A Functional Model Of Development and Expression in an Artificial Organism

Alessandro Fontana

1IEEE
alessandro.fontana@ieee.org

May 20, 2008

Abstract

This paper is placed at the intersection-point between the study of theoretical computational models aimed at capturing the essence of genetic regulatory networks and the field of Artificial Embryology (or Computational Development). A model is proposed, with the objective of providing an effective way to generate arbitrary forms by using evolutionary-developmental techniques. Preliminary experiments have been performed.

1 Related Work

One of the oldest computational models of development is perhaps represented by L-Systems, proposed by Lindenmayer ([Lindenmayer 1968]). L-Systems capture the notion of cell signalling and were proposed to model plant growth. Rewriting or Production Systems (see [Dittrich et al. 2001]) consist of certain entities or symbols (“chemical compounds”) and a set of syntactic rules for performing replacements (“chemical reactions”). Examples of P-systems are CHAM (Chemical Abstract Machine) and ARMS (Abstract Rewriting Systems on Multisets). Random Boolean Networks (RBN’s) were originally developed by S. Kauffman as a model of genetic regulatory networks ([Kauffman 1969]). In the context of the development of multi-cellular organisms, the attractors of RBN’s (i.e. the stable states) are interpreted as the different “cell types” of the organism. Torsten Reil ([Reil 1999]) developed a model called “artificial genome” with biologically-plausible properties and studied the dynamics of gene expression. The model differs from other approaches, such as RBN’s, in that it is entirely based on template-matching in a nucleotide-like sequence. Hugo de Garis ([De Garis 1999]) developed a model for evolving convex and non-convex shapes in 2D reproductive cellular automata. Two genes, which were called operons were used to regulate the growth of each of the arms of the shape. The model was successful in evolving convex shapes but the non-convex shapes (L-shape) presented a problem. Miller ([Miller 2003]) developed artificial organisms (the french flag) based on Cartesian Genetic Programming which is an extension of Boolean networks. Miller’s goal is to evolve a developmental program inside a cell to create multicellular organisms.

2 Introduction and Cell Variables

We represented the phenotype as a 2-dimensional grid, each pixel representing a single cell, the set of all cells being the organism. Cell’s variables are broken down into four categories: spacetime memory, genetic memory, epigenetic memory, temporary memory or substance space (SUSP): figure 1 lists all variables. Through the interplay of these variables, at each cycle a given cell can proliferate, can undergo apoptosis, and can express its phenotype (represented by a color).

3 The Cell Cycle, Development Phase

At each BC value, each cell of the organism carries out the cell cycle, which is divided in two phases: development phase and expression phase. (see figure 2). During the development phase, the cell variables BC (body clock) and
TM (type marker) are fed to DGM to yield the developmental behaviour of the cell. The cell can either proliferate in a rectangle containing the (mother) cell and whose coordinates are specified in DGM, undergo apoptosis (i.e. disappear from the grid) or do nothing. If the cell “chooses” to proliferate, it can do it in two distinct “modes”: growth mode and differentiation mode. In the growth mode the newly created cells are assigned the TM value of the mother; in the differentiation mode each newly created cell is assigned a new value of TM (and TM is incremented by one at each new assignment). At the beginning a single cell is put in the grid with TM=0 (zygote). DGM (see figure 3) is structured as a list of “development operators”. Each operator has a BC value and a TM value: these are the values for which the operator gets activated. Another field holds a “master switch” that defines the type of event that is going to occur: growth-mode proliferation (value=0), differentiation-mode proliferation (value=1), apoptosis (value=2). The last field contains the coordinates of the north-west and south-east vertices of the proliferation rectangle.

The different types of development operators correspond to a painter’s tools: the rectangle proliferation corresponds to the brush, the apoptosis corresponds to the eraser, the differentiation mode serves to “mark” individual cells (through the assignment of different TM values) for further processing in subsequent stages. The presence of an explicit epigenetic memory, which enables the “cell tracking”, constitutes the most relevant feature of our method.

4 The Cell Cycle, Expression Phase

During the expression phase, the substance space SUSP is processed through EGM and the result is used to determine the cell phenotype (represented as a color). EGM is structured as an array of expression operators and SUSP is structured as an array of operands (artificial counterpart of biological chemical substances), that can be broken down into three categories:

- operands that are exchanged with other cells (input operands)
- operands that determine the phenotype (output operands)
- operands that are neither inputs nor outputs (hidden operands)

At the beginning of the expression phase, the input part of SUSP is initialized (the SUSPs of surrounding cells are mapped onto it). Then each operator of EGM “binds” its operands in SUSP and carries out its operation, updating SUSP as a result. This sequence of operations is repeated n times (n is a parameter). At the end the phenotype of the cell (color) is determined from the value of the output operands. An operand (opd) has two fields, one (opd.lab) holding the operand’s “label” (corresponding to the chemical “type”), the other (opd.val) holding the operand’s “value” (corresponding to the chemical concentration). The structure of the operator (oxr) is more complicated. The first field (oxr.inputs) is composed of n (n is the arity of the operator) sub-fields, each corresponding to a different input (oxr.inputs[i]). Each sub-field is composed of an operand (inputs[i].opd) and two parameters to process that operand, one for each operation’s step. The second field is composed of two (output) labels, oxr.outlab0
5 Biological Interpretation and Motivations of the Model

DGM and EGM correspond to the chromosome(s), and the development and expression operators correspond to the individual genes. The epigenetic memory corresponds to the biological epigenetic memory, implemented in real cells through the methylation of genes that inhibits their transcription. The operands in the substance space correspond to RNA pieces and proteins participating in regulatory pathways during the interphase. The possibility of choosing on-the-fly between two labels for the output operand is equivalent to the conditional branching feature and has its biological counterpart in the alternative splicing mechanism, i.e. the possibility that more proteins can be read out
from a single gene. Our model’s cell cycle differs from the biological one in that in the development phase a cell can proliferate over a rectangular area, which in real cells can be achieved only through a series of mitoses, each belonging to a distinct cell cycle. The aspects we are interested to model are those related to the shaping of an organism, achieved in the living world through the mitosis, differentiation and apoptosis of individual cells. We decided to separate these functions into a development part, that provides the overall “shape” of the organism, and an expression part, that provides the phenotype of individual cells. The development part is modeled at a relatively high level (a single development operator can in theory make a cell cover the entire grid), in order to speed-up the “painting” process. The source of the different behaviour of a cell is provided in the development phase by the variable TM (type marker), and in the expression phase by the variable TM (that determines the activation pattern of the expression genes) and the behaviour of surrounding cells (that provides the initial state of SUSP). We believe that the surrounding environment alone is not sufficient to explain the different behavioural patterns observed in real biological systems.

6 Implementations

Preliminary experiments have been performed, for the moment only limited to the development part (creation of colorless shapes). These experiments were aimed at evaluating the capability of the model to “paint” in an effective way, i.e. to evolve shapes. The algorithm was able to successfully evolve some simple shapes, including non-convex ones. Further experiments are under way. These first attempts have shown the emergence of a “painting technique”, consisting in a fast differentiation in the first cycles, followed by touch-ups in the subsequent cycles. The genetic algorithm “understands” that the first thing to do is to create a relatively big mass of cells, marking each of them with a different type marker. This is the only means by which it can later pick individual cells and commit them to a specific fate. Examples of target shapes are reported below (in black), with the best evolved solution (in color).

References

De Garis, H. (1999). Artificial Embryology and Cellular Differentiation. Academic Press.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries: A review. Artificial Life, 7.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology, 22:437–467.
Lindenmayer, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–289.

Miller, J. F. (2003). Evolving developmental programs for adaptation, morphogenesis, and self-repair. *Lecture Notes in Computer Science*.

Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *ECAL99*.

Figure 5: The L and the beetle.

Figure 6: Development sequence of the plane.

Figure 7: The house.

Figure 8: The map of Italy.