

On the State of the Art of POEtic Machines¹

Christof Teuscher

*Logic Systems Laboratory, Swiss Federal Institute of Technology Lausanne,
EPFL-DI-LSL, CH-1015 Lausanne
Tel.: +41 21 693 67 14, Fax: +41 21 693 37 05*

Abstract

Biological systems grow, live, adapt, and reproduce, characteristics that are not truly encompassed by any existing computing system. If one considers life on Earth since its very beginning, three levels of organization can be distinguished: *phylogeny*, *ontogeny*, and *epigenesis*. Machines that combine evolutionary mechanisms (Phylogeny), developmental processes (Ontogeny), and learning algorithms (Epigenesis) are called *POEtic* machines. The goal of this paper is to provide an overview on actual bio-inspired machines with a special glance on so called POEtic machines, systems that grow, evolve, and learn. A second objective is to analyze state-of-the-art models and to delineate possible ways for future work.

Key words: evolution, development, learning, bio-inspired computing machines, hardware

PACS:

1 Introduction

Biological systems grow, live, adapt, and reproduce, characteristics that are not truly encompassed by any existing computing system. The concept of “living” has a number of consequences in terms of adaptation, interaction with the environment, and the ability to deal with limited resources. It is believed that methodologies and technologies that enable the construction of complex artificial systems that live, grow, and adapt in hardware would allow a quantum leap in performance for many computing systems known so far.

Email address: christof@teuscher.ch (Christof Teuscher).

URL: <http://lslwww.epfl.ch> (Christof Teuscher).

¹ This work was supported in part by the Swiss National Science Foundation under grant 21-54113.89 and by the Leenaards Foundation, Lausanne, Switzerland.

If one considers life on Earth since its very beginning, three levels of organization can be distinguished [44, 55, 62]: *phylogeny*, *ontogeny*, and *epigenesis*. Phylogeny concerns the temporal evolution of the genetic program. The phylogenetic mechanisms are fundamentally nondeterministic, with the mutation and recombination rate providing a major source of diversity. This diversity is indispensable for the survival of living species, for their continuous adaptation to a changing environment, and for the appearance of new species. Ontogeny is the developmental process of a multicellular organism. This process is essentially deterministic. Epigenetic processes can be loosely grouped under the heading of *learning* systems and primarily include the nervous system, the immune system, and the endocrine system. These systems are characterized by a basic structure that is entirely defined by the genome (the *innate* part), which is then subjected to modification through the lifelong interactions of the individual with the environment (the *acquired* part).

Machines that combine evolutionary mechanisms (phylogeny), developmental processes (ontogeny), and learning algorithms (epigenesis) are called *POEtic* machines. While each of these models, taken separately, has to a greater or lesser extent been used as a source of inspiration for the development of computing machines, their amalgamation into hardware is a challenge yet to be met and will give birth to novel bio-inspired systems that go beyond traditional computing devices. Alan Kay, a visionary computer scientist and one of the founders of the Xerox Palo Alto Research Center, hit the spot with the following statement: “The computer ‘revolution’ hasn’t happened yet!”². Kay also postulated that the ideal computer would function like a living organism. Each cell would behave in accord with others to accomplish an end goal but would also be able to function autonomously. Cells could also regroup themselves in order to attack another problem or handle another function. So far, this vision remains a dream!

The 21st century promises to be the century of bio- and nano-technology. Exciting new technologies such as self-assembling systems, organic electronics, living intelligent machines (artefacts), hybrid electronical-biological machines, etc., and the ever-increasing complexity of systems will require new design and engineering methods. Machines will probably no longer be designed by man in every detail but evolutionary and adaptive methods will help in building more intelligent and more complex systems. The increasing overall complexity and the increasing number of elements (e.g., transistors, gates, etc.) in a system will force designers—and Mother Nature does this job already very well—to build perfect systems out of imperfect components.

The goal of this paper is to provide an overview on actual bio-inspired ma-

² Talk given at Educom’98,
<http://www.educause.edu/conference/e98/webcast98.html>

chines with a special glance on so called POEtic machines, systems that grow, evolve, and learn. A second objective is to analyze state-of-the-art models and to delineate possible ways for future work. The number of papers in developmental modeling, neural networks, and evolutionary algorithms is huge and there is definitely not enough space to describe in detail all existing models. However, with the bibliography provided, the interested reader should easily be able to probe further.

The remainder of the paper is as follows: Section 2 will provide a short overview on the inspiration engineers might get from nature. In Section 3, the three principal forms of adaptation in nature are described separately: (1) evolutionary algorithms in Section 3.1, (2) developmental models in Section 3.2, and artificial neural networks in Section 3.3. Important and promising POEtic models—models that try to put together evolution, development, and learning—are presented in Section 4. This section does by far not mention all, but only the most promising and interesting approaches. Finally, the conclusion comments existing models and possible future work.

2 Lessons from Mother Nature

“There is such a long and colourful history of engineers, scientists and artificers gaining inspiration from nature that one could be forgiven for thinking that all the best ideas have been spoken for” [3]. Nevertheless, we still have a poor understanding of many natural mechanisms, above all at the molecular scale. Many models have been built to understand biological systems better, but models never work as well as biology. Let us for example take a closer look at the cell, an extremely versatile and adaptive biological assembly: “The simplest living cell is so complex that supercomputer models may never simulate its behavior perfectly.” Thus the beginning of a recent article in the *Scientific American* [27]. The article emphasizes that most attempts to create artificial life or to faithfully model biological systems suffered from a tremendous number of degrees of freedom. The system’s parameters could then be tweaked to produce almost any desired behavior. Further, models are often so complicated that they have absolutely no ability to predict anything. However, often the goal is not to obtain a perfect prediction but rather a reliable approximation. Masaru Tomita [68] believes that the study of the cell will never be complete unless its dynamic behavior is understood. He suggests that the complex behavior can only be understood by means of computer models that allow to undertake complete simulations. In a recent *Nature* article Drew Endy and Roger Brent [20] emphasize that “[p]ast efforts to model behaviour of molecular and cellular systems over absolute time typically were qualitatively incomplete or oversimplified compared to available knowledge, and qualitatively incomplete in the sense that key numbers were unknown.”

Finally, it is estimated that complete *E. coli* simulations could run, perhaps, on a single processor system by 2020.

The failure of most approaches trying to faithfully model biological systems is one of the reasons why a lot of research not seeks to duplicate or copy nature but only tries to mimic it and to get some inspiration from. This is also known as *biomimetics* and *bio-inspiration*. Typical research topics are artificial life (Alife), artificial intelligence (AI), artificial neural networks (ANNs), evolutionary algorithms (EAs), etc. But many bio-inspired approaches have been labeled as failures too for not having lived up to grandiose promises. “At the heart of this disappointment lies the fact that neither AI nor Alife has produced artefacts that could be confused with a living organism for more than an instant” [5], says Rodney Brooks. Something must be wrong! But what? He proposes four possibilities:

- (1) we might just be getting a few parameters wrong;
- (2) we might building models that are below some complexity threshold;
- (3) perhaps it is still a lack of computing power; and
- (4) we might be missing something fundamental and currently unimagined in our models of biology.

I will come back to these statements in the conclusion of this paper where the presented models shall be analyzed with regard them.

Finally, for the sake of completeness, it should all the same be said that Mother Nature not always offers the best ideas. Philip Ball states in a recent Nature article: “To appreciate that nature does not necessarily have all the best ideas, we only need point to the wheel. [...] Nature has good reasons to avoid metallic components, for example, but this does not mean that human engineers strive to do so” [3]. In his book entitled “Cat’s Paws and Catapults”, Vogel [72] also presents elegant and persuasive arguments why it would be foolish to assume that nature has all the best ideas. But this should of course not discourage engineers to continue mimicking nature’s promising and well working inventions.

3 Single and Simple: Adaptation Forms Considered Separately

In the early days of computer science already, people tried to combine evolution, learning, and development. Few people know, for example, that Alan Turing proposed in 1948 artificial neural networks based on very simple elements. He suggested to organize a randomly built network by means of some sort of “genetical search” [65]. Today, evolutionary neural networks—the combination of evolutionary principles with artificial neural networks—are well

established and often used to cope with hard problems in machine learning. The next three sections shall provide a brief overview on single-mechanism models, namely evolution, development, and learning.

3.1 Evolutionary Algorithms

Evolutionary algorithms (EA) are a collection of methodologies inspired by the principles of the biological evolution. The basic concepts go back to the work of Charles Darwin [9]. Later, John Holland first introduced and substantiated the idea of *genetic algorithms* (GAs) [32]. To date, many variations and extensions of algorithms and methods inspired by the biological evolution have been proposed. For a review of the current state of the art, the reader is referred to [24]. One of the most recent milestones was the introduction by John Koza [40] of a method called *genetic programming* that deals with the automatic generation of computer code.

The term evolutionary algorithm usually encompasses a number of related methodologies such as *genetic algorithms*, *evolutionary strategies*, *evolutionary programming*, *genetic programming*, etc. Mathematically speaking, evolutionary algorithms are a broad collection of optimization methods that are particularly suitable for “hard” problems where little is known about the underlying search space. In order to optimize a solution, an evolution process is simulated, in the course of which the parameters that produce an optimal solution are determined. As biological evolution does, evolutionary algorithms maintain a population of *individuals*, each one representing a possible solution for a given optimization problem. Each individual is represented by a finite string of symbols, called *genome*. The *search space*—generally too huge to be exhaustively searched—contains all possible solutions to the problem. The algorithm starts with the creation of an initial population of individuals. Usually, this population is randomly generated, however, some heuristics are sometimes applied in order to reduce the search space at the beginning. All individuals are then evaluated according to a certain *fitness function*. The next step is *selection*—according to fitness, and one of many known selection strategies—of the individuals of generation $g(t - 1)$ to form a new population. In order to explore the search space, new individuals, i.e., new solutions, are created by means of the *crossover* and *mutation* operators. Crossover simply exchanges—according to some strategies—parts of the parent’s genomes to create a child, also called *offspring*. The goal of this operator is to create individuals that move towards the optimal solution. Mutation, on the other hand, randomly changes some symbols on the genome with a small probability. The goal of this operator is to randomly explore the search space. Compared to local optimization methods, i.e., gradient descent, genetic algorithms have the advantage that they less often get trapped in local minima of the function

to be optimized. Since a population of solutions is used, the algorithm can “move away” from local optima if the population finds better solutions in other areas. The disadvantage of genetic algorithms is that there is no guarantee that the process converges to the optimal solution and that they are often computationally very intensive.

Evolutionary hardware is closely related to evolutionary algorithms and refers to hardware that can change its architecture and behavior dynamically and autonomously by interacting with its environment [77]. So far, most evolutionary hardware used evolutionary algorithms as their main adaptive mechanism. It may, however, also be combined with artificial neural networks. Evolutionary hardware is not just a hardware implementation of evolutionary algorithms but goes far beyond. The interested reader is referred to [55, 59–61].

3.2 *Developmental Models*

Biological development is a highly complex, hierarchical program that operates across many different scales and at each level of scale many different mechanisms and self-organizing processes are being involved. The dynamic interactions between the different mechanisms and elements result in a highly complex system. Development in artificial systems is a very important issue and it is commonly believed that it is a key to the generation of highly complex systems (see for example Kitano [36]).

Much of the early work in developmental modeling focused on modeling a particular mechanism only. The mechanism has then be considered in isolation and not in interaction with other mechanism, although it has long been evident that the interactions of different mechanisms are critical to several issues, like for example pattern formation. Today, there is a growing trend towards developmental models based on multiple mechanisms (like cell migration, cell division, attraction forces, environmental influences, etc.). In the following, some of the most important models and work shall be presented.

Besides D’Arcy Thompson’s classical work “On Growth and Form” [67], first published in 1917, Alan Turing was one of the first persons interested in modeling morphogenesis. In his 1952 seminal paper entitled “The Chemical Basis of Morphogenesis” [56,69] he proposed a mathematical theory of cell-cell interaction via chemical substances (also called *reaction-diffusion model*). Turing described several sets of differential equations that governed the interactions between different substances that diffuse and react with each other. The original intent of the reaction-diffusion model was to explain the “breakdown of symmetry and homogeneity” or the emergence of a pattern in an originally homogeneous medium.

In 1968, Aristid Lindenmayer developed a grammar-based technique called *L*-systems [42] (further developed later by Prusinkiewicz [52]). *L*-systems is a rule rewriting formalism that allows to efficiently describe growth, e.g., plant growth. For a particular *L*-system, the growth always starts from the same seed cell, called *axiom*. *Production rules* are used to describe the growth of new cells from the old cells. *L*-systems are often used to model development in combination with evolutionary algorithms and neural networks, are computationally feasible models, but not well suited for models based on local interactions only. Kitano, for example, uses a graph-generation grammar based on *L*-systems where the structure of the network is not directly encoded on the genome [35]. His approach shows better scaling properties, is more biologically plausible, and generates more complex networks, but it is not always easy to determine whether a grammar-based encoding can express every possible network architecture. Another important grammar-based approach comes from Gruau [28–30]. He developed a special encoding scheme called *cellular encoding*. Instead of rewriting characters, the rewriting grammar directly rewrites neurons. A genetic algorithm can then be used to find a grammar tree suitable for a given problem. In order to speed up the search for functional networks, Gruau also included several forms of learning in his model (see for example [30]).

Odell et al. [48] studied in 1981 the mechanical aspects of morphogenesis. They presented a model for the folding and movement of cell sheets and applied it to certain elementary types of gastrulation and neurulation. Work with mechanical models has been continued later with more detailed and more complex models. Cells have different shapes and various forces are applied. Oster et al. [48] and Weliky and Oster [73], for example, use cells that have a polygonal shape in three dimensions. Cells are subjected to forces that are due to osmotic pressure and to the elastic membrane. To probe further: Fleischer [22] provides an overview on developmental modeling in his Ph.D. thesis.

Membrane computing (also called *P*-systems) is a new and emerging branch of computing initiated by Paun [49]. A *P*-system, which is related to various areas such as *L*-systems, formal language theory, and multiset processing, is a computing model that highly abstracts from the way the alive cells process chemical compounds in their compartmental structure. Multisets of objects are placed in compartments defined by the membrane structure and the objects evolve by means of *reaction rules* associated with each compartment. Objects are described by symbols or by strings of symbol and can pass through membranes that can change their permeability. Membranes can also be created, dissolved and divided. By using the reaction rules in a nondeterministic, maximally parallel manner, one gets transitions between the system configurations. Many *P*-systems are computationally universal, i.e. equal in power to Turing machines. Membrane computing has principally been motivated by mathematics and not because it is a plausible biological model. Nevertheless,

it is a powerful and efficient approach that can quite realistically model many biochemical reaction of biological cells, developmental processes, cell replication, cell division, etc.. To the best of our knowledge, *P*-systems have never been implemented in hardware.

More recently, the *Embryonics project* (embryonic electronics) [43,44], an undergoing project in our lab, implements quasi-biological development in real hardware. The main goal is the design of highly-robust integrated circuits, endowed with the properties usually associated with the living world: *self-repair* and *self-replication*. Recently, the BioWatch, a large-scale implementation of a fault-tolerant and self-repairable watch has been presented [63]. To the best of our knowledge, the Embryonics project is the only approach that implements quasi-biological growth processes in real hardware. A weak point of the Embryonics project is that there exists no methodology and no tools allowing to construct automatically artificial organisms. Thus, all organisms are coded by hand so far. Furthermore, the model is not flexible enough to allow for a direct implementation of learning and evolutionary mechanisms.

3.3 Artificial Neural Networks

Since the beginning of neural network research more than fifty years ago, the gap between the goals of neuroscientists, mathematical modelers, and computer scientists increased. Neuroscientists are principally interested in the search for a complete understanding of biological information processing systems whereas computer scientists are interested in solving complex engineering problems using sophisticated artificial neural networks. These models are often “neural” only in the sense that they have been inspired by neuroscience but not at all because they are faithful models of biologic neural and cognitive phenomena.

An artificial neural network is an information processing system which is made up of a number of simple, highly interconnected processing elements—the *neurons*—which process information in parallel. As a simplification, the neuron might be considered as a sort of *detector* that detects the existence of some set of conditions and that responds with a signal that communicates the extent to which those conditions have been met. Artificial neural networks can be considered as an alternative approach to the problem of computation just as biological neural networks are one of many possible solutions to the problem of processing information.

Few people know that Alan Turing pioneered artificial neural network research in 1948 already in describing different computing machines built out of very simple, neuron-like elements connected together into networks in a largely ran-

dom manner [65]. However, the birth date and opening shot in neural network research was the 1943 paper by McCulloch and Pitts [45]. With regard to their work, there is however no doubt, that Turing’s work on neural networks goes importantly beyond their earlier work. It was Wesley Clark and Belmont Farley and not Hebb who first simulated in 1954 an artificial neural network [8, 21]. In 1956, Rosenblatt unveiled his neuron—the *perceptron*—that was principally based on Hebb’s ideas [31]. Hebb suggested that a mass of neurons could learn if their connection-strengths change according to some rule—today known as the *Hebbian rule*. Today, a wide variety of different neural models and learning algorithms exist and artificial neural networks are a broadly accepted and viable computational model for many problems.

One of the disadvantages of artificial neural networks is that they are still designed most often by hand and thus reflect important theoretical claims, experience, and knowledge on the part of the modeler. There are very little general rules on how to design connectionist models. The network’s architecture and topology is often the most important part and even the best learning algorithm is doomed to failure if the underlying network topology is not suitable for a given problem. Another problem is that the number of free parameters of most network models exponentially grows with the network size. Thus, even the ever increasing computational power (underlying Moore’s law) won’t be able to overcome this obstacle and we definitely need new approaches for building and simulating huge artificial neural networks.

4 Together we are Stronger: Towards POETic Machines

Development, evolution, and learning are adaptation forms that allow individuals and species to adapt to a changing environment on different time scales. Sometimes, this adaptability is even considered as “intelligence” *per se*. In recent years, there has been a great interest in combining learning and evolution with artificial neural networks (see for example [76]). Evolutionary artificial neural networks (EANN) make their adaptation much more efficient and effective to changes in the environment. Evolution can act at different levels, for example: the connection weights, the architecture, the learning rules, etc. Thereby, network encoding scheme is a key issue. There are basically two main encoding schemes: (1) direct encoding and (2) indirect encoding. The direct encoding scheme encodes all the details of the network whereas the indirect encoding schemes only encodes the most important parameters. One potential problem of the direct encoding scheme is the scalability. On the other hand, the indirect encoding scheme often finds more compact genotypical representations, may not always find compact phenotypical representations, but seems biologically more plausible.

In nature, the development of a nervous system can roughly be separated into three phases: (1) first, nerve cells are generated through cell division; (2) the cells grow out axons and dendrites along specific routes in order to form a provisional interconnectivity; (3) last, the synaptic interconnections are refined and remodeled according to the pattern of electrical activity in the neural network. The final phase normally continues into an individual's adult life. Most actual neural network and evolutionary neural network approaches don't model the three above mentioned phases together but are restricted to the one or two phases only.

Evolutionary systems and hardware (see also Section 3.1) are often classified in two categories: (1) *intrinsic* and *extrinsic* hardware. Extrinsic hardware simulates evolution by software (*offline evolution*) and only downloads the best configuration to hardware in each generation. On the other hand, intrinsic hardware simulates evolution (*online evolution*) directly in hardware, i.e., each genome is used to (re-)configure the hardware. Since a single individual in nature does not evolve itself, extrinsic evolution seems more biologically plausible. Therefore, most often, the phylogenetic axis is sufficiently well represented by encoding the entire system on a genome. It is then rather easy to apply offline evolution (extrinsic hardware).

The following sections shall give a short overview on some of the most promising and most important models trying to put together evolutionary principles, growth processes, and artificial neural networks. For a short introduction to development, learning, and evolution in animates, the interested reader is also referred to the paper of Kodjabachian and Meyer [37].

4.1 A Hexagon Based Tissue

The diploma thesis of Jens Astor [1] and further work in collaboration with Chris Adami [2] presented a model of decentralized growth and development of artificial neural networks inspired by developmental biology and the physiology of nervous systems. The model is particularly interesting since each neuron is completely autonomous and its behavior is determined only by the genetic information contained within the cell and by the local concentration of substrates. An artificial chemistry is used to model the chemicals and substrates. The model implemented follows the four basic principals of molecular and evolutionary biology [2, p. 190]:

- *Coding.* The model should encode networks in such a way that evolutionary principles can be applied.
- *Development.* The model should be capable of growing a network by a completely decentralized growth process, based exclusively on the cell and its

interactions.

- *Locality*. Each neuron must act autonomously and be determined only by its genetic code and the state of its local environment.
- *Heterogeneity*. The model must have the capability to describe different, heterogeneous neurons in the same network.

Astor and Adami state that “[o]ne of the key features of a model implementing the above principles will be the absence of explicit activation functions, learning rules, or connection structures. Rather, such characteristics should emerge in the adaptive process.”

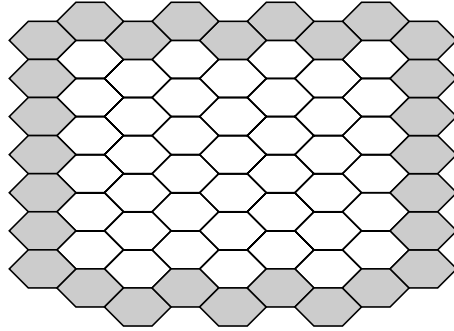


Fig. 1. Tissue made up of hexagon units with boundary elements that absorb substrates.

The model they used is two-dimensional surface (“tissue”) made up of hexagon unit (see Figure 1). Hexagon units have been chosen in order to always have the same distance to each neighboring unit. Each hexagon harbors certain chemical concentrations of substrates. The tissue is surrounded by a special boundary that absorbs the substrates and thus models the diffusion in infinite space. Four different classes of substrates are distinguished: *external*, *internal*, *cell-type proteins*, and *neurotransmitters* (for more details see [1,2]). Furthermore, the tissue can harbor different neural cell types (*actuator cell*, *sensor cell*, and common neurons) and connections. Common neurons represent hidden neurons whereas actuator and sensor cells represent the input and output layer of the network. There are no special connection cells. A hexagon unit that contains a neuron also contains dendritic and axonic connections to other cells. The growth of axons and dendrites is guided to their target by means of diffusible growth factors. Each neural cell contains a genome that encodes its behavior. An important aspect of the model is that genes are regulated and regulate each other. A neuron’s function is mainly determined by its genetic information and the local concentrations of substrates. In order to allow for learning, the model also assigns for each dendritic connection a weight for each type of neurotransmitter. The weight can be modified by different expression commands. A simulation starts by creating a number of sensor and actuator cells that depend on the complexity of the problem. Then, a single initial neuron is placed in the middle of the grid. The neuron then starts growing axons and dendrites, produces offspring cells, and might initiate cell differentiation.

Astor and Adami’s model has successfully been applied to some very simple problems and they agree on the fact that “[...] the system has to be evaluated in more depth” [2] and that experiments on a large scale have to be done. As with many other artificial life model, the number of free parameters is high, the genotypes are large, and the search space thus huge. With the current computational power, only very simple problems can be considered³. It has been shown that the model is able to give rise to the development of artificial neural networks based on local interactions only. The existing model could be improved in considering cell death as well as morphology. Cells are currently always located at the same place from their birth on and cannot move along a chemical gradient. In order to make to model more biologically plausible, one might also consider including cell membranes in the model.

4.2 A Multiple-Developmental Model

The importance of dynamic morphology in neural-like systems has been demonstrated in the Ph.D. thesis of Kurt Fleischer [22] in 1995 (an early version of the work was published in 1994 [23]). Fleischer introduces a model of multicellular development that combines elements of the chemical, cell lineage, and mechanical models of morphogenesis pioneered by Turing [56, 69], Lindenmayer [42], and Odell [48]. Cell migration is an important aspect of development, especially of the neural development. His model—and this distinguishes it from previous work in reaction-diffusion systems and grammar based systems—allows cells to move freely within the environment. The internal state of each cell in the model is represented by a time-varying state vector that is updated by differential equations. The driving question behind Fleischer’s work was the following [22]: “What features of a representation scheme and a developmental process are important for describing a wide range of structures, and also provide the robustness and developmental properties?”

He was able to show by means of different simulation experiments that his model represents a wide range of biologically relevant phenomena in two and three dimensions. Among other applications, he also applied his model to the evolution of artificial neural networks using a developmental model, however, he too failed to show revolutionary results as there are too many parameters in his model. His largest developmental model—made up from 70 cells—was able to follow gradients of diffusion chemicals. Fleischer also applied his model to synthetic biology with the main goal to study multicellular morphogenesis and pattern formation.

³ The authors propose, however, an asynchronous and distributed system that allows to search for good genomes in parallel on the internet: <http://norgev.alife.org>.

Let's take a closer look into Fleischer's model. He identified the following features of biological development as being critical to morphogenesis and pattern formation [22]:

- *Boundary*: A cell has a boundary surface with some shape and location.
- *Chemicals*: Chemicals exist within the cell, on the cell surface, and in the extracellular space.
- *Sensors*: A cell can sense local information from the extracellular environment, and the proportion of its surface chemicals which are bound to complementary chemicals.
- *Cell lineage*: A cell can control the orientation of its next cleavage, as well as the initial states and cell types of its immediate children.
- *Equations of motion and shape change*: A cell is capable of exerting forces to move and change shape, which are mediated by viscous dynamics, collisions with other cells and obstacles, and adhesion due to binding surface chemicals.
- *Death*: A cell can cause its own death.

In order to overcome difficulties in modeling the system, Fleischer has chosen to model cell shapes with circles or spheres instead of an arbitrary shape. Furthermore, each intercellular chemical is represented by a single value and each surface chemical is assumed to be uniformly distributed on the boundary of the cell. From these features and abstractions, the model has been built. Each cell has an array of state variables that represent its intercellular and surface chemicals. State changes are effected using differential equations, the *cell state equations*. Input from the cell's sensors are parameters to the cell state equations. The cell's behavior is determined by the *cell behavior functions*. Cells are only able to access local information and do not know their absolute position and orientation in the world. Their behavior is only affected by their environment. Fleischer has also implemented a simple abstraction of neural growth using growth cones which are modeled as small cells with few restrictions. The growth cones are connected to the parent cell by a neurite and growth cones and cells communicate via a set of state variables which are held in the neurite. It is also possible to simulate spiking neurons.

Fleischer's approach revealed also several limitations. For example, the abstraction of the spherical cell shape is rather unrealistic and a more detailed cell shape model is desirable. Nevertheless, the thesis presents a multiple-developmental model that can represent a variety of biological phenomena. "The next step is to apply the model to a particular biological system" [22].

4.3 *Agents are going Cellular*

Another biologically defensible model of development that also includes evolutionary techniques and neural network-like structures has been proposed by Frank Dellaert and Randall Beer [15–17]. They were mainly interested in the synthesis of autonomous agents—including bodies and control systems—without using a direct phenotype to genotype mapping since this is more biologically plausible. The first principal component of Dellaert and Beer’s approach is that gene expression is regulated by a *genetic regulatory network*. The cell state is represented as a binary vector which is updated via a boolean network à la Kauffman [34]. The second component of their work consists of a simple two-dimensional cellular simulator that models the cellular level. Each cell is represented by a square element that can divide in any of the two directions (vertical or horizontal). The last aspect taken into account in their model concerns the intracellular communication which is extremely important in biological development. Thus, the future state of a cell will not only depend on its own state but also on the state of the surrounding cells.

A simulated cell cycle consists of two phases: (1) interphase and (2) mitosis. During the interphase, the cell state is synchronously updated until a steady state is reached. During mitosis, it either stays intact and waits for the next interphase or the cell divides into two daughter cells whereas its state vector is inherited by the two daughter cells. The daughter cells then differentiate into distinct cell types (i.e., sensor, actuator, or control-neuron cells). No cell movement has been implemented in the model. The organism itself is a two-dimensional square of cells. Development starts from a single cell (square), the zygote, then subsequently divides according to the state of the genetic regulatory network (see Figure 2).

Dellaert and Beer have successfully shown that their model is evolvable and that it can be optimized to perform some task in the fully developed organism (e.g., an obstacle avoidance task [17]). However, it has been impossible to evolve complex organism from scratch. It should also be noted that in general, no learning takes place in their model (they did, however, without giving any details, some experimentation in [17]).

4.4 *An Artificial Retina*

In 1998, Alistair G. Rust investigated in his Ph.D. thesis [54] models inspired by the biological neural development. The model he chose as appropriate develops three dimensional neuron-to-neuron interconnections. Rust’s ultimate goal is having an artificial neural network model which could automatically

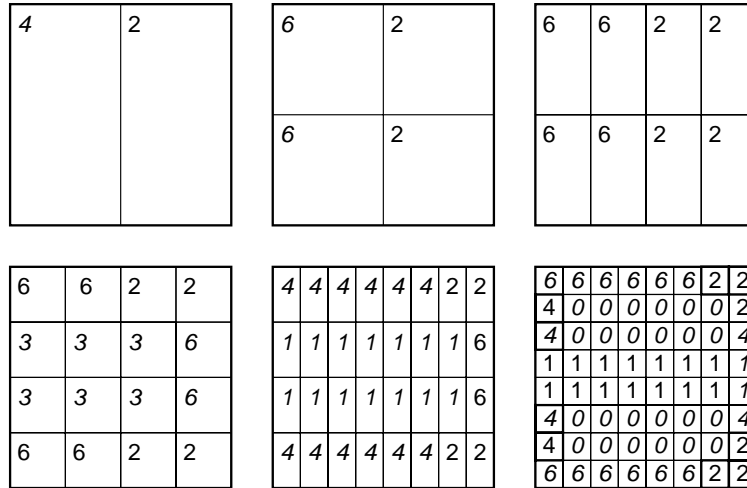


Fig. 2. The six consecutive stages of development in one of Dellaerts organisms. Digits denote cell types, italics indicate that in the corresponding cell the color/cell type changed relative to the previous developmental stage. Source: [15, p. 91].

bootstrap to a given application in developing networks of arbitrary size, connectivity, and functionality. As a testbed application, the development of an artificial edge-detecting retina has been chosen. An artificial retina model presents a number of advantages. The network is considered to be hardwired and no learning and feedback mechanisms are required. The structure is modular and arranged in layers. For edge-detection, only three types of neurons are used: (1) cones, (2) bipolars, (3) and horizontal neurons. And last but not least, the retina has been chosen since it is a widely studied model and thus provides exemplar solutions.

The neurons used are stationary in the environment and are able to grow out axons and dendrites. The growth process is guided by a set of abstracted developmental rules that each neuron contains. Neurons, axons, and dendrites emit local chemical gradients of artificial neurotrophin. Rust states that “[t]he design of the outgrowth rules is motivated by Stryker’s assertion that the development of precise structure of the brain can be governed by imprecise rule [64].” Parameters allow the developmental phase of the network to be flexible and adaptive. A genetic algorithm has been used to search through the developmental parameter and to evolve an artificial retina.

In summary, the networks built do not bear strong resemblance to classical artificial neural architectures. The model, however, incorporates interactive self-organization during the developmental phase and thus creates complex three-dimensional neural structures using relatively few and simple rules. In addition, a novel activity-based pruning mechanism has been realized.

Rust has shown that a simple artificial edge-detecting retina can be evolved using genetic algorithms to identify sets of optimal parameters. The network’s

connectivity self-organized by means of interactive neurite branching, self-regulated outgrowth rates, and pruning methods.

4.5 On Growing Intelligence

For Jari Vaario and Setsuo Ohsuga (for more details see also [70]), intelligence is the capability of adaptation to a given environment. They identify the following forms of adaptation [71]:

- development (=ontogeny);
- learning (=epigenesis);
- natural selection and genetic changes (=phylogeny).

In their “On Growing Intelligence” paper, the above mentioned forms of adaptation are simulated in the context of autonomous systems. Designing an autonomous system for a dynamical environment by hand is a difficult task since it is almost impossible to foresee all possible situations and behaviors. Nature’s answer to this problem is adaptation. Vaario and Ohsuga illustrate the life cycle of a nervous system as shown in Figure 3.

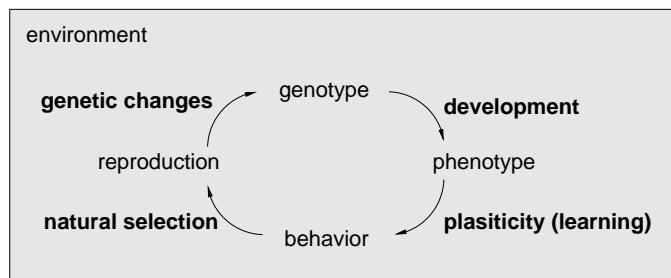


Fig. 3. The life cycle of a nervous system according to Vaario and Ohsuga illustrating the different forms of adaptation: development, learning, natural selection, and genetic changes.

Vaario and Ohsuga’s overall model consists of three interleaved models: (1) neurons, (2) organisms, and the (3) environment. The modeling method used is based on the production rules resembling L-systems [42]. *Interleaved L-systems* are L-systems that may have other L-systems inside it and that are able to modify these systems as well. “For example, the L-system of the environment has several organisms which are defined by L-systems” [71]. The developmental process (the L-system) grows the neural structure, whereas the evolutionary process gradually generates more complex systems. Vaario and Ohsuga successfully applied his method to the design of Braitenberg vehicles [4]. The biggest restriction of the model and its implementation was that large scale simulations with many individuals were almost impossible due to the computational resources required. Another major problem was the system’s bootstrapping: it was extremely difficult to find initial production rules

that allowed for further and more complex development. Thus, they explicitly designed some simple systems first and started evolving them to more complex ones. However, this still was no guarantee that more complex systems evolved. In the future, they want to include neural plasticity based learning for the organisms.

4.6 Cells Divide, Differentiate, and Die

Another model that uses artificial genetic regulatory networks to control the development of artificial neural networks has been proposed in 1997 by Peter Eggenberger [19]. The model is particularly interesting because it evolves three dimensional neural networks by means of strictly local interactions between the cells and the genes only. Each cell basically contains the same genome and cell differentiation is achieved due to different signals that each cell gets from the environment, i.e., from other cells. The genetic regulatory network controls cell division, cell differentiation, and cell death. The model also includes cell migration. Whereas the number of genes in the genome of most models proposed so far grew with the number of neurons, the genome in Eggenberger's model does not necessarily grow with a growing number of neurons. This might be crucial for an efficient implementation of large networks. For gene regulation and expression and for cell differentiation, a sort of artificial chemistry has been implemented. As the cells become different, they will produce different substances. Two different classes of cell adhesion molecules are used to connect two neurons together. The connection further contains a weight that can be changed by a Hebbian rule and synapses are either inhibitory or excitatory.

Although the model seems not having been applied to real-world applications, it is a promising approach. The construction of neural networks was possible with no explicit encoding of the network structure, the cell types, the cell's position, and interconnections. In the future, Eggenberger wants to automatically build re-entry maps between different neural network layers in order to obtain networks that can automatically adapt to different tasks.

4.7 Phenotypic Plasticity

In [47], Nolfi and Parisi present simulations of the evolution of populations of neural networks. They principally had two objectives:

- (1) to propose a more realistic genetic coding of neural network architectures that better approximates biological facts, and

- (2) to study the genotype-to-phenotype mapping as a process with a temporal notation.

So far, most work that applied genetic algorithms to the construction of artificial neural networks uses an instantaneous genotype-to-phenotype mapping, i.e., organisms are born with a fully constructed (“mature”) nervous system. The network can only be changed by applying some form of learning. Nolfi and Parisi note that “[i]n real organisms, however, the mapping from genotype to phenotype is not instantaneous but takes time extending from conception well into extra-uterine life.” Nolfi and Parisi’s method allows to establish connections during the growth process. The connection between two neurons is established when a growing axonal branch reaches the neuron. For testing their animats, they used a two-dimensional environment divided into square cells. Each animat is equipped with sensors that allow to detect the food elements that are randomly distributed in the environment. At any particular moment, an animat occupies one single cell and it can only move one single cell at each moment. When the animat senses a food pellet, it will eat it and the food disappears. The simulation results have shown that evolution chooses very simple network architectures in terms of the number of units and connections used. The architectures also tend to be structured in functional sub-nets. Furthermore, they think that by integrating a growth process during the life of an individual organism, “[...] the way is open to study genetically controlled neural and behavioral development as an additional dimension of change besides evolution and learning.”

In an extension of their work, Nolfi et al. [46] allowed a neuron to grow its branching axon only if the neuron’s activation variability exceeds a genetically specified threshold. Thus, the environment influences the neural development, in other words, the genotype is environmentally sensitive and develops into a different phenotype as a function of the environment. This is also called *phenotypic plasticity* (for more details see [57]). The extension of their work might be considered as a simulation of growth factors as they exist in nature.

4.8 *Artilect or Fartilect?*

Visionary brain builder Hugo de Garis has two goals in his life⁴: (1) to build artificial brains, and (2) to raise the alarm on a possible gigadeath artilect war. His interest for genetic programming, artificial neural networks, and artificial embryology goes back up to his Ph.D. thesis in 1992 [10]. Since his thesis, de Garis concentrated on the so called CAM-brain machine (CAM stands for Cellular Automata Machine). The CAM-brain machine is a highly specialized FPGA (Field Programmable Gate Array) based machine that allows to

⁴ Source: <http://www.cs.usu.edu/~degaris/>.

grow and evolve three-dimensional cellular automata based artificial neural networks. De Garis approach for building artificial brains is principally based on the number of neurons only. He believes that the more neurons a system has, the more intelligent it will become, however, without giving any definition of “intelligence” (if there exists one). As Howard Gardner says, “[b]efore intelligence can be enhanced or artificially created, it has to be defined” [26]. The CAM-brain machine is certainly impressive, but brute computational power should definitely not be confound with “intelligence”. For more details on the CAM-brain hardware, see for example [13, 38]. Despite his rather controversial approach, the model he uses is very interesting as it combines growth, evolution, and neural networks in three-dimensions. The CAM-brain machine can evolve neural modules—each consisting of up to 1’152 neurons—with a highly complex an almost arbitrarily interconnection topology in three dimensions. A neural module can receive signals form up to 188 other brain modules and can send signals to up to 64’640 other modules. Axons and dendrites are capable of multiple branching, forming hundreds of connections within each module. The dentritic and axonal structure is directly evolved in hardware using genetic algorithms. Note that, in order to evolve millions of neurons in the CAM-brain, the FPGAs are time shared between multiple neural modules during an fitness evaluation run. The neural model is called *CoDi* for “Collect” and “Distribute” [12, 39]. Figure 4 shows the three different cell types: neuron, dendrite, and axon cells. Each neuron can have up to six axons and up to six dendrites. Note that if there are N axons, there will be $N - 6$ dendrites. A 4-bit accumulator in each neuron sums up incoming signals and fires when a threshold is exceeded. Inputs can further be configured (determined by the neuron’s chromosome) as inhibitory or excitatory and thus either adds or subtracts from the internal accumulator. A dendrite cell can have up to five inputs (and one output) that are fed into a logical gate (e.g., OR or XOR gate). Likewise, an axon cell can have up to five outputs (and one input). During evolution, blank cells perform no function and are used to grow new dendrites and axons. The growth process starts form a blank cellular space with some neurons determined by genetic algorithm. The growth of axons and dendrites then alternates with each clock cycle. Growth directions are guided by the 6-bit chromosome (one for each direction) assigned to each cell. A blank cell that receives a growth signal from one of its neighbor cells becomes for example a dendrite cell or an axon cell. This rather simple mechanism allows to grow very complex three-dimensional networks. After the neural growth phase, the network is switched to the signaling mode.

The CoDi model works with streams of 1’s and 0’s. The interpretation is thus not straightforward and a more sophisticated coding, both efficient and evolvable might be useful. The new coding chosen is called *Spike Interval Information Coding (SIIC)*. The SIIC representation—a sequence of bits or spikes—is inspired by Rieke et al. [53]. The procedure for decoding such a spike train consists of convolving it with a special *convolution filter*. The result

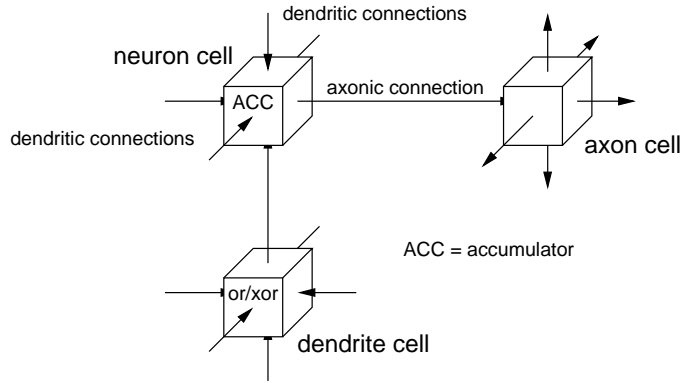


Fig. 4. CAM-brain 3D cellular automata space with neuron, dendrite, and axon cells. Source [39].

obtained is called the *estimated signal*, a time-dependent signal that is output from the neural network module. The inverse process, namely, an algorithm which takes as input a binary numbered time-dependent (“analog”) signal and outputs a spike train, uses a *deconvolution filter*. The SIIC representation thus requires additional convolution and deconvolution filters. For more details about SIIC, see for example [39].

Initial evolution results using the real CAM-brain machine (and not just simulations) are reported in [11]. The experimental results are rather modest and the initial goal of controlling a kitten robot is by far not (yet?) reached. Nevertheless, work on the next generation CAM-brain (BM2) has already started [14].

5 Conclusion

Present computational approaches do not have the performance, flexibility and reliability of neural information processing systems [75]. At least, no artificial system created so far was able to come close to human-like intelligence and behavior and even the most complex artificial systems created are by far much simpler than their natural counterpart. Moreover, many a researcher simply went into the wrong direction by considering too simple and too abstract models. Cahill et al. state: “Simplicity is seductive, and often wrong” [6]. It is widely believed that complex systems cannot be created by hand and that complex information processing structures can only evolve in a complex world rich in information. From that point of view, all man-made technological systems including all human institutions are probably below the threshold of true complexity. Some people make a distinction (yet not a strict one) between *complex* and *complicated* systems. Complicated systems remain understandable even though they might be built up by a large number of components. Complex systems, on the other hand, are not understandable and the closer

one takes a look at them, the more details become visible. Engineering has certainly created many extremely complicated and impressive systems, but they are all “understandable”. Large integrated circuits built up from millions of transistors are a good example for a complicated system whereas a biological cell might be considered as a complex system. Thus, the ultimate goal for the future is the automatic creation of complex and not just complicated artefacts by means of new approaches, e.g., POEtic machines.

All approaches presented in Section 4 went into the right direction (i.e., towards more complex and more rich models) although they only produced very modest results and although some much simpler system currently outperform them. One of the reasons why complex models often are less powerful than simple models is that simulating complex things requires an enormous amount of computational power that we don’t actually have at our disposal.

Let us now come back to Rodney Brooks’ four statements briefly presented in Section 2 already. First, do the above mentioned models not perform well because of wrong parameters? This seems very unlikely. “One would expect that someone would have stumbled by now across a combination of parameters that work qualitatively better than anything else around” [5]. Even though the search spaces are rather huge in all described models, we could thus practically eliminate this kind problem. One might perhaps find some better parameters, but it is really unlikely that there will be a quantum leap in performance. Second, do the models lack complexity? It seems that the basic ideas and the basic components are available, but “[...] we just have not yet put enough of them together in one place, or in one model” [5]. I believe that this is true and that much more complex (and not complicated!) systems with many more hierarchical levels are required. This point is, however, closely related to the actual limitation of computing power. More computing power will not directly result in anything fundamentally new but it might simply help simulating larger parts of complex models. Thus, progress should not be expected from some “stupid” brute force approach but rather from a highly sophisticated (maybe already existing) model. A biologically inspired neuromolecular model that is much more complex than conventional connectionism models has recently been published by Jong-Chen Chen and Ruey-Dong Chen [7]. The network captures intra- and interneuronal information processing on multiple levels. Evolutionary learning, for example, takes place on six different levels. The model is promising as it goes beyond traditional models, i.e, it is more complex, it is biologically plausible, it offers several hierarchical levels, and it has been implemented in hardware (FPGAs) in order to allow to operate on a real-time basis and to provide an architectural paradigm for molecular or neuromolecular electronic technologies. Unfortunately, this model wasn’t able to show some fundamentally new characteristics either. Is it still below a critical complexity level? At this juncture, the question is still out. Another question often raised is whether there is any advantage in implementing things in hard-

ware. It is sometimes argued that in order to create intelligent artefacts we necessarily have to experiment with a physical realization since it is believed that intelligence arises (among other reasons) from an artefact's interaction with its environment. I believe that to understand the mechanisms, we do not need realizations. After all, even the best realization is only a poor "simulation" of a biological system. The situation, however, is different if a hardware realization allows to notably speed up computations, which is quite often the case. From that point of view, hardware realizations will thus very probably continue to play a key role in the future. Hugo de Garis' CAM-brain hardware is a good example for that tendency.

Finally, do the models lack some unimagined features? "One possibility is that some aspect of living systems is invisible to us right now" [5]. This, in fact, is not completely impossible. Philosophers like John Searle [58] and Roger Penrose [50, 51] argue that the intrinsic properties of the brain may not be modeled by any computer and that the human brain involves other mechanisms (e.g., quantum phenomenon, etc.). Might hypercomputation [66] be a candidate for novel computing machines? It is very unlikely, but no definite answer could be given and many a scientist believe that the brain computes within the Turing limit only. Others stressed out that an neural network architecture based on adjusting connection weights is inadequate for cognitive modeling. Gallistel [25], for example, persuasively argued that biological computation requires a register architecture that can store and that operates on representations. Thus, are today's artificial neural networks really well suited for brain-like computation? Brooks mentioned another possibility for new and unimaginable discoveries: some kind of new mathematics. "We may simply not be seeing some fundamental mathematical description of what is going on in living systems and so be leaving it out of our AI and Alife models" [5]. However, none of the mathematical candidate models like dynamical systems, chaos theories, etc. revealed undiscovered fundamental descriptions so far.

In summary, we feel that there is a clear need for new approaches that go beyond traditional models and beyond traditional engineering. Using the terms of philosopher Thomas Kuhn, we just need a "paradigm shift" [41], something different to anything anyone has currently thought of. POETic machines represent one possible direction and an ideal foundation for future research in the creation of novel, highly complex and intelligent systems. I completely agree with IBM's autonomic computing [33] manifest: "Without new approaches, things will only get worse". It's time for a change, "[...] we need to create more complex systems" and to overcome computational limitations. Let us go complex, not simple!

References

- [1] J. C. Astor. A developmental model for the evolution of artificial neural networks: Design, implementation, evaluation. Master's thesis, University of Heidelberg, September 3 1998.
- [2] J. C. Astor and C. Adami. A developmental model for the evolution of artificial neural networks. *Artificial Life*, 6(3):189–218, 2001.
- [3] P. Ball. Life's lessons in design. *Nature*, 409:413–416, January 18 2001.
- [4] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA, 1984.
- [5] R. Brooks. The relationship between matter and life. *Nature*, 409:409–411, January 18 2001.
- [6] L. Cahill, J. L. McGaugh, and N. M. Weinberger. The neurobiology of learning and memory: Some reminders to remember. *Trends in Neurosciences*, 24(10):578–581, October 1 2001.
- [7] J.-C. Chen and R.-D. Chen. Toward an evolvable neuromolecular hardware: A hardware design for a multilevel artificial brain with digital circuits. *Neurocomputing*, 42(1–4):9–34, January 2002.
- [8] W. A. Clark and B. G. Farley. Generalisation of pattern recognition in a self-organising system. In *Proceedings of the Western Joint Computer Conference*, pages 86–91, 1955.
- [9] C. Darwin. *The Origin of Species*. John Murray, London, 1859.
- [10] H. de Garis. *Genetic Programming: GenNets, Artificial Nervous Systems, Artificial Embryos*. PhD thesis, Brussels University, 1992.
- [11] H. de Garis, A. Buller, L. de Penning, T. Chodakowski, and D. Decesare. Initial evolution results on CAM-brain machines (CMBs). In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN'2001)*, volume 2130 of *Lecture Notes in Computer Science*, pages 814–819. Springer-Verlag, Berlin, Heidelberg, 2001.
- [12] H. de Garis, A. Buller, M. Korkin, F. Gers, N. Nawa, and M. Hough. ATR's Artificial Brain ("CAM-Brain") Project: A Sample of what Individual "CoDi-1Bit" Model Evolved Neural Net Modules can do with Digital and Analog I/O. In A. Stoica, D. Keymeulen, and J. Lohn, editors, *Proceedings of the First NASA/DOD Workshop on Evolvable Hardware*, pages 102–110. IEEE Computer Society, Los Alamitos, CA, July, 19-21 1999.
- [13] H. de Garis, M. Korkin, F. Gers, and N. E. Nawa M. Hough. Building an Artificial Brain Using an FPGA Based CAM-Brain Machine. *Applied Mathematics and Computation Journal, Special Issue on Artificial Life and Robotics, Artificial Brain, Brain Computing and Brainware*, 111, 2000.

- [14] H. de Garis and N. Macias. Initial functional and architectural thoughts for a 2nd generation brain building machine (BM2). Available here: <http://www.cs.usu.edu/~degaris/papers/>, 2001.
- [15] F. Dellaert. Toward a biologically defensible model of development. Master's thesis, Case Western Reserve University, Department of Computer Engineering and Science, January 1995.
- [16] F. Dellaert and R. D. Beer. Toward an evolvable model of development for autonomous agent synthesis. In R. A. Brooks and P. Maes, editors, *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 246–257, Cambridge, MA, 1994. A Bradford Book, MIT Press.
- [17] F. Dellaert and R. D. Beer. A developmental model for the evolution of complete autonomous agents. In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, *From Animals to Animats 4: Proceedings of the 4th International Conference on Simulation of Adaptive Behavior*, pages 393–401, Cambridge, MA, 1996. A Bradford Book, MIT Press.
- [18] Georg Dorffner, editor. *Neural Networks and a New AI*. Chapman & Hall, London, 1994.
- [19] P. Eggenberger. Creation of neural networks based on developmental and evolutionary principles. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicod, editors, *Proceedings of the International Conference on Artificial Neural Networks ICANN'97, Lausanne, Switzerland*, volume 1327 of *Lecture Notes in Computer Science*, pages 337–342, Berlin, Germany, 1997. Springer-Verlag.
- [20] D. Endy and R. Brent. Modelling cellular behaviour. *Nature*, 409(6818):391–395, January 18 2001.
- [21] B. G. Farley and W. A. Clark. Simulation of self-organising systems by digital computer. *Institute of Radio Engineers Transactions on Information Theory*, 4:76–84, 1954.
- [22] K. W. Fleischer. *A Multiple-Mechanism Developmental Model for Defining Self-Organizing Geometric Structures*. PhD thesis, California Institute of Technology, Pasadena, CA, 1995.
- [23] K. W. Fleischer and A. H. Barr. A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In C. G. Langton, editor, *Artificial Life III*, volume XVII of *SFI Studies in the Science of Complexity*, pages 389–416, Redwood City, CA, 1994. Addison-Wesley.
- [24] D. B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, second edition, 1999.
- [25] C. R. Gallistel. Foraging for brain stimulation: Toward a neurobiology of computation. *Cognition*, 50:151–170, 1994.
- [26] H. Gardner. *Intelligence Reframed: Multiple Intelligences for the 21st Century*. Basic Books, New York, NY, 1999.

- [27] W. W. Gibbs. Cybernetic cells. *Scientific American*, 285(2):43–47, August 2001.
- [28] F. Gruau. Genetic systems of boolean neural networks with a cell rewriting developmental process. In D. Whitley and S. D. Schaffer, editors, *Combination of Genetic Algorithms and Neural Networks*. IEEE Computer Society Press, Los Alamitos, CA, 1992.
- [29] F. Gruau. *Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm*. PhD thesis, Ecole Normale Supérieure de Lyon, 1994.
- [30] F. Gruau and D. Whitley. The cellular developmental of neural networks: The interaction of learning and evolution. Technical Report 93-04, Ecole Normale Supérieure de Lyon, Institut IMAG, Januar 1993.
- [31] D. Hebb. *The Organization of Behavior*. John Wiley, New York, 1949.
- [32] J. H. Holland. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.
- [33] P. Horn. *Autonomic Computing: IBM's Perspective on the State of Information Technology*. IBM, Armonk, NY, October 2001.
- [34] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York; Oxford, 1993.
- [35] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.
- [36] H. Kitano. Building complex systems using developmental process: An engineering approach. In M. Sipper, D. Mange, and A. Pérez-Urbe, editors, *Proceedings of the Second International Conference on Evolvable Systems (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 218–229, Berlin, Germany, September 1998. Springer-Verlag.
- [37] J. Kodjabachian and J. A. Meyer. Development, learning and evolution in animats. In P. Gaussier and J.-D. Nicoud, editors, *Proceedings of PerAc'94: From Perception to Action*, Los Alamitos, CA, 1994. IEEE Computer Society Press.
- [38] M. Korkin, H. de Garis, F. Gers, and H. Hemmi. CBM (CAM-Brain Machine): A Hardware Tool which Evolves a Neural Net Module in a Fraction of a Second and Runs a Million Neuron Artificial Brain in Real Time. In *Proceedings of the Genetic Programming Conference, GP'97*, Stanford, USA, July 1997.
- [39] M. Korkin, N. E. Nawa, and D. de Garis. A “spike interval information coding” representation of ATR’s CAM-brain machine (CBM). In M. Sipper, D. Mange, and A. Pérez-Urbe, editors, *Proceedings of the Second International Conference on Evolvable Systems (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 256–267, Lausanne, Switzerland, September 1998. Springer-Verlag, Berlin, Germany.

- [40] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- [41] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1962.
- [42] A. Lindenmayer. Mathematical models for cellular interaction in development, parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.
- [43] D. Mange, M. Sipper, A. Stauffer, and G. Tempesti. Toward robust integrated circuits: The embryonics approach. *Proceedings of the IEEE*, 88(4):516–540, April 2000.
- [44] D. Mange and M. Tomassini, editors. *Bio-Inspired Computing Machines: Towards Novel Computational Architectures*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1998.
- [45] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [46] S. Nolfi, O. Miglio, and D. Parisi. Phenotypic plasticity in evolving neural networks. In P. Gaussier and J.-D. Nicoud, editors, *Proceedings of PerAc'94: From Perception to Action*, Los Alamitos, CA, 1994. IEEE Computer Society Press.
- [47] S. Nolfi and D. Parisi. Growing neural networks. Technical Report PCIA-91-15, Institute of Psychology, CNR, Rome, 1991.
- [48] G. M. Odell, G. Oster, P. Albrecht, and B. Burnside. The mechanical basis of morphogenesis I: Epithelial folding and invagination. *Developmental Biology*, 85:446–462, 1981.
- [49] G. Paun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000. First published in a TUCS Research Report, No 208, November 1998, <http://www.tucs.fi>.
- [50] R. Penrose. *The Emperor's New Mind*. Oxford University Press, 1989.
- [51] R. Penrose. *Shadows of the Mind*. Vintage, 1995.
- [52] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.
- [53] F. Rieke, D. Warland, R. de R. van Steveninck, and W. Bialek. *Spikes: Exploring the Neural Code*. A Bradford Book, MIT Press, Cambridge, MA; London, UK, second edition, 1996.
- [54] A. G. Rust. *Developmental Self-Organisation in Artificial Neural Networks*. PhD thesis, University of Hertfordshire, July 1998.
- [55] E. Sanchez, D. Mange, M. Sipper, M. Tomassini, A. Pérez-Urbe, and A. Stauffer. Phylogeny, ontogeny, and epigenesis: Three source of biological inspiration for softening hardware. In *Proceedings of The First International*

- Conference on Evolvable Systems: From Biology to Hardware (ICES96)*, volume 1259 of *Lecture Notes in Computer Science*, pages 35–54, Heidelberg, 1997. Springer-Verlag.
- [56] P. T. Saunders, editor. *Collected Works of A. M. Turing: Morphogenesis*. North-Holland, Amsterdam, 1992.
- [57] S. M. Scheiner. Genetics and evolution of phenotypic plasticity. *Annual Review of Ecological Systems*, 24:35–68, 1993.
- [58] J. R. Searle. *The Rediscovery of the Mind*. MIT Press, Cambridge, MA, 1992.
- [59] M. Sipper, D. Mange, and E. Sanchez. Quo vadis evolvable hardware? *Communications of the ACM*, 42(4):50–56, April 1999.
- [60] M. Sipper and E. M. A. Ronald. A new species of hardware. *IEEE Spectrum*, 37(3):59–64, March 2000.
- [61] M. Sipper and E. Sanchez. Configurable chips meld software and hardware. *IEEE Computer*, 33(1):120–121, January 2000.
- [62] M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Pérez-Urbe, and A. Stauffer. A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. *IEEE Transactions on Evolutionary Computation*, 1(1):83–97, April 1997.
- [63] A. Stauffer, D. Mange, G. Tempesti, and C. Teuscher. BioWatch: A giant electronic bio-inspired watch. In D. Keymeulen, A. Stoica, J. Lohn, and R. Zebulum, editors, *Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware, EH-2001*, pages 185–192. IEEE Computer Society, Los Alamitos, CA, 2001.
- [64] M. P. Stryker. Precise development from imprecise rules. *Science*, 263:1244–1245, 1994.
- [65] C. Teuscher. *Turing’s Connectionism. An Investigation of Neural Network Architectures*. Springer-Verlag, London, September 2001.
- [66] C. Teuscher and M. Sipper. Hypercomputation: Hype or computation? *Communications of the ACM*, 2002. (To appear).
- [67] D. W. Thompson. *On Growth and Form*. Cambridge University Press, Cambridge, UK, 1961.
- [68] M. Tomita. Whole-cell simulation: A grand challenge of the 21st century. *Trends in Biotechnology*, 19(6):205–210, June 1 2001.
- [69] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, B 237:37–72, 1952.
- [70] J. Vaario. *An Emergent Modeling Method for Artificial Neural Networks*. PhD thesis, The University of Tokyo, 1993.
- [71] Jari Vaario and Setsuo Ohsuga. On growing intelligence. In Dorffner [18].

- [72] S. Vogel. *Cat's Paws and Catapults*. Penguin Books, London, UK, 1999.
- [73] M. Weliky and G. Oster. The mechanical basis of cell rearrangement. 1: Epithelial morphogenesis during fundulus epiboly. *Development*, 109:373–386, 1990.
- [74] S. Wermter, J. Austin, and D. Willshaw, editors. *Emergent Neural Computation Architectures Based on Neuroscience. Towards Neuroscience-Inspired Computing*. Number 2036 in Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, Berlin, Heidelberg, 2001.
- [75] S. Wermter, S. Austin, D. Willshaw, and M. Elshaw. Towards novel neuroscience-inspired computing. In Wermter et al. [74], pages 1–19.
- [76] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, September 1999.
- [77] X. Yao and T. Higuchi. Promises and challenges of evolvable hardware. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 29(1):87–97, February 1999.