the unfolding structure to guide overall development. Much as in the development of natural organisms, the localized construction process is a function of its surrounding structure and the description residing in every cell. By writing morphogenic messages on their environment, the individual cells contribute to global morphogenic signals that other cells can use to modify their local developmental actions. This coupling can lead to global organization.

Overall, our self-constructing process can be seen as follows. The final organism is encoded in the description of the stem cell as a kind of state machine in which the states, once activated, become persistent. Each state corresponds to a population of cells that has a particular functionality by virtue of the particular set of elementary intracellular mechanisms that it expresses. Transitions between the states (and so the development of later cell populations) are triggered by local environmental conditions, which are themselves a function of which states have previously been activated.

In this article we write an explicit description for the example of a simple multicellular organism that expresses attractive or aversive foraging behavior analogous to a Braitenberg vehicle [6]. We demonstrate its self-construction and -repair, and evaluate its behavioral performance.

2 The Model

2.1 The Cell

The basic building block of our multicellular organism is a cell (Figure 1a). Each cell is equipped with a description and a constructor. The description is composed of genes that encode explicit specifications of individual intracellular factories, but do not explicitly specify the entire organism. The genes are activated by the local concentration of morphogens. Activation permits the constructor to read the gene, and so to construct the specific factories that correspond to that gene.

The cells interact with the environment through chemical diffusion of morphogens. The coefficients of diffusion through the membrane for the different chemicals are constant. Because we have chosen diffusion through the membrane to be passive in our model, the cells have no active control over which chemicals enter their interior from the environment. Once a particular cell is formed by replication, its further dynamics are determined entirely by its internal chemical concentrations and the functions of its expressed factories. They interact with the environment solely through their reactor dynamics. It is important to note that individual cells cannot perform complicated algorithm-like computations, nor can they communicate globally. It is only the interplay of the local dynamics of all the cells in the growing population that gives rise to the nontrivial structure of the organism.

2.1.1 The Description Code

The code specifying the rules for the construction of the organism is identical in each cell of the same organism. When the cells divide, an exact copy of the code is made and placed into the daughter cell. The description encodes:

- The definition of the lineage tree in the form of a differentiation graph that encodes the chemical conditions under which genes become active
- Which factories the constructor should build, depending on the cell type
- The specification of the factories (competences and reactors).

The structure of the organism is completely specified by these local criteria and depends on them. Because the code is identical in all the cells, the differentiation mechanism has to activate and deactivate parts of it, depending on the cell type. In principle, other types of organisms could be built by just applying another code.

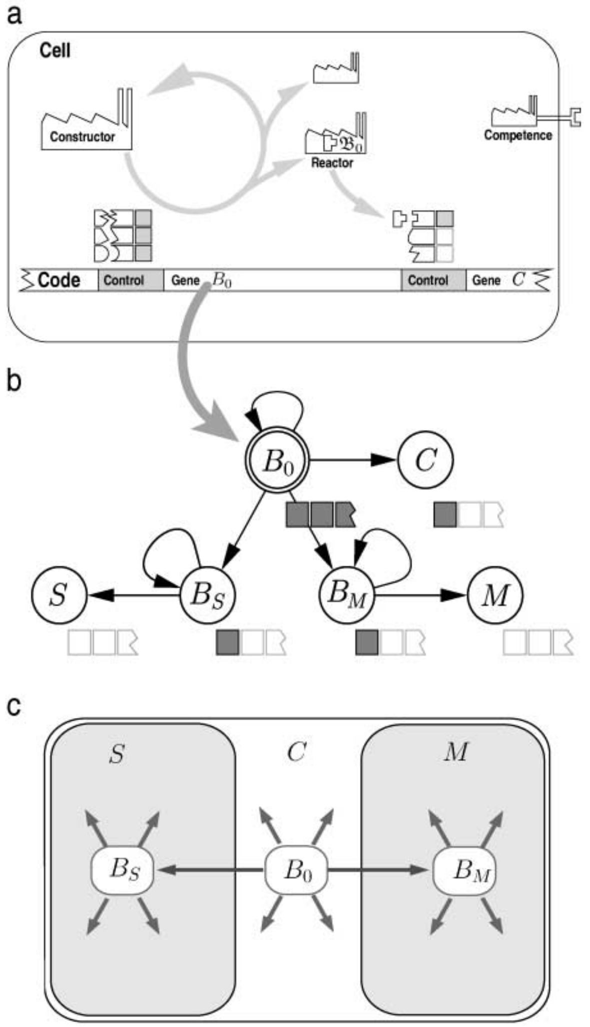


Figure 1. The cell, the description, and the physical instantiation of the description. (a) Schematic of a cell and its most important components: a description code and factories. The description is analogous to DNA; it consists of genes coding for specific sets of factories. Genes have control regions that encode the conditions required for their activation or suppression (small polygons denote intracellular factors; the conditions they match are indicated by a shaded square). Activated genes enable the construction of factories (B_0 in the figure). There are three types of factories: the *constructor*, which reads active genes and builds other factories; *reactor* factories, which act on intracellular morphogen concentrations; and *competence* factories, which implement cellular competences such as migration. Each gene corresponds to a particular cell type, and so the same symbol (B_0, C, etc.) denotes both the gene and its cell type. (b) The cell lineage tree: Nodes denote cell types, and edges the lineage paths between the types. Blast cells lie at the root of the tree, and specialization occurs centrifugally toward the fully differentiated leaves. Passage along the edges is controlled by the concentrations of intracellular factors and diffusible morphogens. The cell in the figure is expressing the factor $\mathfrak{B}_0$, which is one condition for activating the gene C. If, in addition, certain environmental conditions are also met (code in supplementary material [2]), then daughters of B_0 will follow the path from B_0 to C. (c) The body plan of the physical instantiation that is laid out by the description: Cells of the same types cluster together and build functional entities. The blast cells B_* form smaller populations and create and maintain their functional cell populations (for example, B_S cells give rise to the S cell population).

The code defines local objectives and actions in order to fulfill those objectives. We do not specify the explicit structure of the grown organism, but only describe what a single cell must achieve in order for the whole organism to assemble to the target structure. Because a cell can only act locally, the construction of the organism relies on global cooperative phenomena like those specified in Appendix A.1.1.

2.1.2 Modular Components within the Cell (Factories)

The cells can contain a set of modular functional components (factories) (Figure 1a). All factories are described in detail in Appendix A.1.

- *Constructor factory:* This factory initially reads the active gene in the description code and builds the corresponding factories associated with the active gene. The constructor factory is cloned into a newly born cell at cell division.
- *Chemical reactors:* These implement the intracellular mechanisms involved in the production or consumption of morphogens. They set up the global morphogenic gradients across the organism (AxisReactor, InterAxisReactor), and are also responsible for the production of morphogens (SourceReactor, ConstReactor) that are used for the maintenance of the preferred population size as well as intracellular signals.
- *Competences:* These express physical capabilities of cells, such as chemotaxis (MigrateCompetence), implementation of sensory or motor modalities (SensorCompetence and MotorCompetence), axonal growth (AxonCompetence), and cell division (DivideCompetence).

The activation of these cellular mechanisms depends on the description of the system and on the environmental context in which the cell resides. The central principle of our construction method is the appropriate coordinated instantiation of reactor factories. By contrast, the competence factories are merely convenient stubs, whose detailed implementation is not directly relevant to the process of self-construction that we describe in this article. Thus, in our example organism below, the MotorCompetence and SensorCompetence implement physical properties that give rise to the behavior of the final organism and are not directly involved in the self-construction and self-repair of the organism.

2.1.3 Differentiation Scheme

Differentiation of a cell is governed by its local environment, its description code, and its history. The description code is analogous to real DNA. It consists of genes that code for specific sets of intracellular factories. As in nature, the genes have control regions that encode the conditions for their activation or suppression [22, 25, 9]. These chemical conditions are simple thresholds that specify whether the gene is switched on or off (see Table 1 in Appendix A.2). If all its conditions are met, a gene is activated. The encoding regions contain the description of the factories to be instantiated. A newly born cell is initially undifferentiated, but inherits the chemical configuration of its mother cell. If a gene of the undifferentiated cell becomes active, the constructor reads its encoding region and constructs the required factories, so differentiating the cell. Once differentiated, the cell remains locked to its type, because it is defined by the intracellular factories it has constructed.

The conditions under which a gene is activated are programmed in the description to reflect the *developmental need* for a cell of a specific type. These conditions might only become true in a cell that is born at a specific moment in development and at a particular position.

Genes can also be regulated by chemicals that do not diffuse across the cell membrane. These chemicals are released from reactors within the cell. Because non-diffusible chemicals are local to the cell, they represent the state or the history of this cell. Non-diffusible chemicals are denoted by a fraktur font ($\mathfrak{B}_0, \mathfrak{B}_M, \mathfrak{B}_S$). These chemicals enable the activation of a specific gene in an offspring of the cell, because they, like all other chemicals, remain present in the interiors of daughter cells after cell division. This inheritance mechanism gives rise to a lineage tree (Figure 1b) where only a subset of cell types can arise from a given mother cell. The offspring subset of cell types for a given mother cell type expresses a requirement condition for the non-diffusible marker chemical of their mother cell type. Thus, the development of the entire organism from a single progenitor cell can be represented as a lineage tree in which the nodes denote cell types and the edges denote the developmental paths between the types. Specialization occurs centrifugally along edges of the lineage tree, allowing substructures on one branch to develop independently of those on other branches. For example, cells derived from the sensory blast cells, B_S, will set up and maintain the sensory system autonomously and independently of other subsystems (Figure 1c).

Cell replication occurs as follows: As described above, a gene whose conditions in the regulatory region are active expresses a *developmental need* for a cell of this type. Cells containing DivideCompetences will now become active. The DivideCompetence will sense active genes within its cell and consequently initiate a cell division. During a cell division, the chemical configuration of the mother cell is cloned inside the newly created cell. The mother cell remains of the same type, while the daughter cell is at first in an undifferentiated state. However, being subject to the same internal and external chemical conditions as its mother cell, the gene that triggered the cell division becomes active also in the daughter cell. The constructor factory of the daughter cell now reads the activated gene and instantiates the corresponding factories. It thus becomes of the expressed type, which is, by virtue of the non-diffusible markers, necessarily the same type or a subtype of the mother cell.

2.2 The Environments

The world in which the system is embedded is divided for convenience into two environments of different scale (Figure 2). The *local environment* is a two-dimensional lattice whose size is on the order of the organism's diameter and is used to model the immediate physical environment of the organism. The developing cells are located at the nodes of the lattice. Passive diffusion of morphogens occurs along the edges of the lattice, and also from the nodes into the interior of the cells. The natural neighborhood relation on the lattice provides the basis for cell adhesion and local migration of cells. Individual cells can only sense other cells located at neighboring sites, and they can only migrate along the edges of the lattice. In principle, multiple cells can be co-located on top of one lattice node. But, if they are of the same type, they are very likely to move to a free neighboring node because of constraints they must optimize (see Appendix A.1.2).

Various morphogens diffuse along the edges of the lattice. Each chemical species has its own diffusion coefficient. Diffusion is governed by the discretized diffusion equation:

$$c^i_{\mathbf{r}} \leftarrow c^i_{\mathbf{r}} + \tau D_i \sum_{\mathbf{r}\sim\mathbf{r}'} (c^i_{\mathbf{r}'} - c^i_{\mathbf{r}}), \tag{1}$$

where $c^i_{\mathbf{r}}$ is the concentration of chemical i at location $\mathbf{r}$, and $\mathbf{r} \sim \mathbf{r}'$ holds true for neighboring nodes $\mathbf{r}$ and $\mathbf{r}'$, and where D_i is the diffusion coefficient for chemical i, and τ is a small integration constant.

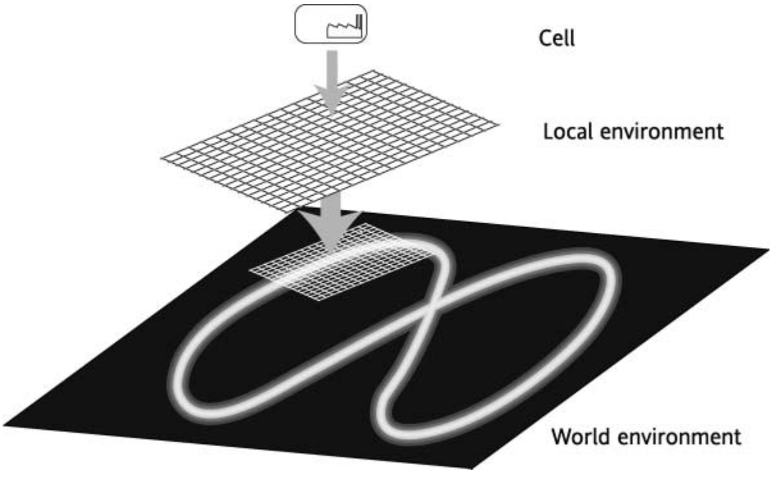


Figure 2. The two environments of the simulation. The cells reside on nodes of the lattice in the *local environment*. The local environment itself is embedded in the *world environment*, so that each cell receives a physical location in the world environment. Forces applied to the local environment result in movement and rotation of the local environment with respect to the world.

The 2D diffusing grid is itself embedded in the second, larger scale *world environment*. Goods are scattered in this larger world. Cells that express the SensorCompetence (see Appendix A.1.2) are able to sense the concentration of good at their location in the greater environment. The local environment (and so the whole population of cells) is moved with respect to the greater world environment by means of those cells that express the MotorCompetences. The MotorCompetences apply forces to the local environment. Together, they result in a single forward force $\mathbf{f} = \Sigma_i \mathbf{f}_i$ and a torque $M = \Sigma_i M_i$, which is applied to the center of gravity of the cell population. The translation $\mathbf{x}$ and rotation θ of the local environment is then computed by the following equations that approximate dynamics of a rigid body with strong friction:

$$m\dot{\mathbf{x}} = \mathbf{f}, \tag{2}$$

$$I\dot{\theta} = M, \tag{3}$$

where m and I are appropriately chosen constants.

3 Design and Development

The potential of our developmental scheme can be demonstrated by the self-construction and self-repair of a simple, but nontrivially behaving, functional organism. The final structure and function of this organism is similar to vehicle 2b (aggression) proposed by Braitenberg [6]. It contains a sensory input population and motor output population. Excitatory axons connect the left side of the sensory map to the the right motor cells, and vice versa. This connectivity yields an organism capable of targeting and seeking objects of sensory relevance in the environment.

3.1 Design of the Code

The final structure of the self-constructing organism is shown in Figure 1c. This structure is reflected in the description code (Figure 1b) in the following way: Each gene corresponds to a cell type, and each cell type is responsible for the setup and maintenance of a particular substructure of cells. In our example, we call cells that contain a DivideCompetence *blast cells*. They owe their name to their biological counterparts, which are incompletely differentiated progenitor cells that give rise to differentiated cells through asymmetric division. The skeleton of the hierarchical code consists of a tree of blast cells (B_0, B_M, and B_S). A blast cell constructs its appropriate substructure by generating a population of differentiated cells. B_0 are the topmost blast cells. They give rise to the population C of cells that are the scaffolding for the entire organism (Figure 1c). By placing AxisReactors (see Appendix A.1.1) in the specification of the C cells, a body axis forms within the population of the C cells and divides the organism into front and back. B_M and B_S cells govern the development of the motor and sensory substructures consisting of M and S cells, respectively. Their genes encode a MigrateCompetence that causes them to migrate to their corresponding extremities of the organism (motors in the back and sensors in the front). M and S cells only arise in an environment that has already been prepared by the higher level structure created by B_0 and C cells. The M cells contain AxisReactors to further divide into a left and a right subpopulation. The S cells contain an InterAxisReactor to form an antiparallel division to that of the M cells in order for the axons to be able to find the correct motor cells.

Each blast cell and its corresponding differentiated cells maintain their designated substructure of the organism independently. Self-construction and self-repair are therefore localized within these substructures.

For our example of a Braitenberg vehicle, we need two levels in the hierarchy: a global body plan on the top level, and a motor and sensory population on the second level. The code is provided in

Appendix A.2 in tabular form; for the exact parameter values see the supplementary material [2]. The tree structure is implemented by the intracellular chemicals $\mathfrak{B}_0, \mathfrak{B}_M,$ and $\mathfrak{B}_S$ and their corresponding activation conditions in the genes.

3.2 Development

This subsection describes the sequence of steps in the development from a single cell to the functional organism. Figure 3 illustrates the state of the organism and the state of the genes in a highlighted cell.

3.2.1 Figure 3a: Start of Ontogeny, Placing a Single Cell in the Environment

The development process begins when a single cell is placed in the environment. This cell contains the complete information needed for the self-construction of the organism. The environment has no specific chemical configuration. According to the graph on the left-hand side, the cell will differentiate into further B_0 cells (yellow). By virtue of their SourceReactor, these cells release morphogen b_0 into the environment.

The threshold $\widehat{b_0}$ of the B_0 condition defines the size of the B_0 population. Because the morphogens b_0 released by B_0 cells' SourceReactors diffuse away in the environment, the population must reach a critical size in order establish a stable chemical concentration. This size depends on the diffusion coefficient and the production rate of the morphogens. The critical size has the property that the production of chemicals inside the volume of the body is at the same rate as their diffusion away through the body's surface. A similar mechanism for the determination of the size of growing organisms was already noted by D'Arcy Thompson almost a hundred years ago [34].

The B_0 cells will form an initial population of stem cells, until conditions for C are reached ($b_0 > \widehat{b_0}$) and $B_0 \rightarrow B_0$ is switched off. The size of this initial population is not critical. However, the more stem cells an organism has, the faster it will grow. And with more stem cells, an organism is more likely to survive damage. If all stem cells are killed, the organism can not rebuild its primary structure.

3.2.2 Figure 3b: Dividing into Positional Structure Cells *C*

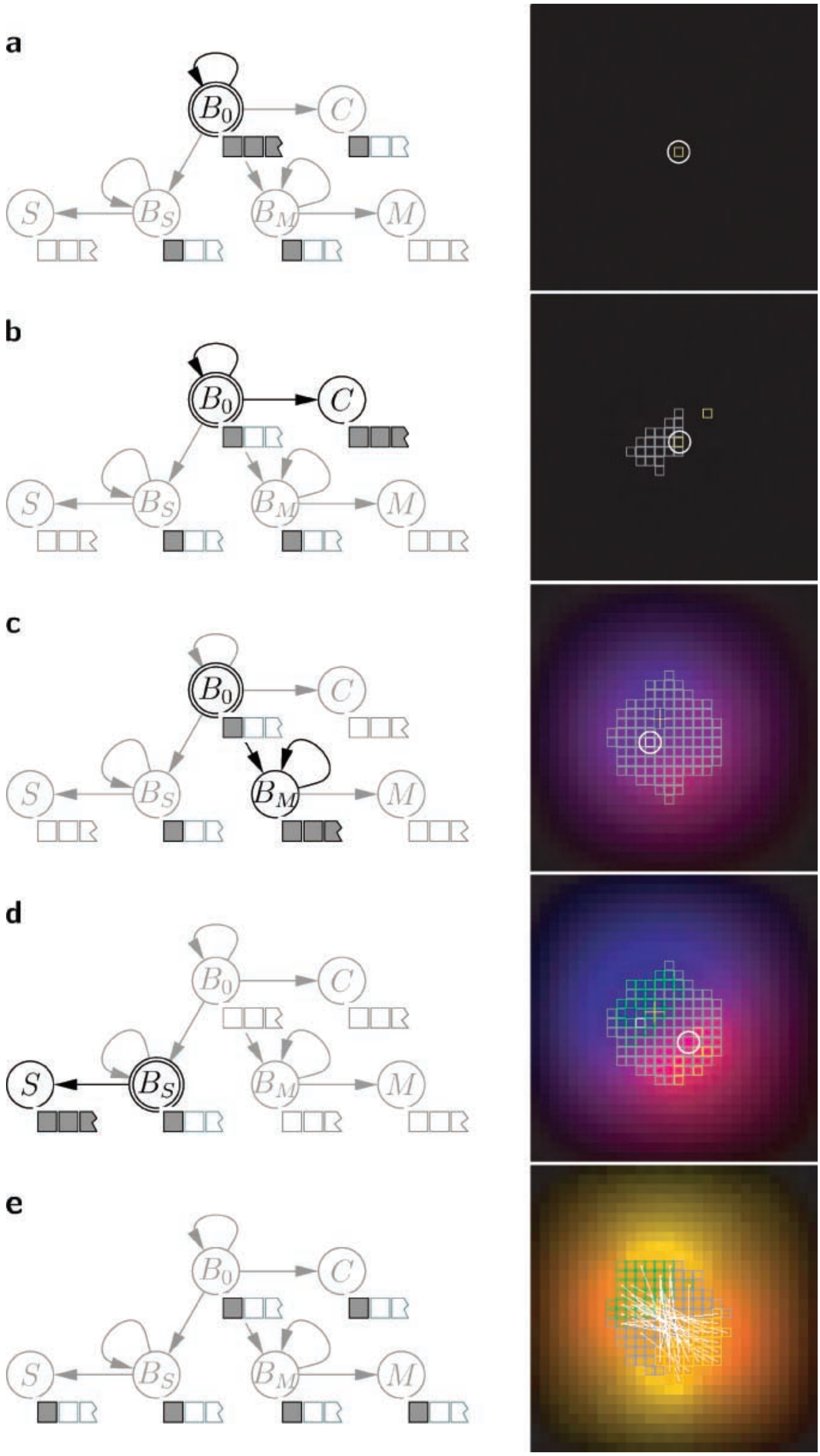
Once the B_0 population has been set up and the C conditions have been reached, the B_0 cells begin dividing into C cells. The same mechanism that determined the size of the population of B_0 cells will limit the size of the population of C cells: Their SourceReactors produce the marker chemical c, and the B_0 cells will stop dividing into C cells when their critical concentration according to the parameter $\hat{c}$ is reached within the population. As specified in the code, the C cells instantiate AxisReactors that will create the body axis.

3.2.3 Figure 3c: Setting Up the Positional Structure

At this point, the population of C cells has reached its critical size, and division into further C cells is switched off.

The AxisReactors in the cells of type C will set up a body axis by segmenting the population into two equal-sized clusters of minimal boundary length (see Appendix A.1.1). This segmentation happens as a cooperative effect among the population of identical AxisReactors and is not influenced by any external signals, except the noise in the environment to break the initial symmetry. While still composed of type C cells, the two clusters are defined by their chemical footprints: The back cluster is identified by high concentrations of g_1^C, while the front cluster has high concentrations of g_2^C.

Figure 3. (on facing page) The development of the behaving organism starting from one cell: On the right-hand side, the cell population in its developmental stage. The cells are depicted as outlined squares. Red and blue correspond to two chemicals g_1^C and g_2^C important for the setup of the positional structure. The ranges are scaled to suit the illustration. The white lines in the bottom panel depict axons; the white squares at the end of the axon are the axon terminals. The left-hand side depicts the differentiation graph corresponding to the circled cell on the right. The type of the cell is depicted by the double circled node. The shaded nodes represent cell types that are not being produced at this point.


a
B_0
C
S
B_S
B_M
M
b
B_0
C
S
B_S
B_M
M
c
B_0
C
S
B_S
B_M
M
d
B_0
C
S
B_S
B_M
M
e
B_0
C
S
B_S
B_M
M

According to the conditions for B_M ($g_1^C > \theta$) and B_S ($g_2^C > \theta$), when the clusters have completely formed, the B_0 cells will start to divide into B_M (blue) and B_S (orange) cells, depending on their environment. B_M cells are motor neuroblast cells that generate the motor units, while B_S cells are sensory neuroblast cells that generate the sensory units. The MigrateCompetence allows these specialized blast cells to migrate to the edge of the population, driven by repulsion from the opposing chemical (B_M cells are repelled from chemicals released by C cells in the sensory cluster, and vice versa). That is, motor neuroblast cells migrate to the back while sensor neuroblast cells migrate to the front.

3.2.4 Figure 3d: Preparing to Develop Motor and Sensory Units

The initial population size of motor blast cells has been reached and the conditions for M become true. Consequently, the production of motor cells M (green) has started. The motor and sensory cells will also contain (Inter)AxisReactors and hence form a left-right segmentation within their population. Because AxisReactors segment a population along its longer principal axis and the M population is elongated orthogonally to the body axis, the left-right segmentation will be orthogonal to the global asymmetry.

The sensory subpopulation will form a gradient antiparallel to the motor subpopulation by virtue of the InterAxisReactor: The marker morphogens g_1^M and g_2^M of the motor population influence the orientation of the segmentation of the sensory subpopulation into g_1^S and g_2^S (see Appendix A.1.1).

3.2.5 Figure 3e: The Developed Organism

Once all populations have been established, all genes become inactive and no further cell division takes place. The AxonCompetences in the sensory cells grow axons to reach the motor cells by a simple wiring strategy described in the code: They follow the corresponding marker morphogen—yellow sensory cells target yellow motor cells, and vice versa (see Appendix A.1.2).

The structure of the organism is now in stable equilibrium in the sense that the blast cells have all become quiescent and no longer divide, and all axons have grown to their target. The organism has now finished its development and is behaving as expected (Figure 4).

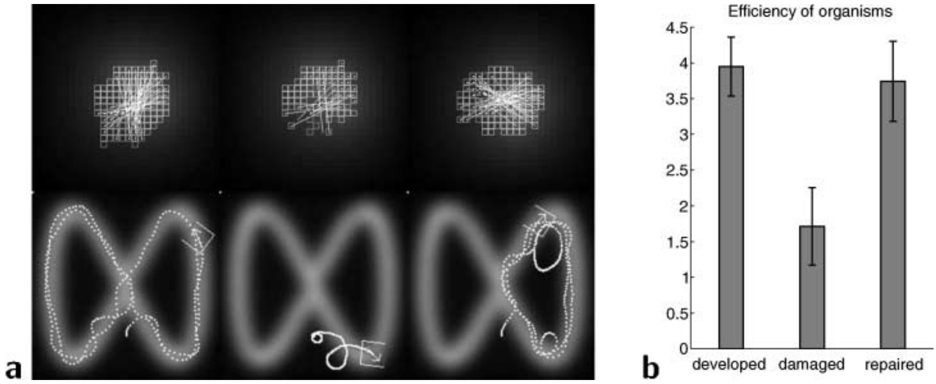


Figure 4. Performance of the organism, and its recovery following damage. (a) Left, top: A fully developed organism in its local environment, at the end of a development sequence (age 120,000 steps). Left, bottom: The organism in its greater environment. The local environment is indicated by the small square. The path of the organism is indicated by white dots at equal time intervals. The simple sensorimotor organization of the organism allows behavior: The organism is able to follow the good track (8-shape) for an extended amount of time. Middle: The damaged organism after removal of a patch of cells: Imbalance in connectivity degrades the performance. Some genes become again active and replenish the damaged cells, and, by the same principles of development described above, the structure and function of the organism is restored. Right: The organism after self-repair (70,000 steps after damage). All the local conditions are again satisfied. The various cell populations and their organizations have been restored. Performance recovers. (b) Mean and standard deviation of the efficiency [(distance traveled)/(goods gathered) in 30,000 steps] for 200 simulated organisms. After damage, the organisms recover.

4 Behavior and Self-repair

The development leads to an agglomeration of different cell types that are organized spatially in a configuration that can support a behavioral function. In this case, the organism's minimal sensorimotor organization is similar to Braitenberg's classical *aggression* vehicle [6], which is attracted to a stimulus such as light or food. We chose a simple task, tracking *goods*, to evaluate the behavioral competence of the mature organism (Figure 4). The sensory cells transduce the local concentration of goods into neuronal activity. This signal propagates along the axon and excites the postsynaptic motor cell. Excited motor cells apply a force in the direction of the longitudinal morphogen gradient of the organism, resulting in a momentum and torque of the compound population.

The distributed construction process not only drives the development of the organism but also permits it to repair itself. When the organism is damaged, the quiescent blast cells near the site of injury sense changes in the concentrations of the morphogens of the destroyed cells. The differentiation conditions in the nearby blast cells will be reactivated, and the cell division and differentiation process will restore the local structure. This process occurs only at locations where damage is and continues only until the cell populations have been restored to their original size and organization. When the equilibrium state is reached, the blast cells become quiescent again. It is this natural reactivation of quiescent blast cells that allows the robust development and repair.

The performance of goods tracking was measured as the *mean efficiency* of the organism, given by the ratio between distance traveled and goods encountered in a trial of 30,000 steps (Figure 4b). Well-developed organisms transform the environmentally mediated sensory activity into an overall coherent forward movement along the path. After a development period (age 120,000 steps), we damaged the organism by removing a circular patch of cells with a radius of about 4 cells. This damage was applied in a random location in the periphery of the organism, destroying a significant fraction of the sensory or motor region. Damaged organisms are unable to control their movement, and hence their performance is significantly worse than that of the mature organism. However, the organism has repaired itself after 70,000 steps, and its foraging performance has recovered. A movie of the development, behavior, and repair of a sample organism is available from the supplementary material [2].

5 Summary and Discussion

In this study, we have described a method for the design of self-constructing and self-repairing organisms. There has been considerable previous research on this topic (see, e.g., [23] and [32] for overviews). In particular, the importance of genotype-phenotype mapping has been discussed in the context of experimental and theoretical biology, as well as in simulations of artificial life.

Recent examples of artificial developing multicellular systems have emphasized the feedback interaction between a genetic description and environmentally derived signals, so elaborating the original von Neumann simplification of an entirely feedforward description. Various gene/environment models have been tested. Dellaert and Beer [10] used random Boolean networks to describe the genetic dynamics, whereas Fleischer and Barr [14] described the dynamics by differential equations. Eggenberger [12], and Bongard and Pfeifer [5] have used models of genes that incorporate regulatory regions and differential gene expression, similar to ours. The role of environmental feedback and homeostasis in gene regulation networks has also been described in a study by Quick et al. [31]. They demonstrated the evolution of a genetic control system that regulates environmental signals. The behavior of their system emerges through immediate regulatory reactions to the current state of the environment.

In general, these previous implementations have relied on genetic algorithms to discover suitable genetic instructions. Besides the inherent resource problems entailed by genetic search, the resulting instructions are usually phenomenological and so lack functional explanation. By contrast with those studies, our approach explores how genetic instructions for particular well-defined cellular mechanisms could be designed, and programmed to implement an entire developmental process. Figure 3 shows how our organism unfolds according to a specific plan that can be encoded as a kind of state machine. This formal description as state machine raises the exciting possibility that self-assembling organisms could in future be designed efficiently, rather than relying on expensive search.

Control of pattern formation is crucial to the design of self-assembly. The formation of global patterns in biological systems has been explained by means of reaction-diffusion systems [36, 17] that use only local interactions. A similar phenomenon has been observed in the behavior of social insects [18, 4]. Grassé described how individual animals can use structures previously built by the collective society to guide their individual further construction actions, rather than following simple sequential recipes. He called this behavior *stigmergy*: "The worker does not guide his work, he is directed by it" [19]. Up to now, the concept of stigmergy has been applied mostly to multi-agent systems in the sense that the agents alter their environment, which in turn feeds back onto their behavior [33, 19]. We use stigmergy in the sense that the whole cell population *is* the environment of the cell, and the local structure of the population guides the action of every single cell. Our genes can be regarded as *stigmergic* rules. The fully developed organism is the equilibrium state of this decentralized construction process. Because it is a stable equilibrium, perturbations of the population structure will be restored through local actions, so providing self-repair. The final organization of the organism is the equilibrium state of the population of cells. The organization maintains itself by homeostasis, a property that it shares with all living systems [27].

Self-repair and regeneration of tissue in multicellular systems has been described by Furusawa and Kaneko [16] and Miller [30, 23] in simulations of cellular models. However, neither of those studies explains what the essential components of self-repair are, nor how this property should be implemented in the genetic code, as we have done here. Furthermore, Miller's system can not downregulate its growth process, and so it grows ad infinitum, because it is only partially sensitive to its larger structure.

von Neumann conceived system construction as direct feedforward translation of a description contained in the parent into a daughter instance by the constructor machinery. More recent studies of self-construction, such as those reviewed above, have taken inspiration from development in biology, which depends on feedback between the expression of genetic description and the environment. Because of the perceived complexity of this process, those authors have relied on evolutionary algorithms to find a suitable instruction code. Our principle contribution has been to demonstrate that the biological-style development can be programmed explicitly. That is, the use of genetic algorithms could be replaced by principled design, while preserving the environmental sensitivity of the construction process.

To achieve this, we modularized the elementary construction processes in such a way that they can be conveniently composed under genetic control. In particular, we have demonstrated how to exploit cooperative organizational phenomena by explicit description to achieve a global target structure that is capable of a behavioral function. In addition to its self-construction, our system is able to repair itself by the very nature of its construction: The self-construction and self-repair are equivalent mechanisms. The principal components necessary for the presented scheme of self-construction are: (1) a physical cell capable of self-reproduction, (2) a differentiation scheme that is influenced by the history of the cell and its local environmental conditions, and (3) modular components that are conditionally activated and can alter the immediate environment of the cell and thus feed back to the differentiation process. It is sufficient that the cells can communicate only by means of passive diffusion of chemical signals. In our model, we have been careful to utilize only local criteria and actions that are in principle realizable in physical systems.

In its present form, our construction process is not universal in the sense of being able to construct any arbitrary structure. As for natural systems, it is very unlikely that any arbitrary artifact can be designed to self-construct. The developmental process and its corresponding description in the gene code constrains the space of possible stable functional configurations. However, we introduce a general mechanism for designing structures that could allow a variety of different target structures. A finer-grained structure than the one shown here can be obtained by further subdividing regions by means of AxisReactors and by providing blast cell types lower in the hierarchy that control the finer-grained development superimposed on the coarser-grained structure established by the earlier development process.

In contrast to traditional external fabrication, self-construction does not employ a global external observer to supervise the assembly process. Nonetheless, we suggest that it is necessary that local processes be sensitive to the global state of the system. We achieve this interaction through cooperative effects like symmetry breaking within the population and a morphogen-sensing mechanism that regulates the size of the population. These locally generated signals arising from global emergent structures reflect qualities of the whole structure and make them accessible to localized processes. Hence, they can be utilized to drive the local development.

Furthermore, we propose a hierarchical structure of the cell-type lineage, such that substructures can grow and maintain themselves independently of the rest of the system, while utilizing structures already set up by earlier processes (higher up in the hierarchy). This is a powerful concept that allows for directed engineering and refinement of these components, while leaving their self-construction and maintenance their own responsibility. This orthogonality of subsystems provides a modularity that facilitates development and repair. In evolving systems, independence between developmental modules also allows more robust evolution by local refinement of subsystems [37].

Novel methods will be required for writing the appropriate description code that will allow development to unfold toward a desired phenotype. Previous studies have relied on genetic algorithms to find the correct genotype, whereas in our scheme the emergent structure is programmed by appropriately choosing the set of cell types and their corresponding factories, thus yielding a cascade of developmental actions with well-defined goals. We have used a hand-designed gene code to demonstrate that we understand how the genetic code within a single cell interacts with the developing organism to generate the specific global structure we planned to build. To achieve a wider range of possible phenotype configurations, future studies should identify and understand more comprehensively the basic developmental modules required for programmable self-constructing and self-repairing systems. Further studies should explore the feasibility of a “soft” configuration language that could be used to directly specify the lineage tree, and so the space of possible phenotypes. Instead of planning systems in terms of construction and assembly, as in traditional engineering, this novel design procedure should aid the specification of local, cellular conditions under which the functional organization of the system is in equilibrium. Then, the final organization is attained by activating the developmental process.

The present work will serve as a foundation for the exploration of the relationship between ontogenesis and learning. So far we have used only structurally induced effects to influence the construction of the system. Further structural and functional complexity could be acquired by the development process through its constant feedback with a complex environment. In the future we will include functional effects into the developmental process (for example, sensory input could help the formation of retinotopic maps through functional dependence of axonal growth). Such models could be used to quantify the balance between prespecification and learning or to illustrate how complexity can be acquired from the environment in a developmental process.

Principles of self-construction and -repair are relevant not only for biology, but also for the design and construction of advanced software, machines, and buildings. Distributed sensitivity to its very own structure and local goal-directed behavior of components will find applications everywhere where systems are embodied in a real world and interfaces cannot be well defined a priori. In order to apply our hierarchical construction approach more broadly, we will need to abstract the organizational processors (reactors) and the messages (morphogens) that they emit.

A physical implementation of such systems in mechanical technology seems still far ahead. However, recent advances in automated design and construction [24] and in the field of self-assembling robots [41] show that in principle, mechanical self-construction is possible. Biological implementation might be in reach sooner: Cells have been stripped down to their essential components and genes necessary to survive [20]; and there are chemical models for biological self-assembling multicellular systems [21]. In the emerging field of *synthetic biology*, attempts are being made to catalogue simple gene sequences and their functions in order to later assemble them into complex circuits in synthesized DNA [1]. Recently, genetic engineering has achieved successes in this direction [7, 35, 3].

Acknowledgments

We acknowledge the helpful comments and suggestions of Jorg Conradt, Armin Duff, Dylan Muir, Bobby Rohrkemper, Daniel Rubin, and Frederic Zubler.

References

1. Registry of standard biological parts. http://parts.mit.edu/.
2. Supplementary material. http://www.alife.org/publications.html#rothetal.
3. Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., & Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature*, *434*(7037), 1130–1134.
4. Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Franks, N. R., Rafelsberger, O., Joly, J., & Blanco, S. (1998). A model for the emergence of pillars, walls and royal chambers in termite nests. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *353*(1375), 1561–1576.
5. Bongard, J., & Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In H. Beyer, et al. (Eds.), GECCO 2001: Proceedings of the Genetic Evolutionary Computation Conference (pp. 829–836). San Mateo, CA: Morgan Kaufmann.
6. Braitenberg V. (1984). *Vehicles: Experiments in synthetic psychology.* Cambridge, MA: MIT Press.
7. Bulter T., Lee S. G., Wong W. W., Fung E., Connor M. R., & Liao J. C. (2004). Design of artificial cell-cell communication using gene and metabolic networks. *Proceedings of the National Academy of Sciences of the USA*, *101*(8), 2299–2304.
8. Chaitin G. J. (1969). On the length of programs for computing finite binary sequences: Statistical considerations. *Journal of the ACM*, *16*, 145–159.
9. Cherry J. L., & Adler F. R. (2000). How to make a biological switch. *Journal of Theoretical Biology*, *203*(2), 117–133.
10. Dellaert F., & Beer R. (1994). Toward an evolvable model of development for autonomous agent synthesis. In R. Maes & P. Maes (Eds.), *Artificial Life IV.* Cambridge, MA: MIT Press.
11. Drexler, K. E. (1986). *Engines of creation: The coming era of nanotechnology.* New York: Anchor Books.
12. Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In P. Husbands & I. Harvey (Eds.), *Proceedings of the 4th European Conference on Artificial Life (ECAL97).* Cambridge, MA: MIT Press.
13. Feynman, R. (1959). There's plenty of room at the bottom. http://www.zyvex.com/nanotech.feynman.html.
14. Fleischer, K., & Barr, A. H. (1994). A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In C. G. Langton (Ed.), *Artificial Life III* (pp. 389–416). Reading, MA: Addison-Wesley.
15. Freitas, R. A., Jr., & Gilbreath, W. P. (1980). *Advanced automation for space missions.* Proceedings of the 1980 NASA/ASEE Summer Study. NASA.
16. Furusawa, C., & Kaneko, K. (1998). Emergence of multicellular organisms with dynamic differentiation and spatial pattern. *Artificial Life*, *4*(1), 79–93.
17. Gierer, A., & Meinhardt, H. (1972). A theory of biological pattern formation. *Kybernetik*, *12*(1), 30–39.
18. Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubiterms* sp. La théorie de la stigmergie: Essai d'interprétation du comortement des termites constructeurs. *Insectes Sociaux*, *6*, 41–83.
19. Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, *5*, 173–202.

20. Hutchison, C. A., Peterson, S. N., Gill, S. R., Cline, R., White, O., Fraser, C. M., Smith, H., & Venter, J. (1999). Global transposon mutagenesis and a minimal Mycoplasma genome. *Science*, *286*(5447), 2165–2169.

21. Jakab, K., Neagu, A., Mironov, V., Markwald, R. R., & Forgacs, G. (2004). Engineering biological structures of prescribed shape using self-assembling multicellular systems. *Proceedings of the National Academy of Sciences of the USA*, *101*(9), 2864–2869.

22. Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford, UK: Oxford University Press.

23. Kumar, S., & Bentley, P. J. (Eds.) (2003). On growth, forms and computers. London: Elsevier Academic Press.

24. Lipson, H., & Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, *406*(6799), 974–948.

25. Mannervik, M., Nibu, Y., Zhang, H., & Levine, M. (1999). Transcriptional coregulators in development. *Science*, *284*(5414), 606–609.

26. Marée, A. F., & Hogeweg, P. (2001). How amoeboids self-organize into a fruiting body: Multicellular coordination in *Dictyostelium discoideum*. *Proceedings of the National Academy of Sciences of the USA*, *98*(7), 3879–3883.

27. Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and cognition, the realization of the living*. Dordrecht, The Netherlands: D. Reidel.

28. Mazzola, L. (2003). Commercializing nanotechnology. *Nature Biotechnology*, *21*(10), 1137–1143.

29. Meinhardt, H., & Gierer, A. (1980). Generation and regeneration of sequence of structures during morphogenesis. *Journal of Theoretical Biology*, *85*(3), 429–450.

30. Miller, J. (2004). Evolving a self-repairing, self-regulating, French flag organism. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*. Berlin: Springer-Verlag.

31. Quick, T., Nehaniv, C. L., Dautenhan, K., & Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In W. Banzhaf et al. (Eds.), Advances in Artificial Life: 7th European Conference, ECAL 2003 (pp. 266–277). Berlin: Springer-Verlag.

32. Stanley, K. O., & Miikkulainen, R. (2003). A taxonomy of artificial embryogeny. *Artificial Life*, *9*, 93–130.

33. Theraulaz, G., & Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, *5*, 97–116.

34. Thompson, D. W. (1961). *On growth and form*. Cambridge, UK: Cambridge University Press.

35. Tsiavaliaris, G., Fujita-Becker, S., & Manstein, D. J. (2004). Molecular engineering of a backwards-moving myosin motor. *Nature*, *427*(6974), 558–561.

36. Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London, B*, *237*, 37–72.

37. von Dassow, G., & Munro, E. (1999). Modularity in animal development and evolution: Elements of a conceptual framework for EvoDevo. *Journal of Experimental Zoology*, *285*(4), 307–325.

38. von Neumann, J. (1966). *Theory of self-reproducing automata*. Urbana, IL: University of Illinois Press.

39. Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D*, *10*, 1–35.

40. Zheng, W., Buhlmann, P., & Jacobs, H. O. (2004). Sequential shape-and-solder-directed self-assembly of functional microsystems. *Proceedings of the National Academy of Sciences of the USA*, *101*(35), 12814–12817.

41. Zykov, V., Mytilinaios, E., Adams, B., & Lipson, H. (2005). Robotics: Self-reproducing machines. *Nature*, *435*(7039), 163–164.

Appendix

A.1 The Modular Factories

The following factories are used by the developmental process. The mechanisms offered by the factories are quite general. In this section we will describe them and how they are used in our example organism.

A.1.1 Reactor Factories

The reactor factories implement a set of differential equations that express the cellular chemodynamics.

ConstReactor This reactor maintains the concentration of a specific chemical c_i at $\hat{c}$ by the production rule

$$\dot{c}_i = \alpha \cdot (\hat{c} - c_i). \tag{4}$$

In our model we have used this reactor to maintain the concentration of a nondiffusible marker chemical that characterizes its cell type. Every cell contains a reactor of this type.

SourceReactor The SourceReactor produces chemical c_i at a constant rate r:

$$\dot{c}_i = r. \tag{5}$$

In our model, we have used SourceReactors to maintain the sizes of the various cell populations. Within a population, all cells release their characteristic diffusible marker at a constant rate. Because the volume of any cell population grows more rapidly than its enclosing surface, there will come a time when the overall production of the marker within the population exceeds the overall outward diffusion of the marker across the enclosing surface of the population. At this time, the concentration of the marker within individual cells begins to rise, and this signal is used to inhibit further cell division.

AxisReactor AxisReactors are used to establish the spatial organization of cell populations (see Figure 5a). They do so by creating gradients of morphogens, whose concentrations provide signals for the conditionally activation of various processes in other cells.

AxisReactors are based on rate equations proposed by Meinhard and Gierer [29]. A given AxisReactor generates chemical dynamics within each cell that lead to the formation of opposing gradients of two morphogens g_1 and g_2 across an entire population of cells. These gradients provide an axis for development. The gradients arise by a cooperative-competitive process: local competition

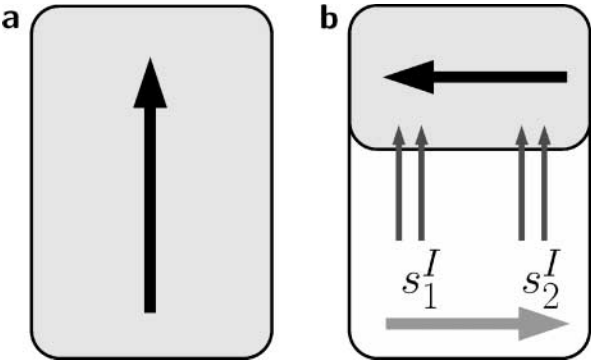


Figure 5. The AxisReactor and InterAxisReactor create a global gradient across the population they are instantiated in. (a) The AxisReactor is instantiated in all the *C* cells and thus creates a gradient across the whole organism. This two-state structure is used to determine front and back of the organism. (b) The InterAxisReactor is instantiated in the sensory cells and uses the influence of the motor cells to determine its direction.

between the production of morphogens g_1 and g_2, and long-range cross-facilitation of their production by two additional signal chemicals s_1 and s_2. This process follows the dynamics

$$\dot{g}_1 = \frac{cs_2}{a + g_2^3} - \alpha g_1, \tag{6}$$

$$\dot{g}_2 = \frac{cs_1}{a + g_1^3} - \alpha g_2, \tag{7}$$

$$\dot{s}_1 = \gamma(g_1 - s_1), \tag{8}$$

$$\dot{s}_2 = \gamma(g_2 - s_2), \tag{9}$$

where c is the gain of the facilitation, α is a temporal decay, γ is the gain for the production of the cross-facilitators, and a is an arbitrary constant.

Populations of cells employing the following chemodynamics will stabilize into a two-state structure along the longitudinal axis, where the states are characterized by high g_i and low g_j concentrations for $i \neq j$.

The diffusion coefficient for the long-range cross-facilitators s_1 and s_2 must be higher than that of the short-range competitive chemicals g_1 and g_2. Here g_1 gives rise to the production of s_1, which in turn will cross-facilitate g_2 over a longer distance, and vice versa for g_2 and s_1. The slow diffusion of the g's will influence neighboring cells to belong to the same cluster, whereas rapidly diffusing long-range facilitation chemicals s_1 and s_2 break homogeneity along the axis.

InterAxisReactor InterAxisReactors are used to create spatial organizations with respect to already established gradients. For this purpose, the InterAxisReactors are similar to the AxisReactors but contain an interaction term in the production rule of g_1 and g_2:

$$\dot{g}_1 = \frac{cs_2(\delta + g_1^I)}{a + g_2^3} - \alpha g_1, \tag{10}$$

$$\dot{g}_2 = \frac{cs_1(\delta + g_2^I)}{a + g_1^3} - \alpha g_2, \tag{11}$$

$$\dot{s}_1 = \gamma(g_1 - s_1), \tag{12}$$

$$\dot{s}_2 = \gamma(g_2 - s_2), \tag{13}$$

where g_1^I and g_2^I are the chemical concentrations of the interacting morphogens, and δ is the interaction baseline. If no interaction morphogens are present, these dynamics coincide with the dynamics of the AxisReactor with a facilitation gain $c\delta$. Here δ should be tuned so that the gain is not high enough for segmentation to occur when there is no influence from g_1^I and g_2^I. This way we can ensure this segmentation will wait until the influencing gradient has been set up.

In our example the InterAxisReactor is instantiated by the sensory cells and establishes an antiparallel gradient with respect to the gradient across the motor cells (see Figure 5b). This antiparallel structure is later used to support the crossed sensorimotor connectivity of the growing axons.

A.1.2 Competence Factories

The cells can express predefined competences. Like the chemical reactors, the competences are built by the constructor when the cell differentiates.

DivideCompetence If a cell expresses the DivideCompetence, it can divide asymmetrically into two cells. However, this division occurs only if the differentiation graph has an active edge signaling an environmental need for a certain type of cell (i.e., a gene becomes active). Division will occur with a predefined low probability, which keeps the production rate low. After a binary cell division, one daughter cell remains the mother cell while the other daughter cell will undergo differentiation beginning from the state of its mother and then following the differentiation graph as discussed in Section 2.1.3. This daughter cell can remain of the same type, or become a more specialized type.

MigrateCompetence Cells that express the MigrateCompetence can change their location on the grid. They attempt to optimize their positional objective function, which is calculated from the number of neighbors and the concentration of chemicals in the environment. A cell will try to maximize the number of its neighbors, while attempting to reside alone on a grid point. A term that depends on the chemicals in the surrounding environment encourages the cell to migrate according to its chemical affinity. The description code describes which chemicals a cell of a certain type is attracted to and which chemicals repel it.

Each cell p attempts to minimize its free energy

$$H_{\mathbf{r}} = \sum_{\mathbf{r}\sim\mathbf{r}'} J(s_{\mathbf{r}}, s_{\mathbf{r}'}) + S(s_{\mathbf{r}}) + J_p(\mathbf{c}_{\mathbf{r}}), \tag{14}$$

where $\mathbf{r}$ is the location of the cell p, and $J(s_{\mathbf{r}}, s_{\mathbf{r}'})$ describes the binding energy of two neighboring nodes at $\mathbf{r}$ and $\mathbf{r}'$ and may take the values $-\epsilon$ if both sites are occupied by at least one cell, or 0 otherwise. $S(s_{\mathbf{r}}) = \lambda[(\#\text{ cells of same type at }\mathbf{r}) - 1]$ expresses the stress of a cell at node $\mathbf{r}$. Here λ is large compared to ϵ, so that cells of the same type compete for a space on a particular node. $J_p(\mathbf{c}_{\mathbf{r}})$ expresses the affinity of a cell for the node's morphogen configuration. J_p depends on the cell type and is configured by the MigrateCompetence (specified in the description). A migrational step of a cell is performed according to a Monte Carlo algorithm similar to the cell sorting algorithm employed by Marée and Hogeweg [26]: The cells arrange themselves so as to minimize the free energy of the whole cell population through local optimization. At each active migrational step, a cell picks a random neighboring target node on the lattice and calculates the energy difference ΔH between the current configuration and the configuration after a hypothetical migrational step to the chosen node. The new configuration is accepted if $\Delta H \leq 0$. If $\Delta H > 0$, the new configuration is accepted with probability $P = \exp(-\Delta H/T)$, where T is the temperature of the process and measures the degree of migrational fluctuation of the cells. T is kept constant throughout the course of a simulation. This will result in a fluctuating equilibrium structure. We chose not to use annealing, so that the system is able to react quickly to perturbations and to reorganize if necessary, without having to modulate individual temperature parameters.

AxonCompetence In order to achieve macroscopic behavior, the growing organism needs to build functional components. To support function, axons can signal neural activity of their originating cell to remote cells. A cell expressing the AxonCompetence grows axons constrained by attraction and repulsion to specific morphogens. Axons form synapses with the cells to which they are connected, and so transfer their neural activity to their postsynaptic cell. The growth of the axons is guided by the same mechanism as the migration of the cell through the MigrateCompetence. A growth cone at the tip of the axon optimizes its objective function by migration according to a Monte Carlo process.

As specified by the code, the objective function encodes the optimal chemical environment and includes a competition among axon terminals on the same postsynaptic cell. This way, axon terminals will distribute homogeneously within the target environment. During the migration of the growth cone, the axon grows or retracts accordingly.

SensorCompetence The SensorCompetence senses the concentration of good at its cell in the world environment. The concentration produces a proportional cellular activity that can be transmitted via an axon.

MotorCompetence The MotorCompetence applies a force proportional to the cells' activity to the whole organism (see Section 2.2). The direction of the force with respect to the body orientation is specified in terms of chemical gradients and read from the description code. The morphogens of the global positional gradient (AxisReactor) are used to determine the direction of the force.

The morphogenesis of the actual muscle or effector that forces a movement of the organism is omitted in our model for simplicity. It is important to note, however, that in principle this morphogenesis could be simulated. We have a clearly identified region of the organism that is responsible for movement, and the specialized motor stem cells could be programmed to further divide into more specific cells that compose an effector.

A.2 Description Code

Table 1 shows the description code in tabular form. The upper part of the table describes conditions on morphogens for a particular gene or cell type to become activated: The expressions in the table denote the comparison of the morphogen concentration to some constant. Once the gene is activated (all conditions are true) and the cell has differentiated, the corresponding factories (lower part of the table) are instantiated. The entries in the lower part are instantiation parameters to the factories and represent symbols for different morphogens. A check mark signifies that this competence is instantiated. For MigrateCompetence, + and − defines whether this cell is attracted or repelled by the corresponding morphogen. The AxonCompetence takes as an argument a list of morphogen to morphogen mappings. A mapping $a \Rightarrow b$ signifies that a cell with high internal a concentration sends its axons to locations with high b concentration. The reactors and competences are explained in Appendix A.1. For example: If the morphogen concentration of $\mathfrak{B}_0$ is greater than η, and that of b_0 is greater than $\hat{b}_0$, but the concentration of c is less than $\hat{c}$, then the cell differentiates into a cell of type C. Once differentiated, the constructor will instantiate the following factories:

- A MigrateCompetence with symbolic argument $+c$. Cells with this competence will migrate towards higher concentrations of c in the environment.
- A ConstReactor with argument $\mathfrak{C}$, keeping the intracellular concentration of $\mathfrak{C}$ constant and greater than zero (in blast cells this regulatory factor is used to determine the branch of the lineage for its descendant cells).
- A SourceReactor for the morphogen c, which produces c at a constant rate and so signals to neighboring cells its existence and allows them to migrate accordingly.
- Finally, an AxisReactor that will be responsible for the creation of the g_1^C, g_2^C gradient, which in turn will be used to activate the genes B_M and B_S, which are conditioned on g_1^C and g_2^C.

The tree structure of the code is implemented by means of the non-diffusing chemicals $\mathfrak{B}_0$, $\mathfrak{B}_M$, and $\mathfrak{B}_S$. For example: A cell of type B_0 includes a ConstReactor for $\mathfrak{B}_0$. Therefore only the genes B_0, C, B_M, and B_S can be activated in a cell of type B_0, and thus a B_0 cell can only divide into cells of type B_0, C, B_M, and B_S, not into more specialized cells of type M or S. This reflects the lineage of the code depicted in Figure 1 (bottom).

Table 1. Description code in tabular form.

Cell types	B_0	C	B_M	B_S	M	S
Morphogens:						
$\mathfrak{B}_0$	$> \eta$	$> \eta$	$> \eta$	$> \eta$		
$\mathfrak{B}_M$					$> \eta$	
$\mathfrak{B}_S$						$> \eta$
b_0	$< \widehat{b_0}$	$> \widehat{b_0}$	$> \widehat{b_0}$	$> \widehat{b_0}$		
c		$< \hat{c}$				
b_M			$< \widehat{b_M}$		$> \widehat{b_M}$	
b_S				$< \widehat{b_S}$		$> \widehat{b_S}$
m					$< \widehat{m}$	
s						$< \widehat{s}$
g_1^C			$> \theta$			
g_2^C				$> \theta$		
Factories and parameters:						
Competences:						
Divide	✓		✓	✓		
Migrate	$+b_0$	$+c$	$+g_1^C, -g_2^C$	$+g_2^C, -g_1^C$	$+g_1^C, -g_2^C$	$+g_2^C, -g_1^C$
Motor					$\Rightarrow s$	
Sensor						✓
Axon						$g_1^S \Rightarrow g_1^M, g_2^S \Rightarrow g_2^M$
Reactors:						
ConstReactor	$\mathfrak{B}_0$	$\mathfrak{C}$	$\mathfrak{B}_M$	$\mathfrak{B}_S$	$\mathfrak{M}$	$\mathfrak{S}$
SourceReactor	b_0	c	b_M	b_S	m	s
AxisReactor		$g_1^C, g_2^C, s_1^C, s_2^C$			$g_1^M, g_2^M, s_1^M, s_2^M$	
InterAxisReactor						$g_1^S, g_2^S, g_2^M, g_1^M, s_1^S, s_2^S$

Supplementary Material

The Self-Construction and -Repair of a Foraging Organism by Explicitly Specified Development from a Single Cell

Fabian Roth, Hava Siegelmann, Rodney J. Douglas

Parameter values

Thresholds

In our simulation we used the following thresholds for the values in Table 1:

η	1.0
$\hat{b}_0$	0.8
$\hat{c}$	10.5
$\widehat{b_M}$	0.3

$\hat{b}_S$	0.3
$\hat{m}$	3.5
$\hat{s}$	3.5
θ	1.8

Diffusion coefficients

Morphogen	Membrane	Environment
$\mathfrak{B}_0$	0.0	0.0
$\mathfrak{C}$	0.0	0.0
$\mathfrak{B}_M$	0.0	0.0
$\mathfrak{B}_S$	0.0	0.0
$\mathfrak{M}$	0.0	0.0
$\mathfrak{S}$	0.0	0.0
b_0	0.8	0.7
c	0.8	0.7
b_M	0.8	0.7
b_S	0.8	0.7
m	0.8	0.7
s	0.8	0.7

Morphogen	Membrane	Environment
g_1^C	0.025	0.10
g_2^C	0.025	0.10
s_1^C	0.600	0.85
s_2^C	0.600	0.85
g_1^M	0.030	0.08
g_2^M	0.030	0.08
s_1^M	0.550	0.65
s_2^M	0.550	0.65
g_1^S	0.030	0.08
g_2^S	0.030	0.08
s_1^S	0.550	0.65
s_2^S	0.550	0.65

The diffusion coefficients for all morphogens used in the simulation. The Membrane column specifies the diffusion coefficients of the morphogen through the cell membrane into the local environment. The Environment column specifies the diffusion coefficient on the lattice of the local environment.

AxisReactor cell type *C***:** $c = 0.13$, $\gamma = 0.05$, $a = 1$, $\alpha = 0.02$

AxisReactor cell type *M***:** $c = 0.13$, $\gamma = 0.1$, $a = 1$, $\alpha = 0.02$

InterAxisReactor cell type *S***:** $c = 0.13$, $\gamma = 0.1$, $a = 1$, $\alpha = 0.02$, $\delta = 0.6$

SourceReactor The rates r for the SourceReactors are the same for all cell types: $r = 0.2$

ConstReactor The gain α for the ConstReactors are the same for all cell types: $\alpha = 0.1$

MigrateCompetence See section MigrateCompetence for a description of the parameters: $\epsilon = 4$, $\lambda = 50$.

The part of the cell's free energy function depending on the chemical configuration is determined by the following formula:

$$J_p(\mathbf{c_r}) = 30 \cdot \langle \mathbf{c_r}, \mathbf{c_a} \rangle,$$

where $\mathbf{c_r}$ is the the chemical configuration at node $\mathbf{r}$, $\langle \cdot, \cdot \rangle$ is the inner product and $\mathbf{c_a}$ is a vector containing the affinities to the specific chemicals, $\mathbf{c_a}$ contains +1 and −1 depending on the cells preference to be attracted or repulsed by the corresponding chemical. This is specified in the Description code.

The temperature for the MigrateCompetences is 5. The temperature for the AxonCompetence is 0.1.