26th Euromicro Conference, Workshop on Digital Systems Design, pp. 356-361.
Maastricht, Netherlands, September 2000.

Reliability Analysis of a Self-Repairing Embryonic Machine

César Ortega[1], Daniel Mange[2], André Stauffer[2], Gianluca Tempesti[2], Andy Tyrrell[1]

[1]Department of Electronics
University of York
York, YO10 5DD, UK
cesar@ohm.york.ac.uk

[2]Logic Systems Laboratory
Swiss Federal Institute of Technology
CH-1015, Lausanne, Switzerland
tempesti@di.epfl.ch

Abstract. The embryonics project proposes a family of cellular architectures with reconfiguration properties inspired by the ontogenesis of multicellular organisms. This paper proposes reliability models for the MICTREE embryonic architecture. The models are used to analyse the reliability of MICTREE arrays with different combinations of spare cells. The methodology followed to attain the models can be used to analyse other cellular architectures with spares.

1 Introduction

The systematic study of artificial cellular systems, like cellular automata, neural networks or processor arrays has gained momentum during the past few years in the arena of evolvable systems. The goal is to understand the emergent behaviours observed in natural cellular systems. To borrow the main principles sustaining these mechanisms and apply them to the design of electronic systems could result in a new approach to fault tolerance [1]. Embryonics has been proposed as a bio-inspired cellular architecture that relies on hardware redundancy (i.e. faulty cells are replaced by spare cells) to achieve fault tolerance [2].

This paper proposes reliability models for the MICTREE embryonic architecture. Section 2 presents a brief introduction to the embryonics project and describes the MICTREE architecture. In section 3, the reliability models for MICTREE organisms are derived. The reliability models are used to analyse embryonic systems with different combinations of spare cells and organisms. Conclusions and proposals for future work are presented in section 4.

2 Embryonics

At the core of the Embryonics project [3, 4] is the idea of splitting a complex application-specific system (a computing machine dedicated to the execution of an arbitrarily complex task, our equivalent of a biological organism) into small processing elements (simple processors implemented as binary decision machines, our equivalent of biological cells), drawing inspiration from the multicellular organisation of living beings (Figure 1).

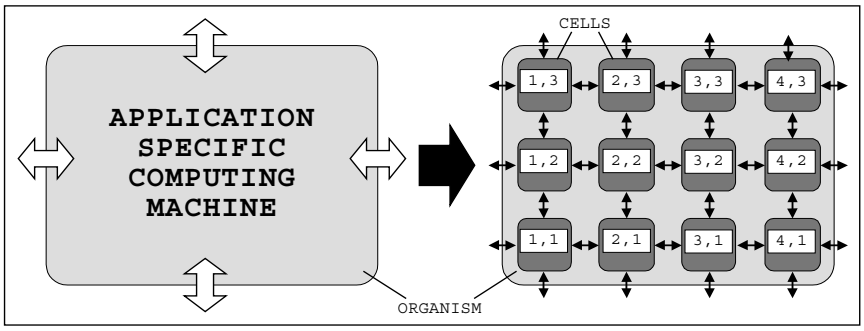

Fig. 1 Multi-cellular organisation of an embryonic system

Obviously, this paradigm is nothing new, and indeed does not in itself owe anything to biological inspiration: it is the basis of all parallel processing in conventional computing systems. To avoid some of the pitfalls inherent in the development and operation of numerous processing elements in parallel, however, we introduced some unique features derived directly from the world of biology. Living beings are, after all, astounding examples of parallel machines executing hugely complex tasks from the parallel operation of myriad simple elements, the cells.

In particular, to define an approach for partitioning the task to be executed into smaller fragments, the biological process of ontogenesis has been investigated [5,6]. Ontogenesis is the mechanism that determines the development of an organism from a single mother cell (the *zygote*) to a full-blown adult. The zygote divides, each offspring containing a copy of the genome (*cellular division*). This process continues (each new cell divides, creating new offspring), and each newly formed cell acquires a given functionality (i.e. liver cell, epidermal cell, etc.) depending on its surroundings, i.e. its position in relation to its neighbours (*cellular differentiation*).

The limitations of current technology do not allow to accurately model this process in artificial organisms: the transistors of today's digital circuits cannot yet be "assembled" from their constituent parts as easily as biological cells. However, current technology does allow designing a system that follows the same basic approach.

The solution consists of implementing organisms as two-dimensional arrays of identical processing elements or artificial cells (labelled MICTREE for "tree of microinstructions"). Each cell stores the same program, but executes different subsets of this program depending on its position within the array (in the same way as a biological cell determines its function depending on its spatial location within the developing organism). In the embryonics approach, the cells are physically present in the array at all times, but are functionally inert until they are provided with a program and with a set of co-ordinates. The program then becomes an artificial equivalent of the biological genome, whose presence in each cell is the basis for the mechanism of ontogenesis. The execution of different subsets of this program in each processor becomes the equivalent of cellular differentiation.

The architecture of embryonic artificial cells follows directly from this approach (Figure 2). Each cell must contain a memory to store the genome and an [X,Y] co-ordinate system to allow the cell to find its position within the array and thus its

function. In addition, it requires an interpreter to read and execute the genome, a set of connections handled by a routing unit to exchange information with its neighbours, and a functional unit to execute the given task.

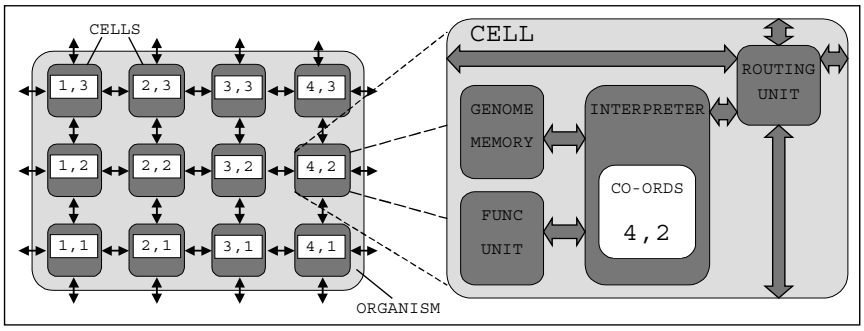

Fig. 2 The architecture of the artificial cells

This approach, not very efficient from the standpoint of conventional circuit design (storing a complete copy of the genome program in each processor is redundant), allows easy exploitation of the most interesting features of biological systems. For example, since the function of a cell depends on its co-ordinates, by allowing the co-ordinates to cycle (Figure 3) it is possible to obtain multiple copies of an organism with a single copy of the program. This process achieves the self-replication, or cloning, of the machine (provided, of course, that the array contains a sufficient number of processors).

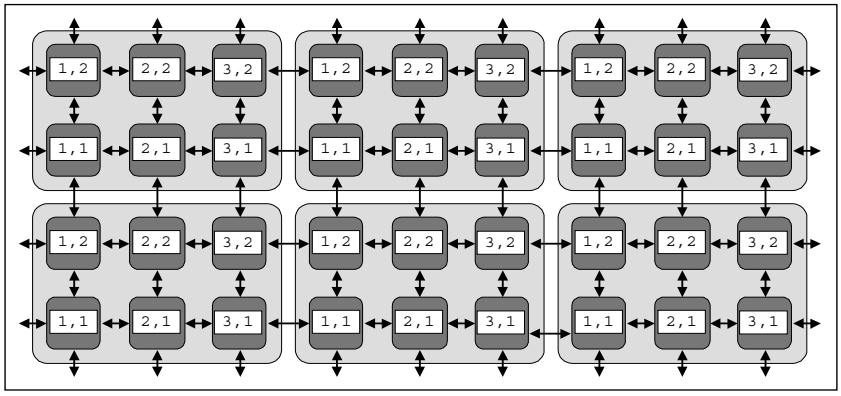

Fig. 3 Automatic replication of an organism (cloning)

More importantly for the subject of this article, the presence of the genome in each cell enormously simplifies the process of self-repair: if a hardware fault should "kill" one or more of the processors, a simple recomputation of the co-ordinates of the array (Figure 4) allows the "dead" processors to be replaced (provided, of course, that a set of “spare” cells is available). This process owes its simplicity to the architecture of the system. When a fault is detected within a cell, the column containing the cell "vanishes" from the array, i.e., it becomes transparent with respect to all horizontal connections (note that the reconfiguration of the array is limited to entire columns of

cells, a trade-off between efficiency and simplicity). The column to the right of the dead column detects its disappearance (its neighbours have different co-ordinates) and recomputes its own co-ordinates, thus starting a cascade effect which will end when the first spare column is reached.

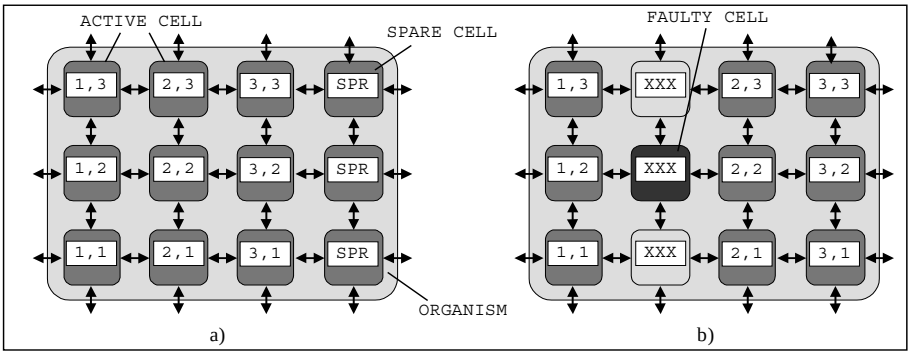

Fig. 4. Reconfiguration of an organism around a faulty cell

While this approach does not exploit a very efficient reconfiguration scheme (sacrificing an entire column for a single faulty cell) [7], it has the compensating advantage of being extremely simple, an almost trivial consequence of the architecture of our systems.

3 Reliability of the MICTREE architecture

To analyse the reliability of MICTREE organisms, the following notation has been used. There are $\boldsymbol{h} \times \boldsymbol{g}$ cells in an organism, from which only a sub-array of $\boldsymbol{h} \times \boldsymbol{f}$ cells will perform the desired function. The remaining $\boldsymbol{g} - \boldsymbol{f}$ columns are spares that replace faulty columns when a fault is detected in a cell. The user defines the number of spare columns that are inserted in an organism to provide a higher level of fault tolerance. Figure 5 shows the structure of a MICTREE organism.

In the following analysis, λ is the failure rate of a single cell. λ is assumed constant and usually expressed as failures per unit time, for example failures per hour or failures per 10^6 hours. The failure rate is determined by the cell's model. The cell model is a function of parameters that describe its physical implementation, operating characteristics and the environment in which the device operates [8].

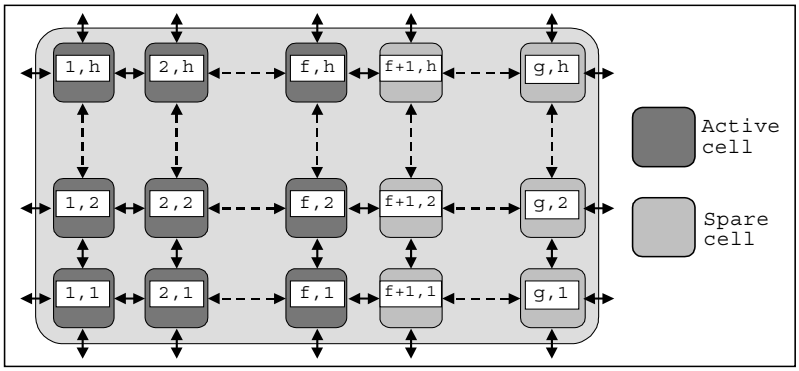

Fig. 5 Internal structure of a MICTREE organism

As mentioned, when a fault occurs in one of the cells within the organism, the corresponding column is logically eliminated by making it transparent to the calculation of co-ordinates. Co-ordinates are shifted to the right until a spare column is reached. At that point, cells will perform a new function according to their new set of co-ordinates. An organism will "live" as long as there are spare columns that can replace transparent ones. When spare columns run out and a new fault arises, the organism "dies", i.e. it stops delivering its service

The reliability of one row of cells can be expressed using the ***f***-out-of-***g*** reliability model. In general, an ***f***-out-of-***g*** system will remain functional as long as ***f*** out of its ***g*** components remain functional. The reliability of an ***f***-out-of-***g*** system with identical elements is given by the following expression [9],

$$R_{f-out-of-g}(t) = \sum_{i=f}^{g} \binom{g}{i} e^{-i\lambda t} \left(1 - e^{-\lambda t}\right)^{g-i} \quad (1)$$

Where the reliability of one cell is assumed to follow the exponential law given by,

$$R(t) = e^{-\lambda t} \quad (2)$$

Expression (1) yields the reliability of one row in the array. The organism's reliability can be expressed as the series connection of ***h*** rows. A system with ***h*** elements connected in series requires all its elements working correctly in order to deliver its function. The reliability of a series system with independent elements is given by the multiplication of the reliability expressions of all its elements [9]. Therefore, the reliability of one organism with ***h*** rows would be given by,

$$R_{org}(t) = \left(\sum_{i=f}^{g} \binom{g}{i} e^{-i\lambda t} \left(1 - e^{-\lambda t}\right)^{g-i} \right)^{h} \quad (3)$$

If the number of cells in the array allows cloning of the original organism, then every clone will perform the same function. Therefore, a correct output will be

available as long as there is one surviving organism. A structure like this is modelled by the reliability expression for ***x*** identical elements connected in parallel [9].

$$R_{total}(t) = 1 - \left(1 - R_{org}(t)\right)^{x} \quad (\mathbf{4})$$

Where ***x*** is the number of organisms in the system.

Figure 6a shows the graphic representation of equation (3) for organisms with different number of spare columns. Figure 6b shows the corresponding representation of equation (4). The following conditions have been assumed:

- Organism's size (***h***×***g***) is 50×50 cells
- Number of organisms in the system (***x***) is 6. The topology (i.e. 3×2, 6×1, etc.) is not relevant.
- Failure rate (λ) is 1 failure every 10^6 hours
- Cells fail independently from each other

For the purpose of comparison, the reliability of a 50 × 50 cells array with no spare columns is also shown. The benefits of adding redundancy are evident.

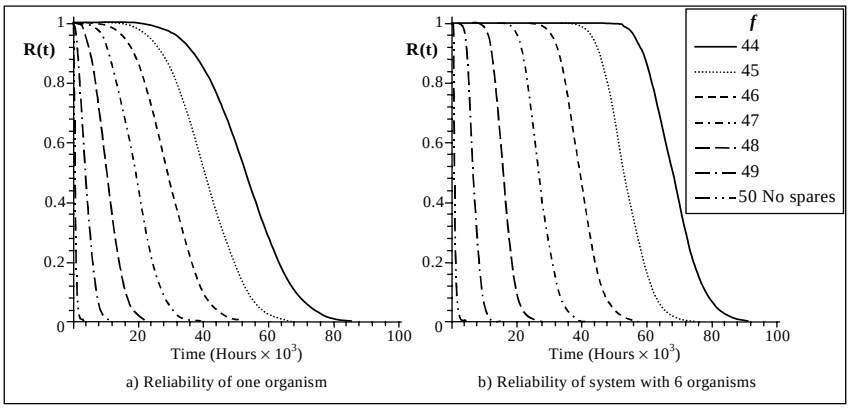

Fig. 6 Reliability of 50×50 MICTREE organism with different number of spare columns

A one-to-one comparison of the graphs in figures 6a and 6b shows the effects of cloning. The more organisms there are the better the reliability of the system. Figure 6 clearly shows that the larger the number of spare columns in the organism, the longer the cell will function correctly. It also shows that the relative improvement in reliability with respect to the number of spare cells increases as the number of spare cells increases. For example, the improvement in reliability when passing from 5 to 6 spare columns is bigger than the improvement when passing from 2 to 3 spares.

Figures 7a and 7b show the graphs of a system with the following characteristics:

- Organism's size (***h***×***g***) is 50×50 cells
- Number of organisms in the system (***x***) is 6
- Number of active cells per row (***f***) is 44, i.e. 6 spares per row
- Different failure rates (λ)
- Cells fail independently from each other

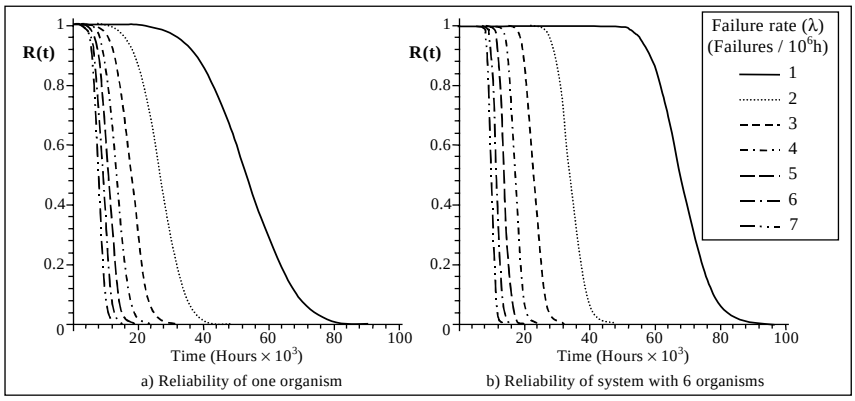

Fig. 7 Reliability of MICTREE system for different failure rates

Figure 7 shows the high reliability associated to small values of λ. Decreasing the value of λ requires an improvement in the quality of the system's components and in the majority of cases, the cost associated with this is too high. However, in highly reliable systems like satellites or specialised control equipment, the cost of decreasing the value of λ is amply justified.

Graphs in figure 8 show the behaviour of system reliability for organisms with different number of rows. All the systems share the following characteristics:

- Organism's width (***g***) is 50 cells
- Organism's depth (***h***) is variable
- Number of active cells per row (***f***) is 44, i.e. 6 spares per row
- Failure rate of cells (λ) is 1 failure every 10^6 years
- Cells fail independently from each other

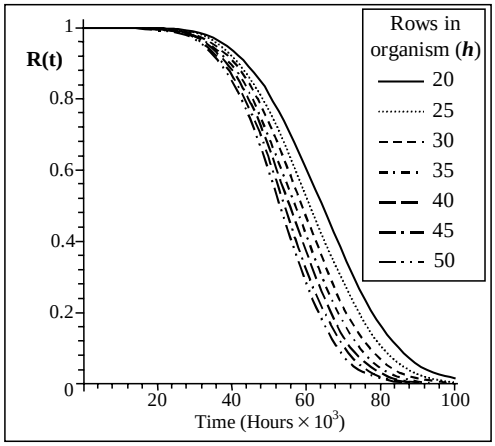

Fig. 8 Reliability of organisms with different number of rows

Figure 8 shows that the reliability of organisms decreases as the number of rows increases. This result is expected from any system whose elements are connected in series. Therefore, when an application is being mapped to a MICTREE array, the number of rows per organism should be minimised. The ideal case would be an organism with only one row, with its reliability given by expression (1).

Figure 9 shows reliability graphs of seven embryonic systems with different number of MICTREE organisms. The arrangement of organisms is irrelevant; as long as one organism survives, the system will deliver its function. The common characteristics of the systems are:

- Organism's size (***g***×***h***) is 50×50 cells
- Number of active cells per row (***f***) is 44, i.e. 6 spares per row
- Failure rate of cells (λ) is 1 failure every 10^6 years
- Cells fail independently from each other

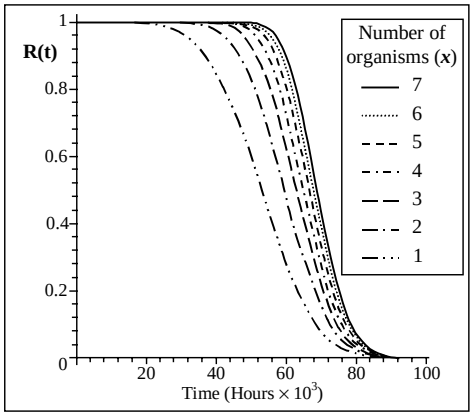

Fig. 9 Reliability of embryonic systems with different number of MICTREE organisms

Figure 9 shows that as the number of organisms increases, the reliability curve becomes steeper, i.e. reliability improves. However, figure 9 also demonstrates that the contribution from new organisms to system reliability decreases as the total number of organisms increases. For example, reliability improves more when passing from 2 to 3 organisms than when passing from 6 to 7. Hence, it is important to consider a cost/benefit analysis before deciding the number of clone organisms allocated in a system.

4 Conclusions

Embryonic arrays exploit hardware redundancy to achieve fault tolerance. The distributed automatic reconfigurability characteristic of embryonic arrays offers considerable advantages over other reconfiguration strategies where, in most cases, a centralised agent, e.g. operating system or central processor, must solve the routing of information problem. For reliability analysis purposes the central agent is connected in series to the array, *i.e.* both must perform their functions correctly in order to

consider the whole system in working order. But the reliability of a series system will always be lower than that of the element with the lowest reliability. Hence, the centralised approach should be avoided for the design of dependable applications.

Diagnosis and reconfiguration functions are performed at cellular level in the embryonics architecture. No centralised agent exists. Spare elements are incorporated at different levels of an embryonic system in order to achieve resilience to faults in their constituent cells and organisms. The reliability models presented in this work allow the analysis of the MICTREE architecture for different combinations of spare cells and organisms.

It has been verified that the best way, in terms of reliability, of colonising a given array of cells is to allocate active cells column-wise. In this way, the number of spare columns within each organism can be maximised, with the corresponding improvement in reliability.

If possible, clones must be considered when mapping an application into an embryonic array. The reliability models presented in this paper have shown that, in a system with clones, reliability improves proportionally to the number of organisms. However, as the number of organisms increases, the contribution to system reliability decreases. Therefore, a cost/benefit analysis must be carried out to determine the optimum number of organisms for a given application.

Further research must carried out in order to determine to what extent the models proposed hold for any fault-tolerant cellular system with spares.

5 Acknowledgements

This work has been supported by the following sponsors: Mexico's Government under grants CONACYT-111183 and IIE-9611310226, the Swiss National Foundation under grant 21-54113.98, the Consorzio Ferrara Richerche, Università di Ferrara, Italy, the Leenaards Foundation, Switzerland and the British Council, UK.

6 References

1. Avizienis A., "Toward Systematic Design of Fault-Tolerant Systems", IEEE Computer, April 1997, Computer Society Press, pp. 51-58
2. Ortega C. and Tyrrell A., "Biologically Inspired Fault-Tolerant Architectures for Real-Time Control Applications", Control Engineering Practice, July 1999, pp. 673-678
3. Mange D. and Tomassini M. (Eds.), Bio-Inspired Computing Machines, Presses Polytechniques et Universitaires Romandes, Switzerland, 1998
4. G. Tempesti, D. Mange, A. Stauffer. "Self-Replicating and Self-Repairing Multicellular Automata". Artificial Life, 4(3), 1998, pp. 259-282.
5. S. F. Gilbert. Developmental Biology. Sinauer Inc., Massachusetts, 3rd edition, 1991.
6. L. Wolpert. The Triumph of the Embryo. Oxford University Press, New York, 1991.
7. R. Negrini, M.G. Sami, R. Stefanelli. Fault Tolerance Through Reconfiguration in VLSI and WSI Arrays. The MIT Press, Cambridge, MA, 1989.
8. Bowles J., "A Survey of Reliability-Prediction Procedures For Microelectronic Devices", IEEE Trans. on Reliability, Vol.41-1, March, 1992, pp.2-12
9. Misra K., Reliability Analysis and Prediction, Elsevier, 1992