ADAPTIVE SYSTEMS IN DESIGNING: NEW ANALOGIES FROM GENETICS AND DEVELOPMENTAL BIOLOGY

John S Gero
Key Centre of Design Computing
University of Sydney NSW 2006 Australia
email: john@arch.usyd.edu.au

Abstract. This paper introduces the notion that analogies other than Darwinian evolution may be used as the bases for useful designing processes. Two classes of analogies are described: those drawn from genetics and those drawn from developmental biology. Two extensions to the genetic analogy are described. The first utilises concepts from genetic engineering, which is the human intervention in natural genetics, and concepts from reverse engineering. The second utilises concepts from developmental biology which is concerned with how the organism develops in its environment once its genotype has been fixed.

1. Introduction

Designing has long been recognised as a difficult, complex and unusual task. The first mention of design goes back to the code of Hammurabi promulgated around 1950 BC. Science has formed the basis of the technology on which engineering sits. It has provided the necessary theory of material behaviour and the experimental methodology to determine such behaviour. Using theories of material behaviour it has been possible to develop formal methods of analysis of the behaviour of configurations of materials (ie designs) under a variety of environmental conditions. However, science has not had the same success in providing any foundation on which to base the technology of formal design methods. More recently, it has been suggested that designing in its fullest sense maps well onto abductive processes which helps explain why it is so difficult to formalise it. In addition to its abductive nature designing is situated: ie designing cannot be predicted since decisions to be taken depend on where the designer is at any particular time and what the designer perceives the situation to be when he is where he is. We will use the word “designing” to denote the act and the work “design” to denote the results of the act to avoid confusion.

Computational processes which support designing do not necessarily require any theoretical foundation and are usually restricted to some subset of the totality of the activities of human designing. This lack of a need for any theoretical foundation provides enormous flexibility on the source for computationally implementable ideas which may support designing as distinct from analysis.

Computational processes which support designing can be grouped into three categories:

(i) those founded on empirical evidence of human designing activity;
(ii) those founded on axioms and their derivations; and
(iii) those founded on conjectures of potentially useful processes.

This third category can be broken into two further subcategories:

(a) conjectures based on analogies with perceived human designing processes, and
(b) conjectures based on analogies with other processes (which are clearly not human designing processes).

Adaptive systems fall into this last subcategory. Human designers do not design with large populations or use any of the machinery of formal adaptive systems when they carry out the act of designing themselves.

What we intend to do in this paper is to briefly set the scene for the *genetic analogy* in designing and then to explore two possible research directions in adaptive systems for designing which may prove to be fruitful. The first extends the genetic analogy by introducing two further concepts, whilst the second draws its stimulus from development biology. Rather than provide detailed examples we will describe the ideas at the conceptual level.

2. The Genetic Analogy In Design

The basic genetic analogy in designing utilises a simple model of the Darwinian theory of improvement of the organism's performance through the "survival of the fittest". This occurs through the improvement of the genotype which goes to make up the organism. This is the basis of most evolutionary systems. Fundamental to this analogy are a number of important operational aspects of the model:

- the design description (structure) maps on to the phenotype
- separation of the representation at the genotype level from that of the design description level
- the processes of designing map on to the evolutionary processes of crossover and mutation at the genotype level
- performances (behaviours) of designs map on to fitnesses
- operations are carried out with populations of individuals.

In designing terms this maps directly onto the method of *designing as search*. We can describe this notion using the state-space representation of computation:

- state space is fixed at the outset
- state space comprises behaviour (fitness) and structure (phenotype) spaces
- genetic operators move between states in structure space, performance evaluated in behaviour space.

Designing as search is a foundational designing method but one that is restricted in its application to routine or parametric designing. In such designing all the possible variable which could occur in the final design are known beforehand as are all the behaviours which will be used to evaluate designs. Since the goal is to improve the behaviours of the resulting designs, the processes of designing during search map well onto those of optimization. This sits well with our notion of genetic algorithms and genetic programming. They can be readily viewed as robust optimization methodologies. Genetic algorithms and genetic programming have been used successfully as analogies of designing methodologies.

In this paper we will briefly explore other analogies which can be drawn from nature and humans' intervention in nature as possible sources for fruitful ideas on which to base design methodologies.

3. Extending the Genetic Analogy in Designing

There are two extensions of the genetic analogy that we will introduce in outline form here. Firstly, it is a well known hypothesis that certain behavioural characteristics of an organism could be genetic in origin. The field of genetic engineering deals specifically with this issue. *Genetic engineering* in natural systems is the human intervention in natural evolution. We will describe how such notions translate into potentially useful design processes. Secondly, in manufacturing there is a process known as *reverse engineering* where an existing product is explored in such a manner to construct a means by which it could be produced without knowing a priori its production methodology. We will describe how reverse engineering ideas fit into the genetic analogy and how they can be used to enhance nonroutine designing.

3.1. Genetic Engineering and Designing

The practice of genetic engineering in natural organisms involves locating genetic structures which are the likely cause of specified behaviours in the organism [1]. This provides a direct analog with finding significant concepts during the process of designing and giving them a specific primacy. The behaviour of the organism is an observable regularity which maps onto the concept and the structure of the genetic material which causes that behaviour is a representation of that concept, albeit a representation which has to be expressed in the organism for the concept to appear. The practice of genetic engineering is akin to the reverse of synthesis in the sense that one aspect of an already synthesised design is converted into the means by which it could be generated. In fact it is more complex than that since it is the behaviour of the already synthesised design which is the controlling factor but the analogy still holds. Let us examine in a little more detail the concept of genetic engineering.

Consider Figure 1 where the population of designs is divided into two groups (it could be more). One group exhibits a specific regularity whilst the other does not. The goal is to locate an "emergent" common structure in the genotypes of those designs which exhibit this regularity. Here "emergent" means that the structure was not intentionally placed there but could be found and represented for later use. Genetic engineering at this symbolic level uses pattern matching and sequence analysis techniques to locate these genetic structures. The process can be summarised as follows:

- locate emergent properties in the behaviour (fitness) space
- produce new genes which generate those emergent properties -> gene evolution
- introduce evolved genes into gene pool.

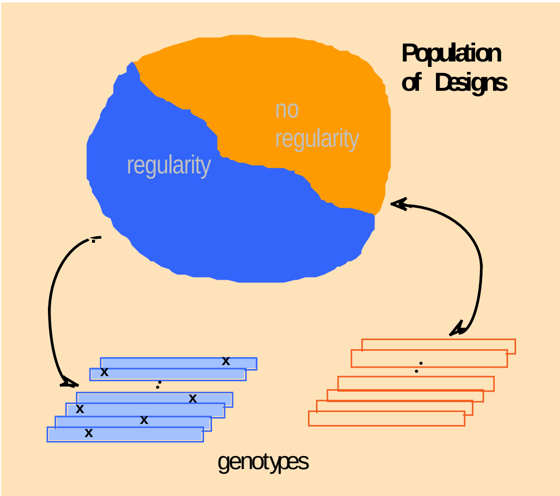

Figure 1. Genetic engineering is concerned with locating groups of genes' regularity, marked as X in the genotypes of those design which exhibit a specific behavioural regularity.

Take as an example the 8 genes shown in Figure 2 represented in the form of state transition rules. These genes are used to form the genotypes of designs within which a regularity is sought.

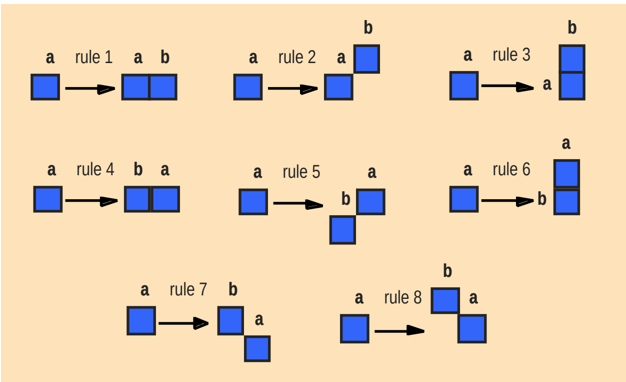

Figure 2. A set of 8 genes in the form of shape transition rules [2].

Figure 3 shows 10 designs produced from those genes. Each design is searched to determine some common regularity.

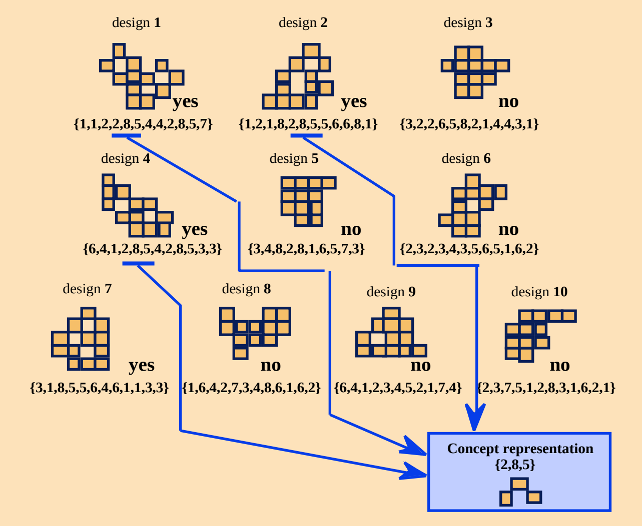

Figure 3. A set of 10 designs produced with the genes in Figure 2 and evaluated according to their regularity (“yes” in this case). Genetic engineering techniques emerge the gene group {2, 8, 5} as being the likely cause of that regularity [2].

The use of genetic engineering concepts allows an evolutionary system to adapt itself in ways different to traditional evolutionary systems when it takes the emergent gene structures and adds them to the alphabet of basic genes from which genotypes can be constructed. The effect of this is to dramatically change the probability landscape of the possible designs which could be produced from the original set of genes. No designs other than those which could have been originally produced are possible, however, the likelihood of designs be selected which exhibit those improved performances will now be increased.

It is possible not only to locate gene structures which map on to good performances, but also to locate gene structures which map on to poor performances. The former evolved genes would be increased in the genotypes whilst the latter would be decreased. The other aspect of genetic engineering concerns the manipulation of these newly found genetic structures or evolved genes. The goal of these manipulations is to improve the resulting performance of the designs. Typical manipulations which have readily modelled computational analogs include:

- gene therapy
- gene surgery
- radiation therapy.

These newly “evolved” genes capture some problem specific characteristics of the genetic representation of the good solutions to that problem. As such they may be able to be re-used in related problems to advantage. Typically each new problem to be solved using optimization techniques is treated anew without taking into account anything which has been learned from previous problems. Genes evolved using genetic engineered provide the basis for learning from previous design episodes and transferring what has been learned to the current design problem.

3.2. Reverse Engineering, the Genetic Analogy and Designing

In the computational model of genetic engineering used in designing the evolved genes are complexes of the original genes. Even when they are mutated they remain complexes of the original genes. As a consequence the boundary of the state space of possible designs is unchanged so that the designs produced are no different to those which could have been produced using the original genes only. In order to produce novel designs, ie designs which could not have been produced using the original genes only, the evolved genes need to be different to simply being complexes of the original genes. In order to "evolve" such genes different processes are required. We can take ideas from reverse engineering in manufacturing and include them in the genetic analogy.

The concept is analogically similar to that of genetic engineering in that emergent properties are looked for and new genes which generate those properties are produced, although the processes are different and the result is quite different. The process can be summarised as follows:

- locate emergent design (phenotype rather than fitness) properties
- reverse engineer new genes which can generate those emergent properties -> gene evolution
- introduce evolved genes into gene pool.

The critical differences between this and genetic engineering occur in two places in this process. The first differences in the locus of emergent properties – these are looked for in the phenotype, ie in the designs themselves rather than in their fitnesses or performances. The second difference is in the means by which "evolved" genes are created.

3.2.1 *Locating emergent features*

How might emergent features in designs be found? One way which has shown promise utilises the process of re-representation, ie an alternate representation is used to represent the design. In that re-representation it may be possible to locate features which were not placed there by the designer but appeared circumstantially. Figure 4 shows a figure produced by Escher. If the original representation is white images and the re-representation is black images then the angels are placed in the figure whilst the devils are emergent features.

Consider as a further example the following: a design is constructed by producing line segments, where line segments contain no branches. If the design is re-represented as maximal lines (i.e. lines in which line segments may be embedded) it is easy to find lines which are composed of more than one contiguous line segment if they exist by comparing the line segments with the maximal lines. If there is a complete isomorphism between the two then no such features have emerged. If there are maximal lines which are not isomorphic with existing line segments then an emergent feature has been found.

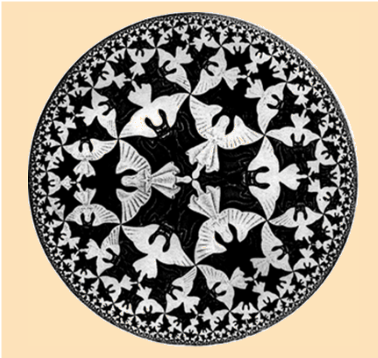

Figure 4. Figure by M. C. Escher "Circle Limit IV"

3.2.2 *Reverse engineering "evolved" genes*

Having located an emergent feature the next step is to reverse engineer a new gene which is capable of producing that emergent feature. This new "evolved" gene is then added to the gene pool. A variety of machine learning-based methods is available for this task. These include inductive substitution of the new representation in the place of the original representation in the design generator, turning constants into variables, and rule-based induction methods.

Evolving genes by reverse engineering is a form of Lamarckism in that characteristics of an organism not directly produced by its genetic makeup are acquired by that organism's genome.

4. The Developmental Biology Analogy in Design

So far the extensions described have all been at the genomic level of an adaptive system. In all of these as in most genetically-based adaptive systems used in designing there is an assumption that the mapping between the genotype and the phenotype is fixed. In natural systems the genotype is expressed through a phenotype through a biological development process which commences with the establishment of a single cell which divides. Further, cell division is a function of not only its genetic programming but also its environment. The normal genetic analogy does not allow for totipotency as occurs in nature at the outset of cell division. One approach is to allow a form of pluripotency to occur as a function of the development environment of the design.

Perhaps more interesting is to specifically model phenotypic plasticity to produce a form of pleiomorphism. This would allow for a form of genotype/phenotype environment interaction during the development of the phenotype. A variety of environmental interactions can be proposed to allow for adaptive mapping between genotype and phenotype. Classes of interactions include the following where “f” is some function:

- phenotype = f(genotype, situation), where situation refers to a state of the environment at some time, or
- phenotype_t = f(genotype, phenotype_{t-1});

both in lieu of :

phenotype = f(genotype).

Examples of such classes are:

Example 1

Here the phenotype is made up of components but the components themselves are some function of the path taken to reach that component. A simple path function would be that each component is in some way a function of the components it is connected to, ie:

- phenotype = {component_1,... component_i,... component_n}
- component_i = f(component_{i-1}, path[i-1,i]).

Example 2

Here the phenotype is developed over some time intermediate periods from a given genotype, during which various intermediate fitnesses control its development in a pleiomorphic sense, ie:

- phenotype = f(genotype, intermediate fitnesses during development).

Example 3

Here the phenotype, as it develops over some time intermediate periods from a given genotype, does so as a function of its expression at the previous time period. This is a crude model of cell division, ie:

- phenotype_t = f(genotype, phenotype_{t-1}).

Models such as these provide opportunities to include both problem- and domain-specific knowledge in the evolutionary process.

5. Discussion

The genetic analogy in designing has been based on a model of Darwin’s survival of the fittest [3, 4, 5]. This has provided a foundation for a body of important work which has implicitly treated designing as a search method largely akin to optimization. The effect of this in designing terms has been to set up a fixed state-space which is then searched for appropriate solutions. Alternative analogies drawn from both genetics and developmental biology offer the opportunity to change the state-space of possible designs in some cases, Figure 5.

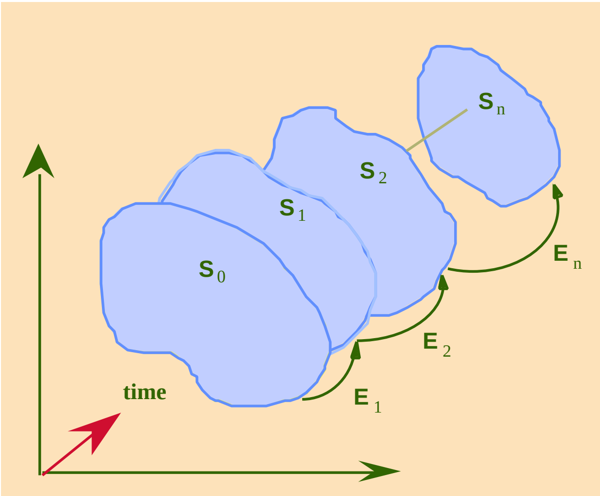

Figure 5. Conceptual model of designing with evolutionary and biological operations (E_i) which modify the state-spaces of possible designs.

The genetic engineering extension to the genetic analogy has been implemented in various environments and has proven to be both interesting and useful in allowing designing which slowly accumulates and represents, albeit implicitly, problem-specific knowledge which aids the solution process. The inclusion of reverse engineering is an obvious extension of genetic engineering to a different locus and has the potential to change the state-space of possible designs. Of interest here is whether the process terminates when applied iteratively.

The introduction of an analogy with developmental biology opens up numerous research paths with possible interest in designing. Concepts from research into natural systems such as switch genes, regulatory genes and gene networks offer a rich ground for fertile ideas.

Acknowledgments

This work is supported by a number of grants from the Australian Research Council.

References

1. Sofer W H, 1991. *Introduction to Genetic Engineering*, Butterworth-Heinemann, Stoneham.

2. Gero J S, Kazakov V, 1996. Evolving building blocks for design using genetic engineering: a formal approach. In: Gero, J S (ed.), 1996. *Advances in Formal Design Methods for CAD*, Chapman and Hall, London, pp 31-50.
3. Holland J, 1992. *Adaptation in Natural and Artificial Systems.* MIT, Cambridge, MA.
4. Goldberg D, 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading.
5. Koza J, *Genetic Programming.* MIT, Cambridge, MA.

This paper is a copy of: Gero, J. S. (1998) Adaptive systems in designing: new analogies from genetics and developmental biology, *in* I. Parmee (ed.), *Adaptive Computing in Design and Manufacture,* Springer, London, pp.3-12.