POEtic Tissue: An Integrated Architecture for Bio-inspired Hardware

Andy M. Tyrrell[1], Eduardo Sanchez[2], Dario Floreano[2], Gianluca Tempesti[2],
Daniel Mange[2], Juan-Manuel Moreno[3], Jay Rosenberg[4], and Alessandro E.P. Villa[5]

[1] University of York, York, UK,
Corresponding author: `amt@ohm.york.ac.uk`
[2] Swiss Federal Institute of Technology at Lausanne, Lausanne, Switzerland
[3] Technical University of Catalunya, Barcelona, Spain
[4] University of Glasgow, UK
[5] University of Lausanne, Lausanne, Switzerland

Abstract. It is clear to all, after a moments thought, that nature has much we might be inspired by when designing our systems, for example: robustness, adaptability and complexity, to name a few. The implementation of bio-inspired systems in hardware has however been limited, and more often than not been more a matter of artistry than engineering. The reasons for this are many, but one of the main problems has always been the lack of a universal platform, and of a proper methodology for the implementation of such systems. The ideas presented in this paper are early results of a new research project, "Reconfigurable POEtic Tissue". The goal of the project is the development of a hardware platform capable of implementing systems inspired by all the three major axes (phylogenesis, ontogenesis, and epigenesis) of bio-inspiration, in digital hardware.

1   Introduction

The implementation of bio-inspired systems in silicon is quite difficult, due to the sheer number and complexity of the biological mechanisms involved. Conventional approaches exploit a very limited set of biologically-plausible mechanisms to solve a given problem, but often cannot be generalized because of the lack of a *methodology* in the design of bio-inspired computing machines. This lack is due to the heterogeneity of the hardware solutions adopted for bio-inspired systems, which is itself due to the lack of *architectures* capable of implementing a wide range of bio-inspired mechanisms.

The aim of this paper is to present some of the preliminary ideas developed in the framework of a new research project, called "Reconfigurable POEtic Tissue" (or "POEtic" for short), recently started under the aegis of the European Community. After a short introduction to the POE model for the design of bio-inspired hardware, the paper will present an outline of the POEtic project and of the main ideas behind the development of the tissue, to finish with some of the preliminary results in the form of an overall architecture designed for the implementation of bio-inspired systems.

2   The POE Model

The goal of the POEtic project is the

> "... *development of a flexible computational substrate inspired by the evolution-ary, developmental and learning phases in biological systems.*"

Biological inspiration in the design of computing machines finds its source in essentially three biological models [1]: *phylogenesis* (P), the history of the evolution of the species, *ontogenesis* (O), the development of an individual as orchestrated by his genetic code, and *epigenesis* (E), the development of an individual through learning processes (nervous system, immune system) influenced both by the genetic code (the innate) and by the environment (the acquired). These three models share a common basis: the genome.

What follows is a very concise and simplified description of the three models and of their application to the conception of electronic hardware. A much more detailed analysis of the three models and of existing hardware implementations, in the form of internal project reports, is available online on the POEtic project website [2].

2.1 Phylogenesis

On the phylogenetic axis we find systems inspired by the processes involved in the evolution of a species through time, i.e. the evolution of the genome. The process of evolution is based on alterations to the genetic information of a species, occurring through three basic mechanisms: *selective reproduction*, *crossover* and *mutation*.

These mechanisms are, by nature, non-deterministic. This represents both their strength and their weakness, when applied to the world of electronics. It is a strength, because they are fundamentally different from traditional algorithms and thus are potentially capable of solving problems which are intractable by deterministic approaches. It is a weakness, because computers are inherently deterministic (it is very difficult, for example, to generate a truly random number, a basic requirement for non-deterministic computation, in a computer).

Even with this disadvantage, algorithms that exploit phylogenetic mechanisms are carving themselves a niche in the world of computing. These algorithms, commonly referred to as evolutionary algorithms (a label that regroups domains such as genetic algorithms [3], evolutionary programming [4], and genetic programming [5]), are usually applied to problems which are either too ill-defined or intractable by deterministic approaches, and whose solutions can be represented as a finite string of symbols (which thus becomes the equivalent of the biological genome). An initial, random population of individuals (i.e., of genomes), each representing a possible solution to the problem, is iteratively "evolved" through the application of mutation and crossover. The resulting sequences are then evaluated on the basis of their efficiency in solving the given problem (fitness function) and the best solutions are in turn evolved. This approach, is not guaranteed to find the best possible solution to a given problem, but can often find an "acceptable" solution more efficiently than deterministic approaches.

It appears, then, that the phylogenetic axis has already provided a considerable amount of inspiration to the development of computer systems. To date, however, its impact has been felt mostly in the development of software algorithms, and only mar-

ginally in the conception of digital hardware. Koza et al. pioneered the attempt to apply evolutionary strategies to the synthesis of electronic circuits when they applied genetic algorithms to the evolution of a three-variable multiplexer and of a two-bit adder [5]. Also, evolutionary strategies have been applied to the development of the control circuits for autonomous robots [6], [7]. Other research groups are active in this domain [8], [9] however, technical issues have posed severe obstacles to the development of evolvable electronic hardware [10]. A possible solution for the POEtic project is given in [11].

2.2 Ontogenesis

The ontogenetic model concerns the development of a single multi-cellular biological organism. This process exploits a set of very specific mechanisms to define the *growth* of the organism, i.e. its development from a single mother cell (the *zygote*) to the adult phase. The zygote divides, each offspring containing a copy of the genome (cellular division). This process continues (each new cell divides, creating new offspring, and so on), and each newly formed cell acquires a functionality (e.g., liver cell, or epidermal cell) depending on its surroundings, i.e., its position in relation to its neighbors (cellular differentiation).

Cellular division is therefore a key mechanism in the growth of multi-cellular organisms, impressive examples of massively parallel systems: the 6×10^{13} cells of a human body, each a relatively simple elements, work in parallel to accomplish extremely complex tasks (the most outstanding being, of course, intelligence). If we consider the difficulty of programming massively parallel computers, biological inspiration could provide some relevant insights on how to handle massive parallelism in silicon.

A fundamental feature of biological organisms is that each cell contains the blueprint for the entire organism (the genome), and thus can potentially replace any other cell [12]: no single cell is indispensable to the organism. In fact, cells are ceaselessly being created and destroyed in an organism, a mechanism at the base of one of the most interesting properties of multi-cellular organisms: healing.

The ontogenetic axis has been almost completely ignored by computer scientists, despite a promising start in the fifties with the work of John von Neumann, who developed a theoretical model of a universal constructor, a machine capable of constructing any other machine, given its description [13]. Given a description of itself, the universal constructor can then self-replicate, a process not unlike cellular division.

Unfortunately, electronic circuits in the 1950s were too primitive to allow von Neumann's machine to be realized, and the concept of self-replicating machines was thus set aside for a long time. Probably the main obstacle to the development of self-replicating machines was the impossibility of physically creating self-replicating hardware. In fact, such a process requires a means to transforming information (i.e. the description of a machine) into hardware, and such a means was unpractical until the introduction of programmable logic circuits (FPGAs), which provide a partial solution to the problem.

The difficulty of implementing processes such as growth in silicon has limited research on this model. The Logic Systems Laboratory at EPFL began to investigate ontogenesis of electronic circuits with its Embryonics project [14]. Another aspect of ontogenesis, morphogenesis (that is, the generation of the *shape* of an organism) has

been the subject of some research [15], [16], and recently the issue of *development* is attracting growing attention as a source of inspiration for artificial systems [17]. More details for this project can be found in [18].

2.3 Epigenesis

The human genome contains approximately $3x10^9$ bases. An adult human body contains around $6x10^{13}$ cells, of which approximately 10^{10} are neurons, with 10^{14} connections. In this case, and in many complex organisms, the genome cannot contain enough information to completely describe all the cells and synaptic connections of an adult organism.

There must therefore exist a process that allows the organism to increase in complexity as it develops. This process can be labeled *epigenesis*, and includes the development of the nervous, immune, and endocrine systems.

Epigenetic mechanisms have already had considerable impact on computer science, and particularly on software design, notably through the concept of learning. The parallel between a computer and a human brain dates to the very earliest days of the development of computing machines, and led to the development of the related fields of artificial intelligence [19] and artificial neural networks.

All living organisms interact with their environment and are capable of responding to sensory inputs. In many cases this behaviour is learnt over a period of time, after which a specific stimulus will trigger the same response each time, sometimes the response may depend on the context of the stimulus. It is now recognised that such behaviour is controlled by neurones or nerve cells and their interactions. The majority of neurones underlying learning are spiking neurones – they communicate through neuronal spike trains. In this project we will focus on the use of spiking (or pulsed) neuronal models to implement learning [20]. These devices are based on leaky integrate-to-threshold and fire neurone models, we will adapt these for implementation in hardware. Learning in spiking neural networks usually takes inspiration from Hebb's ideas, some recent work has suggested that learning in spiking neural networks may be possible using very coarse grained synaptic weightings [21], which are particularly suited to hardware implementation. A view on the way forward for the POEtic project is given in [22].

A more recent addition in the general area of hardware systems and epigenetic processes are artificial immune systems [23]. Here the sophisticated mechanisms associated with "fault tolerance" in nature have been adapted for electronic hardware system designs.

3   The POEtic Project

While each of the three 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 a single system is a challenge yet to be met. The aim of the POEtic project [2] is the development of a computational substrate optimized for the implementation of digital systems inspired by all of the three above-mentioned models.

The POEtic tissue is a multi-cellular, self-contained, flexible, and physical substrate designed to interact with the environment, to develop and dynamically adapt its functionality through a process of evolution, growth, and learning to a dynamic and

partially unpredictable environment, and to self-repair parts damaged by aging or environmental factors in order to remain viable and perform similar functionalities.

Following the three models of bio-inspiration, the POEtic tissue will be designed as a three-layer structure (figure 1):

- The phylogenetic model acts on the genetic material of a cell. Each cell can contain the entire genome of the tissue. Typically, in the architecture defined above, it could be used to find and select the genes of the cells for the *genotype layer*.
- The ontogenetic model concerns the development of the individual. It should act mostly on the *mapping or configuration layer* of the cell, implementing cellular differentiation and growth. In addition, ontogenesis will have an impact on the overall architecture of the cells where self-repair (healing) is concerned.
- The epigenetic model modifies the behavior of the organism during its operation, and is therefore best applied to the *phenotype layer*.
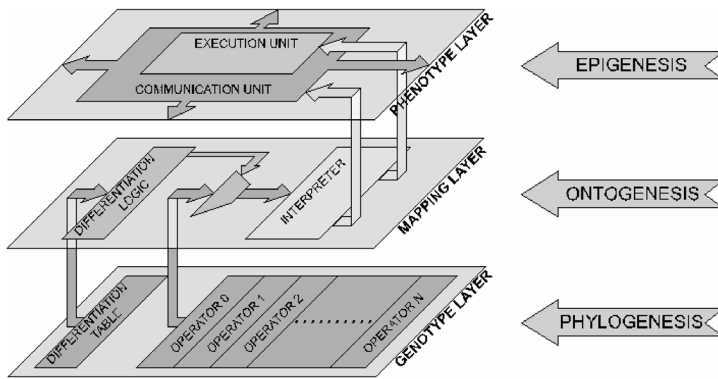

Fig. 1. The three organizational layers of the POEtic project.

Defining separate layers for each model has a number of advantages, as it allows the user to decide whether to implement any or all models for a given problem, and lets the structure of each layer to be adapted to the model. This adaptability, which will be detailed in the next section, is achieved by implementing the cells on a *molecular substrate*, in practice a surface of programmable logic.

Beside this combination of the three main axes of biological self-organization, the proposed tissue presents two more innovative hardware-oriented aspects:

- a layered hardware structure that matches the three axes of biological organization;
- an input/output interface with the external world that allows each cells to perceive and modify its environment when and where necessary.

4   POE Integration

There are at least two advantages in a circuit that integrates phylogenesis, ontogenesis, and epigenesis. The first is that each of the corresponding algorithms can be optimised for a particular class of problems. The second advantage comes from the com-

bination of the different self-organization modalities. For example, instead of combining evolutionary algorithms with off-the-shelf learning algorithms, it will be possible to genetically encode and evolve the rules by which the phenotype plane of the tissue can dynamically adapt to external or internal perturbations without need of additional evolution. It has been shown that in the case of simulated neural networks, by evolving the adaptation mechanisms instead of a fixed configuration pattern, the evolutionary circuits cannot capitalize upon genetically determined solutions that fit a particular environment, but are rather forced to evolve the ability to develop on the fly the structure required by their specific environment [24]. This is quite interesting because it potentially offers a solution to the strong dependency of conventional evolved circuits to the evolutionary environment.

4.1 Cellular Structure

In contrast to conventional cellular electronic devices (such as Field Programmable Gate Arrays), each cell in the POEtic tissue contains the entire genome for the whole tissue. This solution, which is also closer to biological reality, increases the robustness of the system, allowing individual cells to reconfigure their function for growth, self-replication, and self-repair at a local level.

The ontogenetic concept of growth has been largely ignored in the past in the development of bio-inspired systems. The reason is obvious: growth implies the "creation" of new material during the lifetime of an organism (and therefore during the operational life of a machine), a feature which remains impossible to this day in the world of electronic circuits.

However, while the silicon substrate of an electronic circuit cannot be modified, a reconfigurable circuit like the POEtic tissue allows its function to be defined after fabrication, and thus potentially modified at runtime. The tissue allows the realization of a growth process based on information rather than matter.

The process of growth is based on cellular division and on cellular differentiation. From the standpoint of both biological organisms and electronic circuits, these two processes are made possible by the existence of the same genome inside each cell. Despite their structural similarity, each cell can specialize to perform a unique function by "executing" only a specific part of the genetic program depending on the cell's location in the organism. This differentiation allows multi-cellular organisms, including the POEtic tissue, to display behaviors of a greater complexity than those of single-cell organisms.

Finally, biological multi-cellular organisms are capable of surviving an astounding amount of damage. The "fault tolerance" of biological organisms is based on a set of extremely complex systems ranging from the organism-wide immune system to single-cell error correction. The ontogenetic features of the tissue will include a hierarchy of fault-tolerance mechanisms, acting "behind the scenes" with little or no effect on the operation of the machine.

4.2 Layered Organization

The structure in figure 1, naturally accommodates the building blocks and processes that regulate biological life. The genetic material specifying the functionality of the circuit is stored in the genotype layer, the rules and mechanisms of gene expression

that map the genotype into a phenotype are stored and executed in the configuration layer, and the resulting function performed by each cell is implemented and executed in the phenotype layer.

The explicit distinction in three layers allows the study and implementation of complex genotype-to-phenotype mappings that can exploit growth and reconfiguration properties of the tissue, reduce the genome length, and eventually improve the evolvability of the tissue.

The mapping between genotypes and phenotypes is mainly a one-to-one function in artificial evolutionary systems. This relatively straightforward strategy does not capture the complex, but highly regulated, mechanisms of gene expression that characterise the development of biological organisms and barely supports the creation of genetic building blocks that could be exploited by crossover operators. At the same time, since in one-to-one mappings the length of the genotype is proportional to the complexity of the phenotype, large structures such as neuronal circuits are difficult to evolve. POEtic tissue will provide a substrate that naturally supports the implementation in the configuration layer of smarter and more biologically inspired mechanisms of gene expression. This structure, together with the cellular organization of the tissue, requires the evolutionary mechanisms that go beyond conventional genetic algorithms in that they capitalize and leave more space to ontogenetic and epigenetic mechanisms of dynamic self-organization of the tissue.

Finally, the presence of three distinct hardware layers allows the implementation of different mechanisms of self-organization in isolation without necessarily integrating them all. For example, one may decide to implement a given neuronal structure and its learning mechanisms on the circuit without going into evolution and gene expression. Similarly, another may wish to implement conventional electronic functions and exploit only the self-configuration and self-repair properties of the circuit without resorting to evolution and learning. This flexibility of the proposed circuit will indeed be exploited in the project to study the different aspects of self-organization and gradually integrate them all in the final POEtic tissue.

5   A Bio-inspired Architecture

Adaptation and specialization are key features of all biological organisms. As a consequence, no bio-inspired architecture can be rigid, but must rather be able to adapt and specialize depending on the desired application. The POEtic tissue will exploit reconfigurable logic to the fullest extent to allow for these characteristics: the bio-inspired machines will consist of two-dimensional arrays of *cells*, where each cell is a small, fully reconfigurable, processing element.

The role of the tissue is to provide a *molecular substrate* for the implementation of the cells: as organic cells are constituted by molecules, so artificial cells will be constituted by the programmable logic elements of the POEtic circuit. The three layers of the POEtic systems (figure 1) will all be implemented on this substrate.

Integrating reprogrammability in an architecture, however, is far from trivial. The heterogeneity of the applications that can potentially be implemented poses considerable design problems, particularly for the control of the processing elements. A novel kind of architecture needs to be developed to fully exploit the potential of reconfigurable logic for bio-inspired systems.

The three-layer architecture proposed in the POEtic tissue, together with its implementation on the molecular substrate, allows the hardware structure of each layer to be adapted to the bio-inspired mechanism that it implements. What follows is an outline of a plausible structure for each of the three layers, keeping in mind that the reprogrammability of the tissue will allow any or all of these layers to be essentially silent, should the bio-inspired mechanism concerned not be required.

5.1 The Genotype Layer

The genotype layer (figure 2) contains the genetic information of the organism, which consists essentially of:

- A *set of operators*, which define the set of all possible functions for the cell. The quantity and functionality of the operators depends essentially on the application. For example, if the phenotype layer implements an array of neurons, one could imagine that the operators will consist of the internal parameters for the neuron, together with a set of initial connection weights. Instead, if the phenotype implements, say, a sound processing surface, the operators will consist of different filters and sound generation functions.
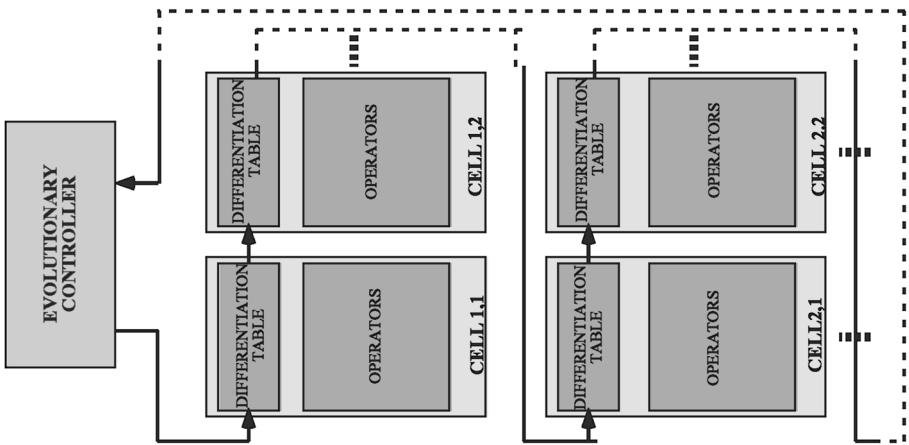

Fig. 2. The genotype layer in an array of cells and the external controller for evolution.

- A *differentiation table*, used to determine which operators will be implemented in which cell. The table contains a compact, perhaps redundant, representation of the operators to be implemented by the cells. The main advantage of maintaining a separate table storing this information is its *size*: the table will represent an extremely compact representation of the structure of the system, and evolutionary mechanisms in hardware will be applied to the table, rather than to the operators themselves. The differentiation table also plays an important role in the mapping layer, as described below.

The key feature of the genotype layer lies in the presence of a set of *dedicated connections*, used to provide user access to the differentiation tables of all the cells in the

system. The advantage of this kind of access lies in the presence of an *on-chip micro-controller*, whose function is to access the tables and to manipulate their contents through a set of user-defined evolutionary mechanisms.

5.2 The Configuration or Mapping Layer

The configuration layer of the POEtic systems is designed to implement the processes of cellular differentiation and cellular division (growth), the bases of the ontogenetic model (the fault-tolerance aspect of the model is also partially present in this layer, but is in general distributed throughout the layers of the array).

In its simplest conception (figure 3), the function of the configuration layer is to select which operator will be implemented by the cell. Each cell must be able to identify its position within the array. A set of coordinates, incremented in each cell and transmitted along the horizontal and vertical axes, is one approach. Once the cell has established its position, it can use this to select from the differentiation table which operator to express. The operator will then be interpreted to generate the appropriate control signals for the phenotype layer.

However, this approach is only a very basic solution to achieve cellular differentiation, and does not exploit many of the most interesting features of the development process.

For example, the solution of figure 3 is limited to a *one-to-one mapping* between genotype and phenotype. From an evolutionary standpoint, however, this approach imposes considerable disadvantages in the application of phylogenetic mechanisms. More complex mappings, based on redundant coding, are being researched.

Also, the coordinate-based approach is not well suited to the implementation of processes that want to draw inspiration from the natural process of *growth*, a key mechanism in the development of biological organisms. Automatic routing between cells will remove the need for a coordinate system, connectivity will be symbolic. More complex differentiation mechanisms, based on L-systems [16] and on cell-signalling gradients, are currently under study.
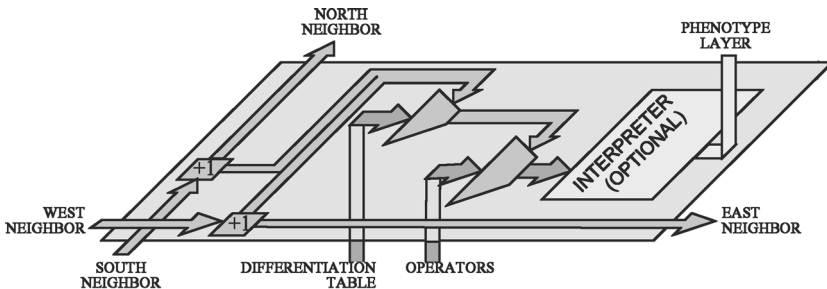

Fig. 3. A possible structure for the configuration or mapping layer.

5.3 The Phenotype Layer

The phenotype layer is probably the most application-dependent layer of the three. If the final application is a "conventional" neural network, the phenotype layer of the cell will simply consist of an artificial neuron. The architecture of the neuron is a

choice left to the user. The POEtic project will concentrate on *spiking neurons* but this choice does not imply that the tissue will be limited to such a model.

However, since the POEtic tissue is meant to allow the implementation of bio-inspired systems that do not necessarily involve exclusively neural-like cells, a more general-purpose phenotype architecture will be required.

A simplified layout of a candidate architecture of the type that is being investigated in the POEtic project is shown in figure 4. It consists of two main units:

- An execution unit, consisting of a set of application-dependent *resources* (e.g., adders, and counters). The resources are defined by the user at design time, and are accessed through a set of input and output ports.
- In view of the massive amounts of connectivity of bio-inspired systems, the communication unit will probably have to rely on *serial communication*. The communication unit is seen by the execution unit as just another resource, with an input and output port.

This kind of architecture has two main advantages: it can support specialisation and adaptation through the modification of the resources depending on the application and it can be integrated in a design environment easily.
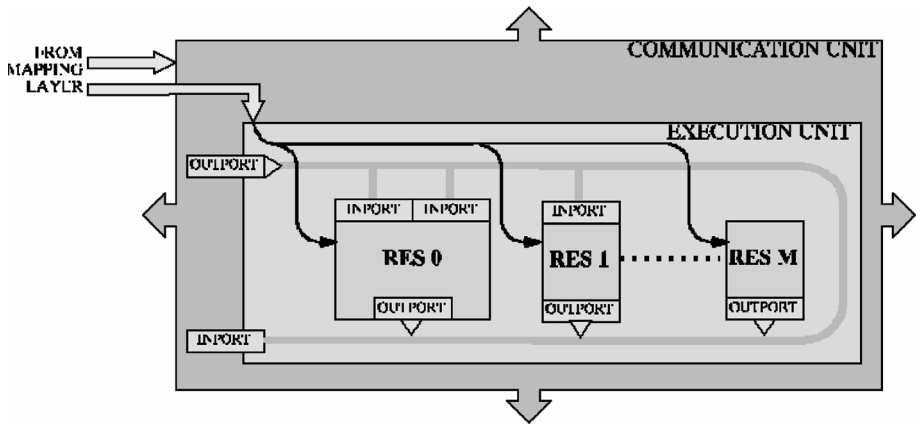

Fig. 4. Outline of a possible architecture for the phenotype layer.

5.4 A Design Methodology

The definition of three separate layers for the different models of bio-inspiration has the advantage of being able to select which of the models are to be used for a given problem. This feature is made possible by the complete programmability of the systems: all the three layers are implemented on the same substrate, a programmable digital logic circuit that incorporates dedicated features for all three models. The implementation of the layers in programmable logic adds the possibility of adapting the architecture of the system to the application, a flexibility that will allow the definition of an integrated design environment. Such an environment should include a *customizable design flow* allowing the user to concentrate on the desired approaches. A typical application-driven design flow for a bio-inspired system could then be:

- Based on the application, the user defines which models will be implemented in the system. The user then accesses a *library of operators* and selects those that will be required by the application. The phylogenetic model can be applied at this stage both to the selection of the functions and to the evolution of novel functions to be added to the library. The presence of the epigenetic model is defined at this stage by selecting functions capable of implementing learning processes.
- The environment will then integrate the desired functions into a *totipotent cell*, that is, a processing element capable of realizing any of the functions. This step represents the implementation of the ontogenetic model. Other optional ontogenetic features, such as growth and self-repair, can be added automatically at this stage.
- Finally, the design environment assembles the cells into a two-dimensional array, integrates the user-specified I/O modules, and generates a configuration for the programmable logic array that will implement the system.

6   Conclusions

Systems inspired by biological mechanisms are carving themselves a niche in several areas of computation, but the solutions adopted often cannot be generalised for the lack of a universal architecture capable of integrating the different approaches.

The POEtic project does not necessarily propose such an architecture: what it proposes is an hardware substrate capable of implementing bio-inspired systems. Its aim is to develop the *molecules* necessary for the development of cellular systems. The usefulness and the practicality of the molecular surface will obviously be validated on a number of bio-inspired applications, ranging from neural networks to music synthesis. But beyond the applications developed within the project, the POEtic tissue will allow the implementation of a variety of bio-inspired systems, and it is our hope that the circuits created in this project will find users well beyond our group. This paper by no means presents the final version of the POEtic system: the project has started only recently, and the ideas presented here are simply early results of our research. We invite all readers to access our website [2], where we regularly post the progress reports and deliverables of the project.

References

1. E. Sanchez, D. Mange, M. Sipper, M. Tomassini, A. Perez-Uribe, A. Stauffer. "Phylogeny, Ontogeny, and Epigenesis: Three Sources of Biological Inspiration for Softening Hardware". Lecture Notes in Computer Science, vol. 1259, Springer-Verlag, Berlin, 1997, pp. 35-54.
2. http://www.poetictissue.org
3. M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.
4. D.B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.
5. J.R. Koza. *Genetic Programming*. The MIT Press, Cambridge, MA, 1992.
6. D. Floreano, J. Urzelai.. "Evolutionary Robots with on-line self-organization and behavioral fitness". *Neural Networks*, 13, 431-443.
7. Krohling, R., Zhou, Y. and Tyrrell, A.M. 'Evolving FPGA-based robot controllers using an evolutionary algorithm', 1st International Conference on Artificial Immune Systems, Canterbury, September 2002.
8. T. Higuchi, M. Iwata, I. Kajitani, M. Murakawa, S. Yoshizawa. "Hardware evolution at gate and function level". *Proc. of Biologically Inspired Autonomous Systems: Computation, Cognition, and Action*, Durham, NC, March 1996.
9. A. Thompson, I. Harvey, and P. Husbands. "Unconstrained evolution and hard consequences", *Towards Evolvable Hardware*, Springer, 1996, pp.136-165.
10. Canham, R.O. and Tyrrell, A.M. 'Evolved Fault Tolerance in Evolvable Hardware' Congress on Evolutionary Computation, Hawaii, pp 1267-1272, May 2002.
11. Roggen, D., Floreano, D., Mattiussi, C. 'A Morphogenetic System as the Phylogenetic Mechanism of the POEtic Tissue. Elsewhere in this volume.
12. C. Ortega, A. Tyrrell, "MUXTREE revisited: Embryonics as a Reconfiguration Strategy in Fault-Tolerant Processor Arrays", Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Berlin, 1998, pp. 206-217.
13. J. von Neumann. *The Theory of Self-Reproducing Automata*. A. W. Burks, ed. University of Illinois Press, Urbana, IL, 1966
14. D. Mange, M. Sipper, A. Stauffer, G. Tempesti. "Towards Robust Integrated Circuits: The Embryonics Approach". *Proceedings of the IEEE*, vol. 88, no. 4, April 2000, pp. 516-541.
15. F. Gruau. "Genetic systems of boolean neural networks with a cell rewriting developmental process", *Combination of Genetic Algorithms and Neural Networks*. IEEE Press, Los Alamitos, CA, 1992.
16. A. Lindenmayer. "Mathematical models for cellular interaction in development, parts I and II". *Journal of Theoretical Biology*, 18:280-315, 1968.
17. H. Kitano. "Building complex systems using developmental process: An engineering approach", Lecture Notes in Computer Science, vol. 1478, Springer-Verlag, Berlin, 1998. pp. 218-229.
18. G Tempesti, D. Roggen, E Sanchez, Y Thoma, R. Canham, A.M. Tyrrell. 'Ontogrnetic Development and Fault Tolerance in the POEtic Tissue', Elsewhere in this volume.
19. P.H. Winston. *Artificial Intelligence*. Addison-Wesley, Reading, MA, 3rd edition, 1992.
20. Maass, W and Bishop C.M. "*Pulsed Neural Networks*", MIT Press, 1999.
21. Fusi, S. "Long term memory: Encoding and storing strategies of the brain", Neurocomputing, 38-40, 1223-1228.
22. Eriksson, J., Torres, O., Mitchell, A., Tucker, G., Lindsay, K., Halliday, D., Rosenberg, J., Moreno, J., Villa, A. 'Spiking Neural Networks for Reconfigurable POEtic Tissue. Elsewhere in this volume.
23. Bradley, D.W. and Tyrrell, A.M. 'Immunotronics: Novel Finite State Machine Architectures with Built in Self Test using Self-Nonself Differentiation', IEEE Transactions on Evolutionary Computation, Vol 6, No 3, pp 227-238, June 2002.
24. Urzelai, J. and Floreano, D. (2001) Evolution of Adaptive Synapses: Robots with Fast Adaptive Behavior in New Environments. *Evolutionary Computation*, **9**, 495-524.