In J. Mira, A. Prieto (eds.)
Bio-Inspired Applications of Connectionism, Proc. of the Sixth International Work Conference on Artificial and Natural Neural Networks (IWANN2001), Granada, Spain, Part II, pp. 1-13.
LNCS 2085, Springer-Verlag, Berlin, 2001.

From Embryonics to POEtic Machines

Daniel Mange, André Stauffer, Gianluca Tempesti, and Christof Teuscher

Logic Systems Laboratory, Swiss Federal Institute of Technology,
CH-1015 Lausanne, Switzerland
daniel.mange@epfl.ch

Abstract. The space of bio-inspired hardware can be partitioned along three axes: phylogeny, ontogeny, and epigenesis. We refer to this as the POE model. Our Embryonics (for embryonic electronics) project is situated along the ontogenetic axis of the POE model and is inspired by the processes of molecular biology and by the embryonic development of living beings.
We will describe the architecture of multicellular automata that are endowed with self-replication and self-repair properties. In the conclusion, we will present our major on-going project: a giant self-repairing electronic watch, the BioWatch, built on a new reconfigurable tissue, the electronic wall or e–wall.

1 Introduction[1]

1.1 The POE model of bio-inspired systems

The space of *bio-inspired* hardware systems can be partitioned along three major axes: *phylogeny*, *ontogeny*, and *epigenesis*; we refer to this as the *POE model* [7], [4]. Where nature is concerned, the distinction between the axes cannot be easily drawn. Indeed the definitions themselves may be subject to discussion. Sipper *et al.* [7] thus defined each of the above axes within the framework of the POE model as follows: the phylogenetic axis involves *evolution*, the ontogenetic axis involves the *development* of a single individual from its own genetic material, essentially without environmental interactions, and the epigenetic axis involves *learning* through environmental interactions that take place after formation of the individual. As an example, consider the following three paradigms, whose hardware implementations can be positioned along the POE axes: (P) evolutionary algorithms are the (simplified) artificial counterpart of phylogeny in nature, (O) multicellular automata are based on the concept of ontogeny, where a single mother cell gives rise, through multiple divisions, to a multicellular organism, and (E) artificial neural networks embody the epigenetic process, where the system's synaptic weights and, sometimes, its topological structure change through interactions with the environment.

[1] Based on the following paper: D. Mange, M. Sipper, A. Stauffer, G. Tempesti. "Toward Robust Integrated Circuits: The Embryonics Approach". *Proceedings of the IEEE*, Vol. 88, No. 4, April 2000, pp. 516-541.

This paper is a description of bio-inspired hardware systems along the ontogenetic axis of the POE model: the Embryonics project. We conclude this presentation with directions for future research, based on the POE model, i.e. new *POEtic machines*.

1.2 Embryonics = embryonic electronics

Our *Embryonics* project is inspired by the basic processes of molecular biology and by the embryonic development of living beings. By adopting certain features of cellular organization, and by transposing them to the two-dimensional world of integrated circuits on silicon, we will show that properties unique to the living world, such as *self-replication* and *self-repair*, can also be applied to artificial objects (integrated circuits) [8], [1]. Self-repair allows partial reconstruction in case of a minor fault, while self-replication allows complete reconstruction of the original device in case of a major fault. These two properties are particularly desirable for complex artificial systems requiring very high level of reliability.

2 A survey of Embryonics

2.1 Biological inspiration

The majority of living beings, with the exception of unicellular organisms such as viruses and bacteria, share three fundamental features.

1. *Multicellular organization* divides the organism into a finite number of *cells*, each realizing a unique function (neuron, muscle, intestine, etc.). The same organism can contain multiple cells of the same kind.
2. *Cellular division* is the process whereby each cell (beginning with the first cell or *zygote*) generates one or two daughter cells. During this division, all of the genetic material of the mother cell, the *genome*, is copied into the daughter cell(s).
3. *Cellular differentiation* defines the role of each cell of the organism, that is, its particular function (neuron, muscle, intestine, etc.). This specialization of the cell is obtained through the expression of part of the genome, consisting of one or more *genes*, and depends essentially on the physical position of the cell in the organism.

A consequence of these three features is that each cell is "universal", since it contains the whole of the organism's genetic material, the genome. Should a minor (wound) or major (loss of an organ) trauma occur, living organisms are thus potentially capable of self-repair (cicatrization) or self-replication (cloning or budding) [8].

2.2 The organism's features: multicellular organization, cellular differentiation, and cellular division

The environment in which our quasi-biological development occurs is imposed by the structure of electronic circuits, and consists of a finite (but arbitrarily large) two-dimensional surface of silicon. This surface is divided into rows and columns, whose intersections define the cells. Since such cells (small processors and their memory) have an identical physical structure (i.e., an identical set of logic operators and of connections), the cellular array is homogeneous. As the program in each cell (our artificial genome) is identical, only the state of the cell (i.e., the contents of its registers) can differentiate it from its neighbors.

In this Section, we first show how to implement in our artificial organisms the three fundamental features of multicellular organization, cellular differentiation, and cellular division, by using a generic and abstract six-cell example. *Multicellular organization* divides the artificial organism (ORG) into a finite number of cells (Figure 1). Each cell ($CELL$) realizes a unique function, defined by a sub-program called the *gene* of the cell and selected as a function of the values of both the horizontal (X) and the vertical (Y) coordinates (in Figure 1, the genes are labeled A to F for coordinates $X, Y = 1, 1$ to $X, Y = 3, 2$). Our final artificial genome will be divided into three main parts: the *operative genome* (OG), the *ribosomic genome* (RG), and the *polymerase genome* (PG). Let us call operative genome (OG) a program containing all the genes of an artificial organism, where each gene (A to F) is a sub-program characterized by a set of instructions and by the cell's position (coordinates $X, Y = 1, 1$ to $X, Y = 3, 2$). Figure 1 is then a graphical representation of organism ORG's operative genome.

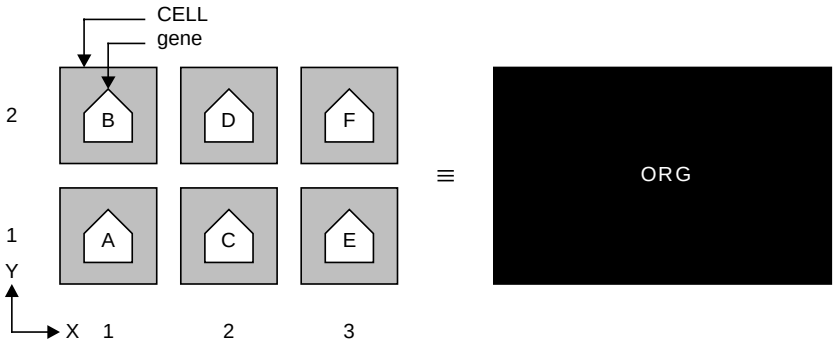

Fig. 1. Multicellular organization of a 6-cell organism ORG.

Let then each cell contain the entire operative genome OG (Figure 2a): depending on its position in the array, i.e. its place within the organism, each cell can then interpret the operative genome and extract and execute the gene which defines its function. In summary, storing the whole operative genome in each

cell makes the cell universal: given the proper coordinates, it can execute any one of the genes of the operative genome and thus implement *cellular differentiation*. In our artificial organism, any cell $CELL[X, Y]$ continuously computes its coordinate X by incrementing the coordinate WX of its neighbor immediately to the west (Figure 2b). Likewise, it continuously computes its coordinate Y by incrementing the coordinate SY of its neighbor immediately to the south. Taking into consideration these computations, Figure 3 shows the final operative genome OG of the organism ORG.

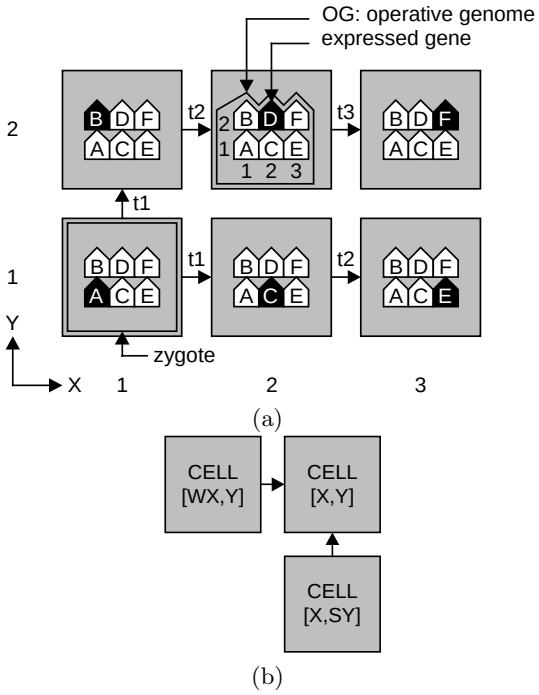

Fig. 2. Cellular differentiation and division. (a) Global organization. (b) Central cell $CELL[X, Y]$ with its west neighbor $CELL[WX, Y]$ and its south neighbor $CELL[X, SY]$; $X = WX + 1$; $Y = SY + 1$; $t1...t3$: three cellular divisions.

At startup, the first cell or *zygote* (Figure 2a), arbitrarily defined as having the coordinates $X, Y = 1, 1$, holds the one and only copy of the operative genome OG. After time $t1$, the genome of the zygote (*mother* cell) is copied into the neighboring (*daughter*) cells to the east ($CELL[2, 1]$) and to the north ($CELL[1, 2]$). This process of *cellular division* continues until the six cells of the organism ORG are completely programmed (in our example, the farthest cell is programmed after time $t3$).

```
OG: operative genome
X = WX+1
Y = SY+1
case of X,Y:
  X,Y = 1,1: do gene A
  X,Y = 1,2: do gene B
  X,Y = 2,1: do gene C
  X,Y = 2,2: do gene D
  X,Y = 3,1: do gene E
  X,Y = 3,2: do gene F
```

Fig. 3. The operative genome OG of the organism ORG.

2.3 The organism's properties: organismic self-replication and organismic self-repair

The *self-replication* or *cloning of the organism*, i.e. the production of an exact copy of the original, rests on two assumptions.

1. There exists a sufficient number of spare cells in the array (at least six in the example of Figure 4) to contain the additional organism.
2. The calculation of the coordinates produces a cycle ($X = 1 \rightarrow 2 \rightarrow 3 \rightarrow 1...$ and $Y = 1 \rightarrow 2 \rightarrow 1...$ in Figure 4, implying $X = (WX + 1)$ modulo 3 and $Y = (SY + 1)$ modulo 2).

As the same pattern of coordinates produces the same pattern of genes, self-replication can be easily accomplished if the program of the operative genome OG, associated with the homogeneous array of cells, produces several occurrences of the basic pattern of coordinates. In our example (Figure 4), the repetition of the vertical coordinate pattern ($Y = 1 \rightarrow 2 \rightarrow 1 \rightarrow 2$) in a sufficiently large array of cells produces one copy, the *daughter organism*, of the original *mother organism*. Given a sufficiently large space, the self-replication process can be repeated for any number of specimens in the X and/or the Y axes.

In order to implement the *self-repair of the organism*, we decided to use spare cells to the right of the original organism (Figure 5). The existence of a fault is detected by a $KILL$ signal which is calculated in each cell by a built-in self-test mechanism realized at the molecular level (see Subsection 2.4 below). The state $KILL = 1$ identifies the faulty cell, and the entire column to which the faulty cell belongs is considered faulty and is deactivated (column $X = 2$ in Figure 5). All the functions (X coordinate and gene) of the cells to the right of the column $X = 1$ are shifted by one column to the right. Obviously, this process requires as many spare columns to the right of the array as there are faulty cells or columns to repair (two spare columns, tolerating two successive faulty cells, in the example of Figure 5). It also implies that the cell needs to be able to bypass the faulty column and to divert to the right all the required signals (such as the operative genome and the X coordinate, as well as the data busses).

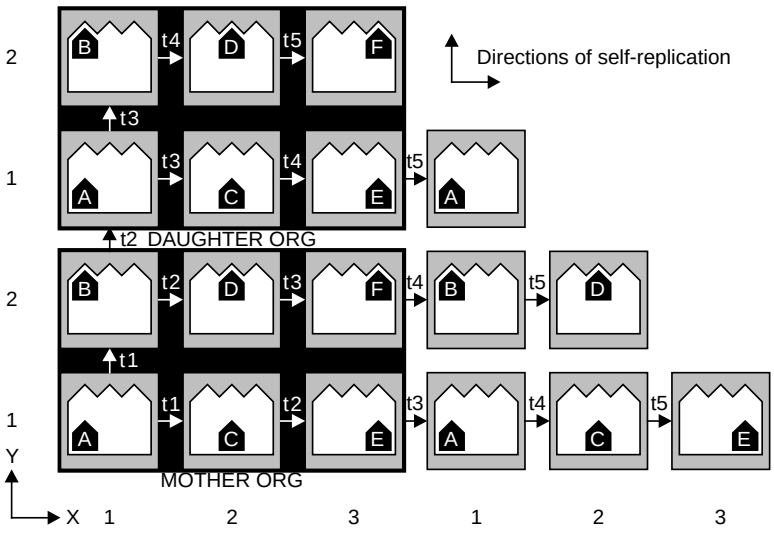

Fig. 4. Self-replication of a 6-cell organism *ORG* in a limited homogeneous array of 6×4 cells (situation at time $t5$ after 5 cellular divisions); *MOTHER ORG* = mother organism; *DAUGHTER ORG* = daughter organism.

Given a sufficient number of cells, it is obviously possible to combine self-repair in the X direction, and self-replication in both the X and Y directions.

2.4 The cell's features: multimolecular organization, molecular configuration, and molecular fault detection

In each cell of every living being, the genome is translated sequentially by a chemical processor, the *ribosome*, to create the proteins needed for the organism's survival. The ribosome itself consists of molecules, whose description is an important part of the genome.

As mentioned, in the Embryonics project each cell is a small processor, sequentially executing the instructions of a first part of the artificial genome, the operative genome *OG*. The need to realize organisms of varying degrees of complexity has led us to design an artificial cell characterized by a flexible architecture, that is, itself configurable. It will therefore be implemented using a new kind of fine-grained, field-programmable gate array (FPGA).

Each element of this FPGA (consisting essentially of a multiplexer associated with a programmable connection network) is then equivalent to a *molecule*, and an appropriate number of these artificial molecules allows us to realize application-specific processors. We will call *multimolecular organization* the use of many molecules to realize one cell. The configuration string of the FPGA (that is, the information required to assign the logic function of each molecule) constitutes the second part of our artificial genome: the *ribosomic genome RG*. Fig-

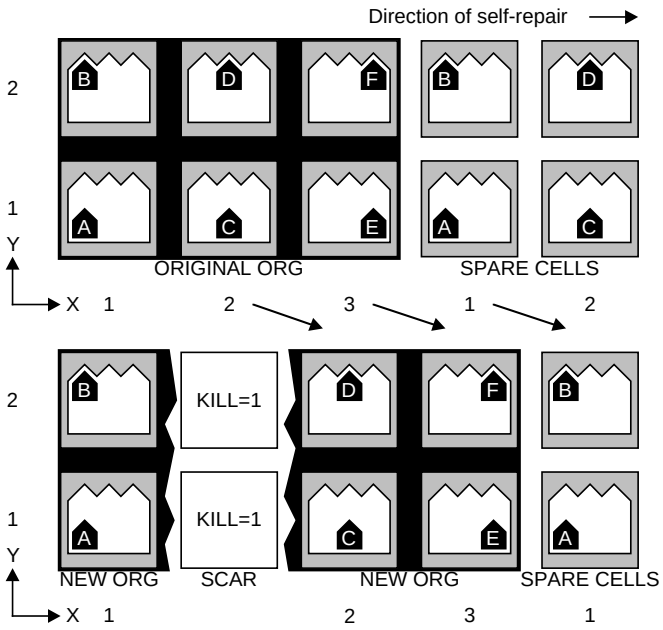

Fig. 5. Organismic self-repair.

ure 6a shows a generic and abstract example of an extremely simple cell ($CELL$) consisting of six molecules, each defined by a *molecular code* or $MOLCODE$ (a to f). The set of these six $MOLCODE$s constitutes the ribosomic genome RG of the cell.

The information contained in the ribosomic genome RG thus defines the logic function of each molecule by assigning a molecular code $MOLCODE$ to it. To obtain a functional cell, we require two additional pieces of information.

1. The *physical* position of each molecule in the cellular space.
2. The presence of one or more *spare columns*, composed of *spare molecules*, required for the self-repair described below (Subsection 2.5).

The definition of these pieces of information is the molecular configuration (Figure 6b). Their injection into the FPGA will allow:

- the creation of a border surrounding the molecules of a given cell;
- the insertion of one or more spare columns;
- the definition of the connections between the molecules, required for the propagation of the ribosomic genome RG.

The information needed for the molecular configuration (essentially, the height and width of the cell in number of molecules and the position of the spare

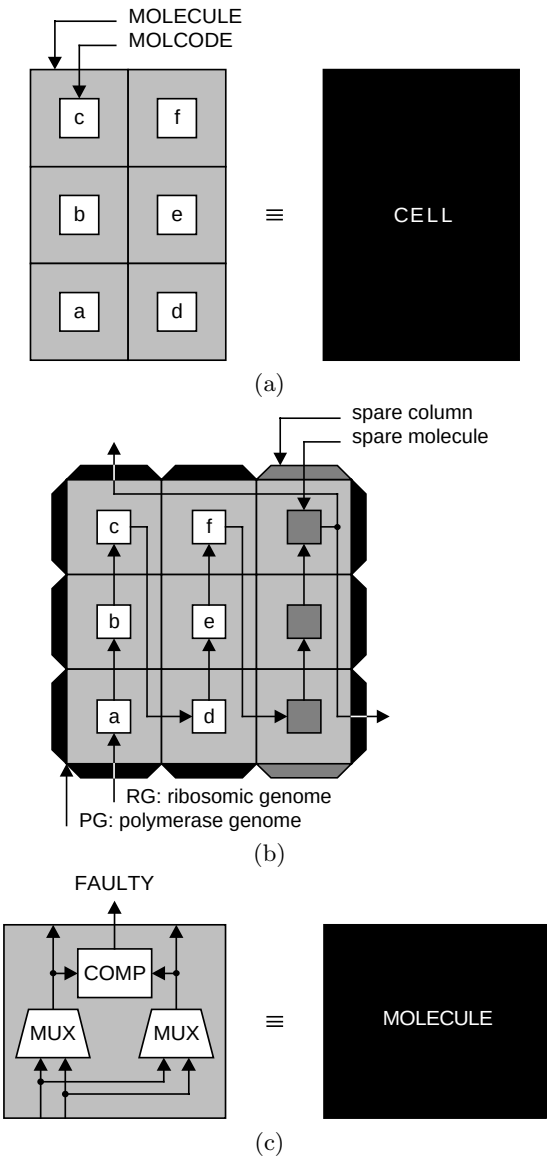

Fig. 6. The cell's features. (a) Multimolecular organization; RG: ribosomic genome: a, b, c, d, e, f. (b) Molecular configuration; PG: polymerase genome: height $\times$ width $= 3 \times 3$; 001 = spare column. (c) Molecular fault detection; MUX: multiplexer; $COMP$: comparator.

columns) makes up the third and last part of our artificial genome: the *polymerase genome* PG [2].

Finally, it is imperative to be able to automatically detect the presence of faults at the molecular level and to relay this information to the cellular level. Moreover, if we consider that the death of a column of cells is quite expensive in terms of wasted resources, the ability to repair at least some of these faults at the molecular level (that is, without invoking the organismic self-repair mechanism) becomes highly desirable. The biological inspiration for this process derives from the DNA's double helix, the physical support of natural genomes, which provides complete redundancy of the genomic information though the presence of complementary bases in the opposing branches of the helix. By duplicating the material of each molecule (essentially the multiplexer MUX) and by continuously comparing the signals produced by each of the two copies (Figure 6c), it is possible to detect a faulty molecule and to generate a signal $FAULTY = 1$, realizing the *molecular fault detection* which will make possible cellular self-repair (described below in Subsection 2.5).

2.5 Cellular self-repair

The presence of spare columns, defined by the molecular configuration, and the automatic detection of faulty molecules (Subsection 2.4, Figure 6b and c) allow cellular self-repair: each faulty molecule is deactivated, isolated from the network, and replaced by a neighboring molecule, which will itself be replaced by a neighbor, and so on until a spare molecule is reached (Figure 7).

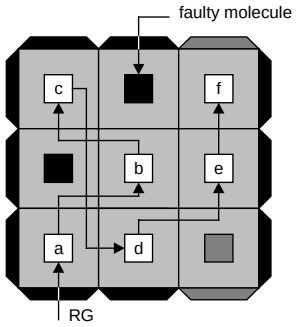

Fig. 7. Cellular self-repair.

The number of faulty molecules handled by the molecular self-repair mechanism is necessarily limited: in the example of Figure 7, we tolerate at most one faulty molecule per row. If more than one molecule is faulty in one or more rows, molecular self-repair is impossible, in which case a global signal $KILL = 1$ is generated to activate the organismic self-repair described above (Subsection 2.3 and Figure 5).

2.6 The Embryonics landscape

The final architecture of the Embryonic project is based on four hierarchical levels of organization which, described from the bottom up, are the following (Figure 8).

1. The basic primitive of our system is the *molecule*, the element of our new FPGA, consisting essentially of a multiplexer associated with a programmable connection network. The multiplexer is duplicated to allow the detection of faults. The logic function of each molecule is defined by its molecular code or $MOLCODE$.
2. A finite set of molecules makes up a *cell*, essentially a processor with the associated memory. In a first programming step of the FPGA, the polymerase genome PG defines the topology of the cell, that is, its width, height, and the presence and positions of columns of spare molecules. In a second step, the ribosomic genome RG defines the logic function of each molecule by assigning its molecular code or $MOLCODE$.
3. A finite set of cells makes up an *organism*, an application-specific multiprocessor system. In a third and last programming step, the operative genome OG is copied into the memory of each cell to define the particular application executed by the organism (electronic watch, random number generator, and a Turing machine being examples we have shown to date) [2].
4. The organism can itself self-replicate, giving rise to a *population* of identical organisms, the highest level of our hierarchy.

3 Conclusion

3.1 Toward a new reconfigurable computing tissue: the e–wall

Keeping in mind that our final objective is the development of very large scale integrated (VLSI) circuits capable of self-repair and self-replication, as a first step, which is the subject of this paper, we have shown that a hierarchical organization based on four levels (molecule, cell, organism, population of organisms) allows us to confront the complexity of real systems. The realization of demonstration modules at the cellular level and at the molecular level [2] demonstrates that our approach can satisfy the requirements of highly diverse artificial organisms and attain the two sought-after properties of self-repair and self-replication.

The programmable robustness of our system depends on a redundancy (spare molecules and cells) which is itself programmable. This feature is one of the main original contributions of the Embryonics project. It becomes thus possible to program (or re-program) a greater number of spare molecules and spare cells for operation in hostile environments (e.g., space exploration). A detailed mathematical analysis of the reliability of our systems is currently under way at the University of York [5], [6].

In our laboratory, the next major step in the Embryonics project is the design of the BioWatch, a complex machine which we hope to present on the occasion

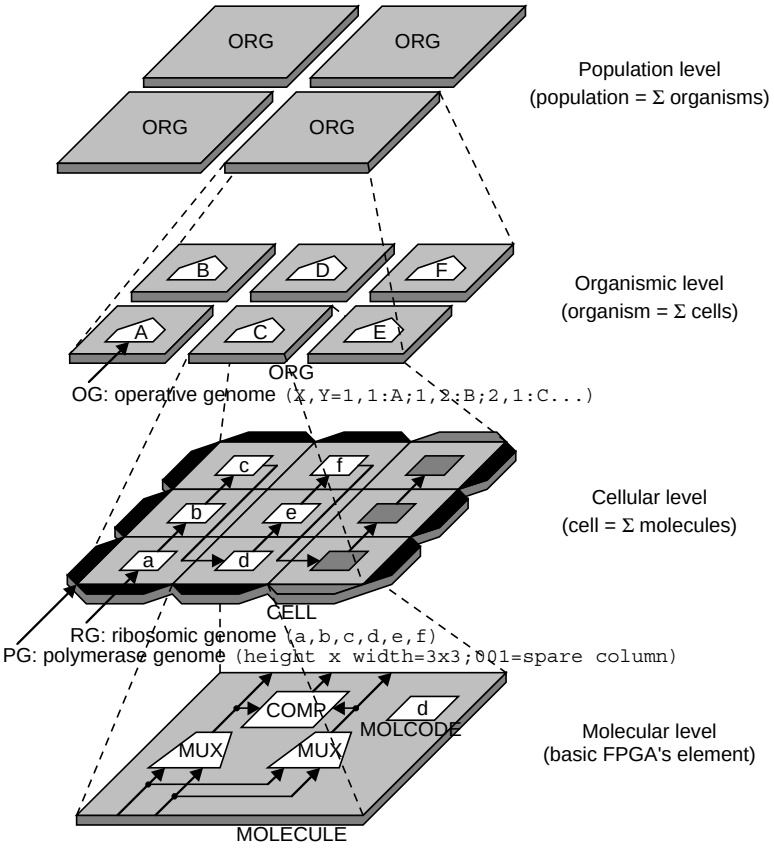

Fig. 8. The Embryonics landscape: a 4-level hierarchy.

of a cultural event which will soon take place in Switzerland. The function of the machine will be that of a self-repairing watch, counting seconds, minutes and hours. The implementation of the watch will take place in a *reconfigurable tissue*, the *electronic wall* or *e–wall* (Figure 9).

Conception, birth, growth, maturity, illness, old age, death: this is the life cycle of living beings. The proposed demonstration will stage the life cycle of the BioWatch from conception to death. Visitors will face a large wall made up of a mosaic of many thousands of transparent electronic modules, each containing a display. At rest, all the modules will be dark. A complex set of signals will then start to propagate through the space (conception) and program the modules to realize the construction of a beating electronic watch (growth). Visitors will then be invited to attempt to disable the watch: on each molecule, a push-button will allow the insertion of a fault within the module (wounding). The watch will automatically repair after each aggression (cicatrization). When the number of

Fig. 9. The BioWatch built in a new reconfigurable tissue: the e–wall (Computer graphic by E. Petraglio).

faults exceeds a critical value, the watch dies, the wall plunges once more into darkness, and the complete life cycle begins anew.

We are physically implementing the 6-digit BioWatch in our two-dimensional e–wall made up of 3200 modules. This wall is ultimately intended as a universal self-repairing and self-replicating display medium, capable of interacting "intelligently" with the user and recovering from faults. In our implementation, each molecule of the watch corresponds to a module in the wall. This module includes: (1) an input device, (2) a digital circuit, and (3) an output display [3].

The module's outer surface consists of a touch-sensitive panel which acts like a digital switch, enabling the user to kill a molecule and thereby activate the self-repair process at the molecular level (signal $FAULTY = 1$ in Figure 6c).

The module's internal circuit is a field-programmable gate array (FPGA), configured so as to realize: (1) the acknowledgment of the external (touch) input, (2) the implementation of a molecule (multiplexer with a programmable connection network), and (3) the control of the output display. This latter is a two color light-emitting diode (LED) display, made up of 64 diodes arranged as an 8×8 dot-matrix. The display allows the user to view the current molecule's state: active, faulty, spare, etc.

The configuration string of the FPGA can be considered as an ultimate part of our artificial genome, necessitated by the construction of the molecular level (Figure 8) on the shoulders of a new, lowest level: the atomic level.

3.2 Toward POEtic machines

We presented the POE model, classifying bio-inspired hardware systems along three axes (phylogeny, ontogeny, and epigenesis), and we described our Embryonics project along the ontogenetic axis. A natural extension which suggests itself is the combination of two and ultimately all three axes, in order to attain novel bio-inspired hardware. An example of the latter would be an artificial neural network (epigenetic axis), implemented on a self-replicating and self-repairing multicellular automaton (ontogenetic axis), whose genome is subject to evolution (phylogenetic axis).

Looking (and dreaming) toward the future, one can imagine nano-scale systems becoming a reality, endowed with evolutionary, reproductive, regenerative, and learning capabilities. Such systems could give rise to novel species which will coexist alongside carbon-based organisms. This constitutes, perhaps, our ultimate challenge.

Acknowledgments

This work was supported in part by the Swiss National Foundation under grant 21-54113.98, by the Leenaards Foundation, Lausanne, Switzerland, and by the Villa Reuge, Ste-Croix, Switzerland.

References

1. D. Mange, M. Sipper, and P. Marchal. Embryonic electronics. *BioSystems*, 51(3):145–152, 1999.
2. D. Mange, M. Sipper, A. Stauffer, and G. Tempesti. Toward robust integrated circuits: The embryonics approach. *Proceedings of the IEEE*, 88(4):516–541, April 2000.
3. D. Mange, A. Stauffer, G. Tempesti, and C. Teuscher. Tissu électronique reconfigurable, homogène, modulaire, infiniment extensible, à affichage électro-optique et organes d'entrée, commandé par des dispositifs logiques reprogrammables distribués. Patent pending, 2001.
4. D. Mange and M. Tomassini, editors. *Bio-Inspired Computing Machines*. Presses polytechniques et universitaires romandes, Lausanne, 1998.
5. C. Ortega and A. Tyrrell. Reliability analysis in self-repairing embryonic systems. In A. Stoica, D. Keymeulen, and J. Lohn, editors, *Proceedings of The First NASA/DOD Workshop on Evolvable Hardware*, pages 120–128, Pasadena, CA, 1999. IEEE Computer Society.
6. C. Ortega and A. Tyrrell. Self-repairing multicellular hardware: A reliability analysis. In D. Floreano, J.-D. Nicoud, and F. Mondada, editors, *Advances in Artificial Life*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 1999.
7. M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Pérez-Uribe, 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.
8. L. Wolpert. *The Triumph of the Embryo*. Oxford University Press, New York, 1991.