In press, *Artificial Life* journal

A Taxonomy for Artificial Embryogeny

Kenneth O. Stanley and Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 USA
{kstanley, risto}@cs.utexas.edu

Technical Report UT-AI-TR-02-297

December 6, 2002

Abstract

A major challenge for evolutionary computation is to evolve exceptionally complex phenotypes such as neural networks, sensory systems, or motor controllers at the same level of complexity as in biological organisms. In order to meet this challenge, many researchers are proposing indirect encodings, that is, evolutionary mechanisms where the same genes are used multiple times in the process of building a phenotype. Such gene reuse allows compact representations of very complex phenotypes. Development is a natural choice for implementing indirect encodings, if only because nature itself uses this very process. Motivated by the development of embryos in nature, we define Artificial Embryogeny (AE) as the subdiscipline of genetic algorithms (GAs) in which phenotypes undergo a developmental phase. An increasing number of AE systems are currently being developed, and a need has arisen for a principled approach to comparing and contrasting, and ultimately building, such systems. Thus, in this paper, we develop a principled taxonomy for AE. This taxonomy provides a unified context for long term research in AE, so that implementation decisions can be compared and contrasted along known dimensions in the design space of embryogenic systems. It also allows predicting how the settings of various AE parameters affect the capacity to efficiently evolve complex phenotypes.

1 Introduction

As the problems tackled by genetic algorithms become increasingly complex, it is becoming apparent that a direct mapping from genotype to phenotype, wherein each unit of the phenotype is represented by a single gene in the genotype, will no longer be effective (Hart et al. 1994). Novel problem domains, such as the evolution of complex neural networks and large commercial buildings (Bentley and Kumar 1999; O'Reilly 2000) require on the order of thousands or even millions of structural units for a single phenotype. If every gene were to map directly to a single unit of phenotypic structure, evolution would be searching through an intractable million-dimensional genotypic space.

In order to be tractable, the number of genes required to specify a phenotype must be orders of magnitude less than the number of structural units composing that phenotype. Nature has shown such representational systems to be possible on an enormous scale. Even with 100 trillion neural connections in the human brain, there are only about 30 thousand active genes in the human genome (2800 million amino acids; Dellaert

1995; Deloukas et al. 1998; Kandel et al. 1991; Zigmond 1999, page 9).

Such representational efficiency is made possible through gene reuse. In an indirect genetic encoding, a single gene may be used multiple times at different stages of development. There are two primary forms of reuse. First, phenotypic structures can occur in repeating patterns, where the same structural theme, perhaps with some variation, appears over and over again. Each time a pattern repeats, the same gene group can provide the specification. Examples of repeating patterns in biological organisms include the numerous left/right symmetries of vertebrates (Raff 1996, pages 302-303), and the numerous receptive fields in the visual cortex (Gilbert and Wiesel 1992; Hubel and Wiesel 1965). Repetition frequently involves variation on a general theme. For example, each vertebrae in the spine is formed similarly to the others, albeit with different incoming and outgoing connections (Zigmond 1999, pages 30-31) .

The second primary form of reuse occurs when the same gene product is used for different purposes in separate developmental pathways. For example, Cohn et al. (1997) found that the same gene product, fibroblast growth factor (FGF), induces the appearance of *both* forelimbs and hindlimbs depending on the part of the body where the FGF is applied. Thus, the same gene can be used to initiate different structures at different locations.

Natural organisms implement gene reuse through a process of development, or embryogeny[1]. The same genes can be used at different points in development for different purposes, and the order in which activations of genes take place determines when and where a particular gene is expressed (Raff 1996). Recently, researchers have begun to replicate this process in artificial developmental systems. The hope is that extremely compact codes can evolve to represent immensely complex phenotypes.

Researchers have used several names for artificial evolutionary systems that utilize a developmental phase, including Artificial Ontogeny (Bongard and Pfeifer 2001), Computational Embryogeny (Bentley and Kumar 1999), Cellular Encoding (Gruau 1994), and morphogenesis (Jakobi 1995). We adopt the term Artificial Embryogeny (AE) to refer to the entire class of such systems[2]. Because AE offers a methodological approach for reaching the level of complexity seen in natural organisms, the evolution of the body and brains of artificial organisms has been a popular goal for researchers in this field (Bongard and Pfeifer 2001; Dellaert and Beer 1996; Hornby and Pollack 2001b; Komosinski and Rotaru-Varga 2001) Thus, the primary objective of AE is to evolve levels of complexity that have heretofore been out of reach. It is not *necessary* for systems to faithfully simulate low-level biological development processes, except insofar as the simulation of these processes may help in achieving high complexity. Significant experimental exploration of possible implementations, both faithful and unfaithful to biology, will be necessary to achieve this goal.

With growing interest in the field, and a large number of systems being introduced, a need has arisen for a common framework to analyze and compare them. By identifying the dimensions along which design decisions can vary, future experiments can reveal their costs and benefits. For this reason, in this paper, we build a taxonomy for AE systems based on a principled analysis of both existing systems, and biological research on embryogeny. Ultimately, this taxonomy makes it possible to predict the outcome of specific design decisions along well-defined dimensions.

We begin by reviewing existing systems within a preliminary framework, distinguishing between grammatical and cell chemistry approaches. All these systems utilize indirect encodings, however, the encoding can either be explicit or implicit (Bentley and Kumar 1999). In an *explicit* encoding, genes that describe steps in a developmental process are only used once. Such encodings differ from *implicit* encodings, where genes

[1]Bentley and Kumar (1999) pointed out that the correct term is *embryogeny* as opposed to *embryology*. Embryogeny is the embryological process of development itself, while embryology is the *study* of the process of development. Since we are attempting to evolve actual developmental systems, we are implementing artificial embryogeny.

[2]*Embryogeny* conveys that systems in this class develop phenotypes using genetic information starting from a small initial structure.

can be activated multiple times. Our review focuses on implicitly encoded AE systems because reuse is one of the primary motivations of AE. We then survey research in biology that provides insight into the mechanisms behind embryogeny. Five major dimensions of development emerge from the biological overview: (1) Cell Fate, (2) Targeting, (3) Heterochrony, (4) Canalization, and (5) Complexification. Finally, we use these five major dimensions to replace our preliminary framework with a mature multidimensional taxonomy of AE methodologies.

2 Review of Artificial Embryogeny

This section reviews prior work in AE by examining two parallel lines of research. The first type is the *grammatical approach*, originated by Lindenmayer (1968). The grammatical approach evolves sets of rules in the form of grammatical rewrite systems. The grammar can be context-free or context-sensitive and can utilize parameters. Variations on this theme include using instruction trees or directed graphs in place of actual grammars.

The second type of AE, the *cell chemistry approach*, is inspired by the early work of Turing (1952), who introduced a mathematical model of diffusion and reaction within a physical substrate. This approach attempts to more closely mimic how physical structures emerge in biology. Cells are arranged in a physical space where simulated proteins can be sent as signals from one cell to another, as in nature. Growth processes such as axons and dendrites can form connections between cells through complex targeting mechanisms. Protein structures are produced by genes in a cell's genome if the proper regulatory proteins already exist inside the cell's cytoplasm. In effect, the proteins inside the cell are like preconditions to rules in the grammatical approach. This approach attempts to approximate the functionality of natural development through physically-motivated implementation. The hope is that by closely simulating the lower-level processes of biological development, more natural, and hence more complex, phenotypes can evolve.

We make the distinction between grammatical and cell chemistry approaches for two reasons: (1) It is currently the most recognizable distinction among AE systems, and (2) it reflects the general division in artificial intelligence between high-level, top-down approaches and low-level, bottom-up approaches. However, it will turn out that this initial distinction is ultimately superficial, and that a more sophisticated taxonomy of AE systems can be derived based on relevant biological dimensions. We begin by introducing existing systems in these two categories, and will introduce a biological taxonomy in the next section.

2.1 Grammatical Approaches

Using grammar to model biological development traces back to Aristid Lindenmayer, who introduced a grammatical rewriting system called *L-systems* (Lindenmayer 1968). Lindenmayer observed that complex natural objects, in particular plants, could be described by iteratively replacing simple parts with more complex parts. This idea is naturally expressed grammatically, where symbols on the left side of production rules are replaced in parallel by strings on the right. The strings generated by L-systems can be interpreted as morphological or graphical descriptions, yielding complex fractal-like objects. In other words, L-systems are implicit encodings that output an explicit encoding as a string of developmental instructions. For example, figure 1 shows how two simple rewrite rules produce a tree structure. Symbols produced by the rules describe directions of growth in the tree: "B" means "grow forward," "$-$" means, "turn direction of growth left," and "$+$", "turn direction of growth right." Thus, L-systems can encode complex morphologies using relatively simple rules.

The grammar can be made more powerful by adding parameters to the rewrite rules, so that the same

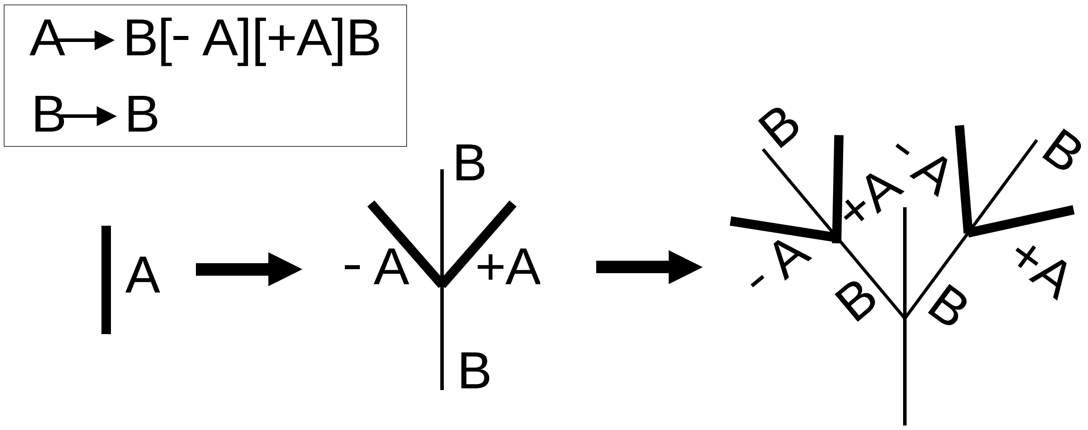

Figure 1: **Grammatical Approach Example (L-systems; Lindenmeyer 1968)..** The two rewrite rules (inset) describe the growth of a tree-like morphology. The symbol A, shown as a thick line in the tree, is the only symbol that is rewritten in this grammar. The symbol B, which does not expand, becomes a thin branch, and $-$ and $+$ determine relative angles of branches expanded from A symbols. This example illustrates how a few simple generative rules can encode a large structure with many components.

rules can generate different structures depending on their parameterization (Lindenmayer 1974). In addition, if one parameter is used as a counter, it can be used to terminate growth processes after a specific number of iterations. Systematically decrementing the counter each time allows rules to naturally express a stopping case when a parameter reaches zero. Consider the following example:

$$
\begin{array}{lll}
A(n): & (n > 1) & \rightarrow B(n)A(n-1) \\
 & (n \leq 1) & \rightarrow A(0) \\
B(n): & (n > 2) & \rightarrow C(0) \\
 & (n \leq 2) & \rightarrow D(0) \\
C(n): & & \rightarrow C(n) \\
D(n): & & \rightarrow D(n)
\end{array}
$$

Starting with A(4), this L-system would produce,

$$
\begin{array}{cc}
 & A(4) \\
\rightarrow & B(4)A(3) \\
\rightarrow & C(0)B(3)A(2) \\
\rightarrow & C(0)C(0)B(2)A(1) \\
\rightarrow & C(0)C(0)D(0)A(0)
\end{array}
$$

In addition to utilizing parameters, rules can also be made context sensitive, or be applied stochastically (Prusinkiewicz and Lindenmayer 1990). In a context sensitive L-system, rule activation is not only contingent upon the correct nonterminal symbol being present, but also upon the symbols that surround that nonterminal. In stochastic L-systems, successors are chosen probabilistically, leading to randomized final structures. As another refinement, Vaario (1994) introduced an elaborate object-oriented extension of L-systems in which individual objects and subobjects in a developing organism contain their own rewrite rules. Different refinements to L-systems affect the kinds of structures they can evolve in varying ways. The effects of such refinements is an open area of research.

Many AE researchers have utilized L-systems and similar grammatical methods (Belew and Kammeyer 1993; Boers and Kuiper 1992; Hornby and Pollack 2001a,b; Kitano 1990; Sims 1994). In such systems, a set of production rules is evolved for each genome. Starting from a canonical embryological start symbol, embryos are grown by repeatedly applying rules to the symbols in the developing embryo.

Interestingly, L-systems were not initially designed to be evolved. They were meant to model or describe the growth of fractal-like designs in nature. Thus, a potential problem for L-systems-based AE is that the capacity to *describe* embryogeny does not necessarily imply a capability to *evolve* embryogeny.

Still, many researchers have pursued the evolution of L-systems and other grammatically-based encodings. One interesting such research direction is the evolution of artificial neural networks (i.e. *neuroevolution*). Neuroevolution has long been a major theme in AE, partly because the potential complexity of neural networks is known to be so high. The first rewrite neuroevolution system was developed by Kitano (1990), who showed that it was possible to evolve the connectivity matrices of artificial neural networks through a set of evolved rewrite rules. Instead of utilizing a string on the right hand side of a rule, Kitano used a small 2 by 2 matrix. If the elements of the matrix are nonterminal symbols, then they are themselves rewritten with 2 by 2 matrices from their respective rules, expanding the size of the matrix exponentially. Eventually, when terminals are reached, a complete matrix of numerical values has been created. The matrix then represents the connectivity of a neural network. Although nondevelopmental encoding schemes have since been shown to perform better (Siddiqi and Lucas 1999), Kitano's early graph generation grammar showed that it was possible to use rewrite rules to encode neural networks for encoder/decoder problems.

Other researchers have experimented with alternative ways of encoding rewrite rules, aiming at grammar encodings more suitable for evolution. Boers and Kuiper (1992) used context sensitive L-systems to evolve neural network topologies. They encoded the rules in a genome as a bit string, which can be recombined using simple genetic operators. Resulting bit strings were then translated back into rewrite rules. Boers and Kuiper were able to evolve solutions to several conventional problems, such as XOR and handwriting recognition. However, these results did not provide conclusive evidence of the capability of L-systems to solve these problems since the weights were found through backpropagation. It is not clear that the complex developmental process was necessary.

While previous work focused on narrow problems that did not necessarily demonstrate the power of grammatical encodings, Sims (1994) chose to evolve the body morphologies and neural networks of artificial creatures in a simulated 3D physical world. In this domain, the power of generative encodings is easy to observe, since resulting creatures are animated in a simulated environment. Sims used directed graphs as the genotypes for his experiment. In these graphs, a node represents a body part and an edge specifies how body parts are connected (figure 2). The nodes and edges together work like L-system rewrite rules: an edge is like an atom of a rewrite rule and a node is like a terminal. By following recursive edges, Sim's system reuses genetic material. 3D animations of evolved creatures displayed strikingly natural-looking gaits.

Using a similar domain as Sims, Hornby and Pollack (2001a,b) applied L-systems to the simultaneous evolution of the body morphologies and neural networks of artificial creatures in a simulated 3D physical environment. Before Hornby and Pollack's experiments, Lipson and Pollack (2000) had already evolved creatures that could locomote in such an environment, and successfully transfered their designs to robots in the real world. However, Lipson and Pollack did not employ a developmental encoding, instead utilizing a

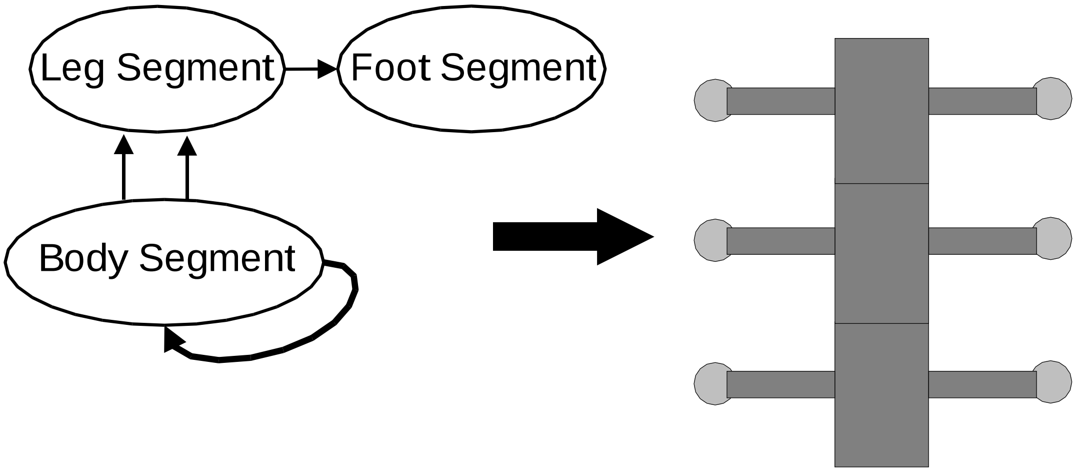

Figure 2: **Development of Body Morphology (Sims 1994).** The graph on the left specifies how the morphology on the right develops. The body segments repeat because of the recurrent loop on the "body segment" instruction node, which allows the reuse of genetic code. The number of repetitions is determined by an evolved parameter in the loop which is not shown. The final structure is a centipede-like creature with six legs and feet on each leg. Sims' work has inspired many AE researchers to explore body-brain evolution in simulated 3D environments.

direct encoding. Hornby and Pollack's work built on this earlier work by using L-systems as a developmental encoding. Using this encoding, they evolved creatures to locomote without dragging most of their parts on the ground. Parametric L-systems with 20 rewrite rules were evolved. The resulting creatures were compared with others evolved with a non-developmental encoding, which specified a sequence of explicit build commands. The developmental encoding evolved dramatically more complex and more fit creatures than the non-developmental encoding. Developmental creatures displayed significantly more body segments and repeating structures. Some creatures even displayed symmetrical arrangements of parts. The results demonstrate that reuse is an important capability for embryogenic systems. However, there are many morphologies that can potentially support locomation. Therefore, finding such a morphology may not be difficult. The question remains whether L-systems can be evolved for a stricter task with a small set of possible solutions.

Even though many rewrite systems evolve neural networks, other structures have been used as well. For example, a sorting network is a simple program that designates the order of place comparisons for sorting a sequence. Belew and Kammeyer (1993) evolved rewrite rules similar to L-systems to specify how sorting networks develop. A sorting network is an ordered set of place comparisons, where an item at one position in a list is compared to an item at a second position. If they are out of order, their positions are switched. A grammar encoding was used because the development of a sorting network can exploit symmetries. For example, the left and right halves of an unordered sequence can be both sorted independently using the same procedure, and then finally combined for a final ordering. The developmental encoding only needs to encode one of the symmetrical procedures and then expand it once for each half of the sequence, thus exploiting the symmetry of the problem. Belew and Kammeyer's evolution of sorting networks demonstrated that AE is applicable to practical problems beyond the evolution neural networks and body morphologies.

L-systems encode rewrite rules that determine the steps in a developing phenotype. Therefore, they are naturally implicit encodings. Some AE researchers have preferred grammatical encodings that are more explicit. These encodings reuse genes through subroutines and recursion, rather than by applying of rules under specified conditions. For example, Gruau (1993, 1994) and Gruau et al. (1996) used *grammar trees* to encode steps in the development of a neural network starting from a single ancestor cell. This system is

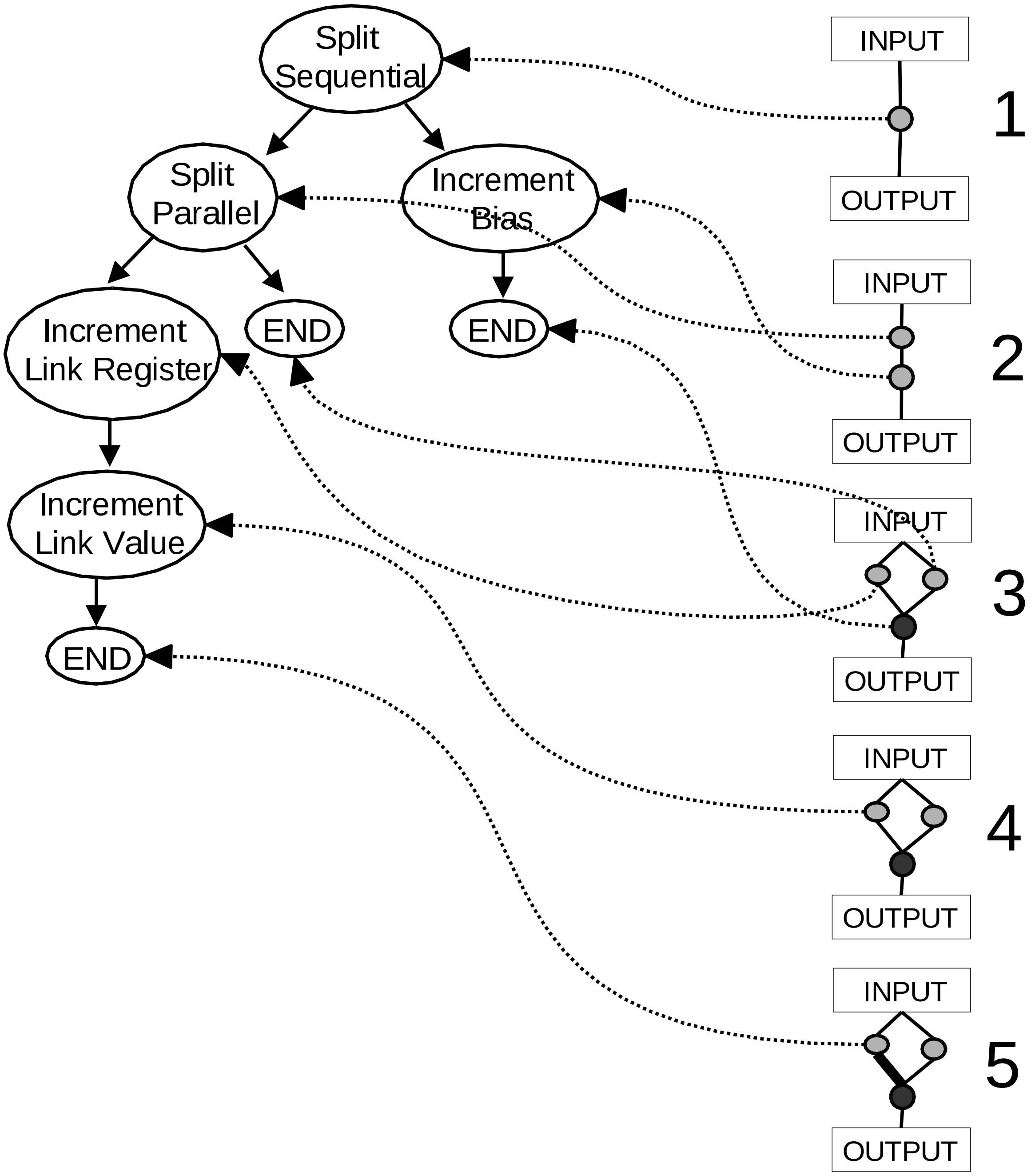

Figure 3: **Cellular Encoding (CE; Gruau 1993,1994) Example.** The grammar tree is shown at left and the network growth is shown from top to bottom in five steps at right. The network begins at step 1 as a single cell. At each step, each network cell is reading from its own part of the tree. Dotted lines identify the location in the tree from which each cell is reading at each step. When a cell splits, its children cells take separate paths in the tree. The "Increment Link Register" instruction is the way a cell knows to which link it should apply any subsequent link-based instructions. In the example, such an instruction occurs immediately following the link register instruction, causing a link's value to increase, represented in the network by a thickening line width. The neural activation bias is represented by cell darkness. This example is based on others given by Gruau (1993). While Gruau uses abbreviated instructions, we spell them out entirely to make the example easy to follow. Gruau (1993) proved that Cellular Encoding grammar tres can describe any network topology.

called *Cellular Encoding* (CE) because its rules apply to single cells in a growing network. The grammar tree contains developmental instructions at each node. For example, CE includes functions that split one cell into two cells, change the values of links between cells, and remove existing links between cells (figure 3).

Developing cells in CE read from different parts of the grammar tree at the same time. A *reading head* for each cell indicates from which part of the grammar tree it is reading. When a cell encounters an *end* instruction, its state is finalized and it stops reading. CE uses a First In First Out (FIFO) queue of cells in order to keep track of which cells are currently executing instructions, and in what order they should be executed. A cell at the front of the queue executes the instruction to which its reading head points, moves its head to the subsequent instruction in the grammar tree, and then goes to the end of the queue. Sometimes cells encounter instructions to divide (there are a variety of cell division methods), in which case the original cell moves its reading head down the left subtree, and the new cell moves its head down the right subtree. Thus, cell divisions allow different cell lineages to follow different developmental pathways.

Gruau showed that by adding a decrementing counter and a *recursion* instruction to the family of available instructions, it is possible to move the reading head back to the top of the tree and reexecute the same developmental code multiple times. In fact, Gruau gave an example of how a neural network that solves the three bit parity problem can be encoded by reusing the developmental code of an XOR network twice (Gruau 1994). In addition, CE can be extended to encode multiple trees, each with the ability to jump into the others, so that individual trees can be reused multiple times like subroutines.

Unlike L-systems, CE encoding was designed to be evolved, because a methodology for evolving grammar trees, Genetic Programming (GP; Koza 1992), already existed. Thus, by representing CE grammar trees as LISP S-expressions, CE took advantage of the GP evolutionary methodology. Gruau showed that CE could evolve repeating structures in problems such as parity and symmetry, where the same procedure needs to be repeated over and over again. In fact, CE was able to evolve solutions for *families* of parity problems by using a parameter for the number of inputs. The most inputs demonstrated was 51, confirming CE's powerful capacity to reuse structure.

CE has also been applied to body-brain evolution. Komosinski and Rotaru-Varga (2001) used a grammar tree encoding similar to CE to evolve the body morphology and neural network for agents in a simulated $3D$ world. They showed that the developmental grammar tree encoding evolved modular structures with repeating parts for simple tasks such as growing to a tall height, and locomotion.

Although initial results with grammar trees on the parity problem and in the body/brain evolution domain are promising, CE has not performed as well in other difficult and well-established problem domains. Stanley and Miikkulainen (2002c) showed that a non-developmental encoding called NeuroEvolution of Augmenting Topologies (NEAT) was able to evolve solutions to a difficult non-Markovian version of the problem of simultaneously balancing two poles on a cart using 25 times fewer evaluations than CE. This comparison suggests that the capacity to reuse genes alone does not ensure efficiency. The ultimate developmental encoding should be able to *both* harness the power of reuse for the discovery of complexity, *and* find solutions quickly.

Luke and Spector (1996) identified several aspects of CE that could be changed to improve performance:

- Crossing over subtrees of CE genomes changes the order in which their operations are executed. Since the effects of CE's operators depend on their execution order, the same subtree in one genome may result in a very different phenotype when inherited by an offspring.
- Because CE's operators are node-centric, CE's ability to make specific and precise modifications to connections is limited.
- CE creates networks by splitting many cells into two or more interconnected cells. Therefore, the

networks tend to be highly interconnected, which may slow down the search in some tasks where high connectivity is not required.

Luke and Spector (1996) suggested *edge encoding* as a solution to these problems. In edge encoding, networks are grown by modifying the edges in a graph rather than the nodes, and the grammar trees are traversed in depth-first rather than breadth-first order. There is currently no experimental comparison between CE and edge encoding. However, the problems with CE that edge encoding is designed to solve are important in their own right, since they reveal the importance of evolutionary bias in grammatical AE. Because grammatical approaches have formal structure, it is possible to construct proofs about the kinds of topologies they can express. For example, Gruau proved that CE can express any possible neural network topologies. However, Luke and Spector's critique, along with the pole balancing results, suggest that expressivity is *not* the most important property of an AE encoding. Rather, the kinds of architectures that encodings are *biased* towards, and their capacity to apply crossover and mutation operators to genomes without damaging their functionality, are of utmost importance. L-systems face the same problems. Thus, there are several challenges for grammar-based encodings in the future: What kinds of architectures do they tend to produce? Are these the kinds of architectures we want? Do they make sense as a genetic code subject to genetic operators?

There is a final question about grammatical encodings that does not concern their efficiency or biases, yet has inspired an entirely different approach to AE: Are they a natural way to implement the process of biological development? In the next section, we review systems that attempt to more closely simulate the low-level aspects of natural embryogeny. Although we do not take the perspective that biological simulation is *necessary* for a good encoding, it may still provide useful insight into the implementation of developmental systems. Because nature has evolved extremely compact encodings, it is important to explore the space of systems that attempt to follow the same path. The next section describes such biologically-motivated systems.

2.2 Cell Chemistry Approaches

Turing (1952) first modeled biological development at the chemical level with his *reaction-diffusion model*. Although the model was not meant to encompass all of development, it did account for patterns seen on the exterior of natural organisms, such as the coloring patterns on bird feathers and animal coats. In the model, a set of equations describe how the concentrations of different substances, or *morphogens*, change over time due to their diffusion and reaction with each other. The patterns produced by the model are strikingly similar to patterns found in nature, validating the model.

According to Turing (1952), a vector of morphogen concentrations, C, changes temporally and spatially as described by Turing's partial differential equation,

$$\frac{\partial C}{\partial t} = F(C) + D\nabla^2 C, \tag{1}$$

where $F(C)$ is a nonlinear function describing how the morphogens represented by C react with each other, and D is a diagonal matrix representing the relative magnitude of the diffusion coefficients for the different morphogens. Thus, the equation describes both reactivity and diffusion at a point in space. The equation is simultaneously applied at every point on a grid, for every time step, yielding a dynamic system that stabilizes on interesting patterns under the right conditions.

Like L-systems, Turing's reaction-diffusion model was not originally designed to be evolved. Although many grammatical approaches to AE follow the original L-system model, cell chemistry approaches tend to use Turing's model only as an inspiration. It describes lower level phenomena than L-systems, and therefore does not lend itself as easily to evolution of complex forms. However, cell chemistry approaches rely at least on the abstract concepts of diffusion and reaction, augmented with additional biologically-inspired components.

For example, Nolfi and Parisi (1991) modeled "diffusion" at the level of axon growth, and "reaction" as the interaction between axons and cell bodies. Thus, their model is significantly higher-level and more abstract than Turing's. Each neuron is defined by a single gene in a fixed-length genome. Each gene describes the branching and positional properties of a corresponding neuron. The branching properties specify how branches will grow, or diffuse, from the cell body of the neuron. When growing axons hit other cells during development, connections are made. Then, in a later phase, non-functional connections are pruned. This approach utilizes growth to create connections without explicitly describing each connection in the genotype. However, since every neuron must be directly specified, genes are not reused, limiting the complexity of what can be represented. Their later work (Cangelosi et al. 1993) extended the model to include cell divisions and migrations based on rewrite rules, in addition to the axonal growth. Interestingly, this new model used ideas from both the grammatical approach and cell chemistry approach in order to allow genes to be used more than once.

In contrast to Cangelosi et al.'s abstract model of neural growth, Fleischer and Barr (1993) produced a lower-level model of neural development more reminiscent of Turing's differential equation models of diffusion. Cells on a $2D$ plane move independently. Each cell has its own internal state that controls its behavior. The outside environment has chemical, mechanical, and electrical properties. Extracellular chemicals diffuse through the environment based on differential equations, just as in Turing's model. In addition, the genome itself is a set of differential equations that take as arguments the cell's internal state and the state of the outside environment. This framework allows cells to move and interact through chemical diffusion and axon growth in order to form neural networks. Fleischer and Barr (1993) drew the interesting conclusion that combining multiple mechanisms to build a pattern can be more robust than using a single mechanism. For example, a cell's identity can both be regulated by its lineage and the chemical messages in its surrounding environment. Fleischer and Barr were able to demonstrate several interesting low-level behaviors. For example, an axon was able to grow along a chemical gradient and find a target neuron while avoiding obstacles. However, their genomes were hand-coded and not evolved. Similarly, Mjolsness et al. (1991) used a model of low-level chemistry and gene activation to simulate the early development of *Drosophila* flies, but did not combine their model with evolution.

Astor and Adami (2000) introduced an artificial chemistry similar to Fleischer and Barr's model, where chemicals diffuse based on diffusion equations, and genes are activated if chemicals that they explicitly specify are located in the cell cytoplasm. Astor and Adami showed that they could hand-code a genome for a neural network that displayed classical conditioning behavior. In other words, through repeated activation, the network learned to associate an output with a stimulus that initially did not activate the output. Astor and Adami (2000) did not report results for evolved networks, although they proposed future evolutionary experiments.

Dellaert (1995) pointed out that the computational complexity of such low-level simulation might make evolution difficult. Thus, Dellaert and Beer (1994a,b) and Dellaert (1995) designed an ambitious and extensive model of development meant to be both biologically defensible and computationally tractable. The model centers on the idea of *genetic regulatory networks* (GRNs), which are networks of interactions between genomes and their environments that lead to a sequence of state changes inside each cell (figure 4). While the diffusion and reaction of various chemicals in the Fleischer and Barr model can be understood

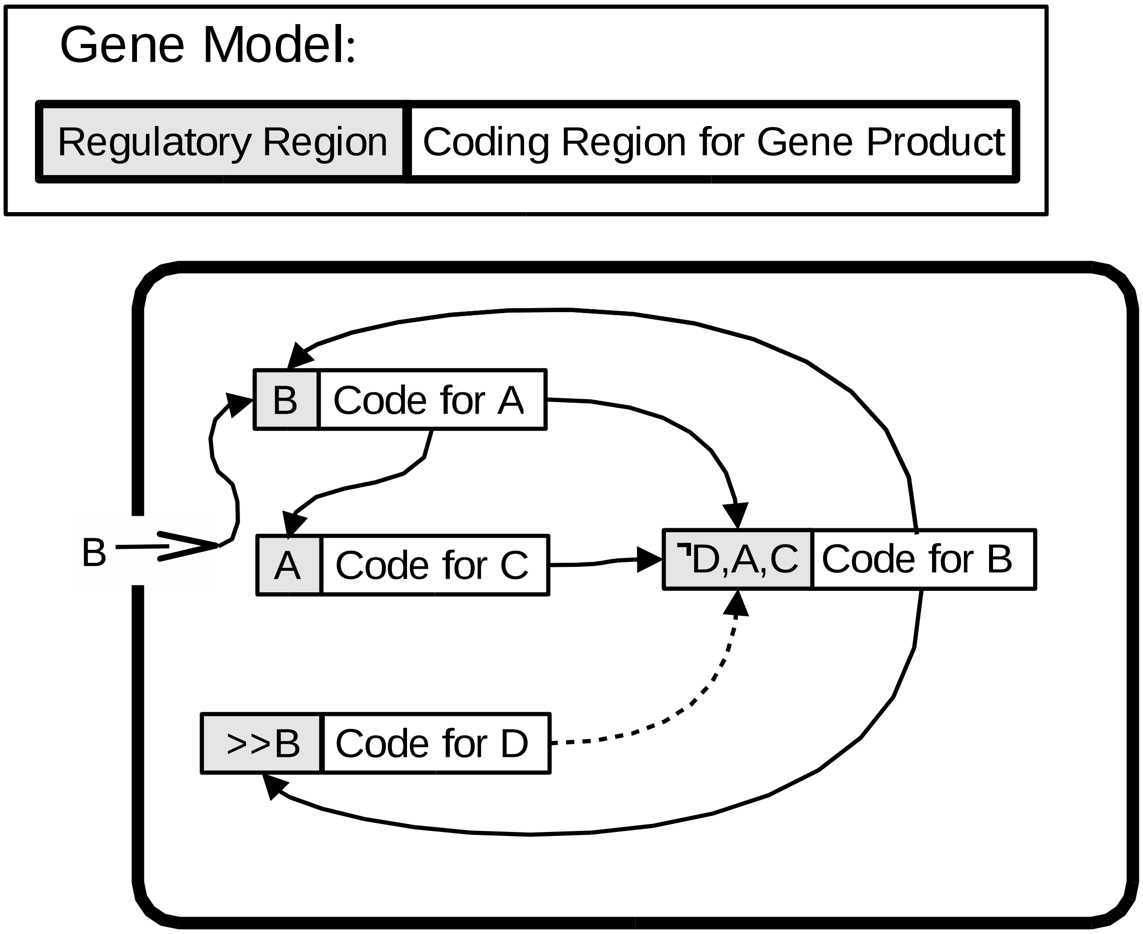

Figure 4: **Genetic Regulatory Network Example.** Each gene is modeled as a regulatory region and a coding region that codes a particular product (e.g. a protein in a natural cell). A simple network showing how different gene products of some genes regulate other genes is shown inside a cell, depicted as a rounded rectangle. The network describes a system that produces a number of products and then turns off when enough of product B is produced. The symbol “$>>$” means a large amount. The diagram shows that the entire network becomes activated when product B enters the cell from an external source. B causes A to be produced, which in turn causes C to be produced by another gene. A and C, without D in the cell cause more B to be created, which in turn feeds back into the production of A, further strengthening the cycle. Eventually, when a great deal of B is present, D is finally produced, stopping the generation of B and ending the feedback cycle. The GRN shows that interesting dynamics can result from the regulatory interactions of different genes. In an AE model, gene products might e.g. cause axons to grow or reduce neural thresholds.

as a GRN, Dellaert and Beer use a more abstract model of GRNs which captures the dynamics of such a system without descending to the physical level.

Dellaert et al. implemented their GRN using boolean functions called *operons* inside the genome. For example, consider the following two operons:

$$Operon \quad 1 : A \wedge \neg B = C$$
$$Operon \quad 2 : A \wedge C = B$$

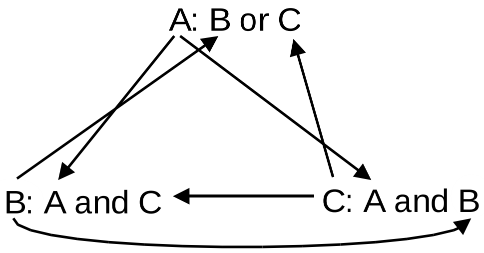

Figure 5: **Random Boolean Network (RBN) Example.** The state of the network is given by the boolean values of A, B, and C. At each timestep, the state values are updated based on their values in the previous timestep. In the approach of Dellaert and Beer (1996), the current state of a cell is given by the current state of its RBN. Individual bits in the state signal events like cell splitting or creating connections between cells. RBNs were introduced by Kauffman (1993) to simulate the protein expression patterns in cells in a developing embryo. RBNs are computationally less expensive than full-blown GRN implementations (figure 4), while exhibiting similar dynamics.

The first equation means that if protein A is present in the cell cytoplasm, and protein B is not present, then protein C is produced and added to the cytoplasm. The presence of A and C satisfies the second operon, yielding B in the cytoplasm for the first time. At that point, because operon 1 only produces its product in the absence of B, the cell would stop producing protein C. This example illustrates how a complex network of interactions can result from operons inside a cell. By representing the presence or absence of a proteins using boolean equations, the computational expense of simulating diffusion through differential equations is eliminated.

Several low-level biological processes utilize the proteins produced by operons. For example, a facility is included to allow one cell to introduce a new protein to another cell through a receptor. In addition, cells grow axons if they have a special axon-growing protein in their cytoplasm. The axons then connect to cells with the proper target protein. The axons find their path by growing around cells with special path designating proteins. Thus, a nervous system can develop in a biologically plausible manner.

Cells divide in the simulation when a special dividing protein is present. After division, both descendant cells are in the same state, which leads to a problem. How can the symmetry ever be broken so that cell differentiation takes place? The model solves this problem by introducing a single symmetry-breaking protein after the first cell-division, and by allowing communication between cells.

The GRN model revealed an important challenge for AE. Dellaert and Beer (1996) *hand-coded* the genome for both the body morphology (i.e. sensors and actuators) and the nervous system of an obstacle-avoiding vehicle. Subsequently, they were able to evolve incrementally improved versions of the vehicle off the initial hand-coded genome. However, they could not evolve such a vehicle from scratch. They attributed this failure to the massive search space introduced by such an expressive encoding. Intractable search spaces are a significant challenge to AE systems in general.

Dellaert and Beer (1996) were able to overcome the large search space problem by greatly simplifying their model, using random boolean networks (RBNs) instead of the more complex operon model. RBNs, initially introduced by Kauffman (1993), are simple boolean expressions whose outputs are connected to the inputs of other boolean expressions, forming a network (figure 5). The network has a state defined by the current outputs of all the boolean expressions, which defines the current state of the cell. In addition, Dellaert and Beer simplified the axonal outgrowth model, such that cells were simply connected if they expressed the right proteins, foregoing the entire axon growth phase. Interestingly, with this simplified model, they were able to evolve a solution to the obstacle avoidance task from scratch, in addition to evolving a line-following vehicle. The lesson is that simple solution structures sometimes perform better than biologically plausible ones; abstraction can be a powerful tool.

Researchers have introduced a number of cell chemistry-based models similar to that of Dellaert and Beer (Bongard and Pfeifer 2001; Eggenberger 1997; Jakobi 1995). Jakobi (1995) also used a GRN model based on proteins interacting with a genome and transferring between different cells. However, instead of boolean expressions, Jakobi utilized *templates*, which are strings of characters that could be matched with proteins. The templates were parts of genes. This encoding allowed the entire genome to be evolved as a string, similarly to DNA. Jakobi evolved simulated robots that could move down corridors and wander without hitting obstacles. Eggenberger (1997) used a similar string encoding with template matching for genetic regulation. Eggenberger also introduced a protein that causes cells to die, since cell death is common in biological development.

Unlike Jakobi and prior researchers, Eggenberger experimented with much simpler tasks than evolving neural networks. Instead, he evolved bilaterally symmetric shapes, to show that symmetry could develop naturally through reuse. A variety of symmetric three-dimensional morphologies evolved, showing how discovering symmetry can be beneficial. Eggenberger's simple symmetry task brings up an important question for AE researchers: Should experiments focus on evolving elaborate nervous systems, like Dellaert and Beer (1996) and Jakobi (1995), or on very simple proofs of concept in order to demonstrate performance potential, like Eggenberger? Given that eventually we need to evolve very complex phenotypes that develop through gene reuse, experiments at this stage should indicate that methods have the potential to reach this goal. Body-brain evolution can sometimes lead to unclear conclusions. For example, even if it is possible to evolve a vehicle that avoids obstacles, does that success tell us anything about why development is necessary, or is it just an easy problem that a non-AE system could have solved just as well? In contrast, bilateral symmetry is a strategy used by many complex organisms to reuse the same structures efficiently. Thus, sometimes simple experiments can demonstrate a potential for complex development.

Like Eggenberger, Bentley and Kumar (1999) also chose to experiment at the level of abstract proofs of concept. They utilized a simpler cell chemistry approach in which colored cells on a grid expanded or died depending on whether their surrounding cells matched preconditions in a genome of rules with preconditions. The preconditions described the configuration of surrounding cells and location in the grid. The encoding allows reuse because preconditions may occur over and over again at different grid locations in the same developmental sequence. Although this encoding resembles grammatical approaches, its preconditions instead check the local environment in a grid of cells. To verify that their encoding was efficient, Bentley and Kumar evolved *tessellating tiles*, which are shapes that can be fit together in groups of four without overlapping. Bentley and Kumar showed that their implicit encoding evolved tessellating tiles more reliably than a number of explicit encoding schemes, supporting the hypothesis that gene reuse can support the development of form. Because both the task and the encoding were simple, the experimental results were easy to interpret, just as with Eggenberger's work on bilateral symmetry.

Even as some cell chemistry AE researchers moved towards simpler proof of concept experiments, recent research builds on the earlier ambitious experiments of Jakobi and Dellaert and Beer. Many researchers using cell chemistry approaches attempt to evolve sophisticated behaviors through the evolution of the brains and bodies of creatures in a $3D$ simulated world, just as Hornby and Pollack (2001b) have done with L-systems. Bongard, Paul, and Pfeifer (2000,2001,2002) used a GRN-based encoding similar to Eggenberger's to evolve such creatures. The creatures were evaluated on their ability to walk towards a block and push it some distance on the ground. Bongard et al. were able to show that repeated phenotypic structures were evolved by reusing genes. Later, Bongard (2002) showed that the genes affecting morphology had evolved a separate developmental pathway from those affecting neurogenesis, even though the same gene products could affect both pathways. This result implied that changes in morphology could occur through mutation without disrupting neurogenesis and vice versa. Thus, a form of modularity was evolved in which the neural network and body morphology develop in parallel yet independently of each other. Such modularity allows evolution to search over varied body-plans without affecting neurogenesis, and to discover different neural networks for the same body-plan. Thus, the GRN-based encoding allowed a high level of flexibility in the kinds of mutations that were feasible.

2.3 Summary

In this section, we have divided the space of implicit encodings for AE into grammatical and cell chemistry approaches. Although they have similar goals, their origins are different. While grammatical approaches tend to be abstract and high-level, cell chemistry approaches are generally lower level and more strictly motivated by the biological mechanisms of development.

Experimental approaches used to test AE methodologies can also be divided into two main categories. While many AE researchers pursue the goal of evolving the brains and bodies of physically realistic creatures in a $3D$ environment, others concentrate on more easily analyzed low-level problems such as symmetric morphology, or neural networks alone. Many AE researchers are interested in body/brain evolution, probably because biological animals evolved through *real* embryogeny. Thus AE should also allow the evolution of the bodies and brains of artificial animals. However, because AE is a young field, it also makes sense to perform simpler experiments in order to test fundamental hypotheses about how the reuse of genes allows the evolution of complex phenotypes.

Hart et al. (1994) argued that development allows a simpler genotypic search space than would be possible through directly searching over phenotypes. Through gene reuse, genetic components can be used as modules, freeing evolution from having to discover the same concept more than once. Experimental evidence supports this argument by showing that genetic reuse and modularity indeed arise in many AE systems. However, no system to date has attained complexity on the scale of natural organisms. Since it is this level of complexity to which AE aspires, more research is needed in order to uncover the properties of different AE encodings.

In fact, the current divided perspective is not necessarily a good description of the space of possible methodologies. In the following sections, we discuss specific aspects of natural development that bear on both grammatical and cell chemistry approaches, expanding the perspective on AE in general. The goal is to identify the dimensions of development that can vary in the implementation of AE systems. We will then use this broader perspective to form a more useful taxonomy of systems than currently exists. This taxonomy will allow future AE research to take place in a unified context.

3 Dimensions of Development

In this section, we analyze natural development by breaking it down into five major dimensions. Not only are these dimensions conceptually useful for understanding development, but they also provide insight into the design of AE systems. We will use these dimensions later to form a taxonomy of AE systems. The taxonomy is based on the ways these dimensions can vary in different AE implementations.

Two important themes underlie the discussion in this section:

- In order to compare different AE systems, it is important to understand the contribution of specific aspects of development to the capacity to evolve complex systems. Thus, the distinction between grammatical and cell chemistry approaches is less important than the differences between specific systems on various dimensions of development.

- Some aspects of development can be implemented on computers as abstractions of biological development that would be impossible in nature. Using such abstractions can significantly improve efficiency.

Hence, identifying the dimensions of development can help in selecting the right combination of developmental mechanisms to produce a computationally efficient AE methodology, i.e. one that can produce phenotypes of extremely high complexity. We now turn to a detailed discussion of each of the following five dimensions of development:

1. **Cell Fate**: The fate of a cell is the eventual role it will come to play during development. For example, a cell may become a neuron or a muscle cell. There are several ways that cell fates are determined in nature. Since cells in AE systems must eventually play a role in the mature phenotype, it is important to consider the means through which those roles can be determined.

2. **Targeting**: The ways that cells can develop connections to target locations is an important aspect of development. Connectivity contributes to the overall functionality of complex systems, particularly neural networks. AE system design can benefit from an analysis of the ways that connections develop in nature.

3. **Heterochrony**: The timing and ordering of events in the embryogeny of a lineage of organisms can change over generations. Such changes can result in different final results, sometimes leading to important innovations in natural organisms. AE researchers may consider whether their encodings allow similar flexibility.

4. **Canalization**: Biological genomes are tolerant to mutations. Several mechanisms allow developing components to adjust to changes caused by mutations in connected components. These mechanisms can be employed in AE systems.

5. **Complexification**: Over the course of biological evolution, new genes are occasionally added to genomes, increasing the complexity of the phenotype. Complexification has led to major innovations in body-plan organization. By implementing a mechanism for handling variable length genomes, AE can also utilize complexification.

3.1 Cell Fate

The fate of a cell determines its ultimate location, connectivity with other cells, and role in the mature phenotype. A cell's fate is determined by the genes that are expressed during its development. For example, liver cells show a different pattern of gene expression than brain cells, even with the same genes in their genomes. AE systems must address a crucial question: How is cell fate determined? In grammatical encodings, the ultimate role of a specific structure is obtained from a grammatical derivation, whereas cells in cell chemistry approaches can derive their ultimate roles through chemical messages from other cells, or from activated genes inside the cell. In this section, we examine the variety of ways that cell fates are determined in nature, all of which can potentially be implemented in AE.

In biology, cell fates are determined through a variety of means. We illustrate these mechanisms through a set of examples, after introducing some terminology. Early in development, groups of cells are undifferentiated and called precursor cells. Their fates are assigned in several ways, frequently through receiving messages from an outside cell called an *organizer*. For illustration, let us assume the precursor cells each take on one of two fates: A or B (figure 6). A variety of static and signal-based strategies are possible (Wilkins 2002):

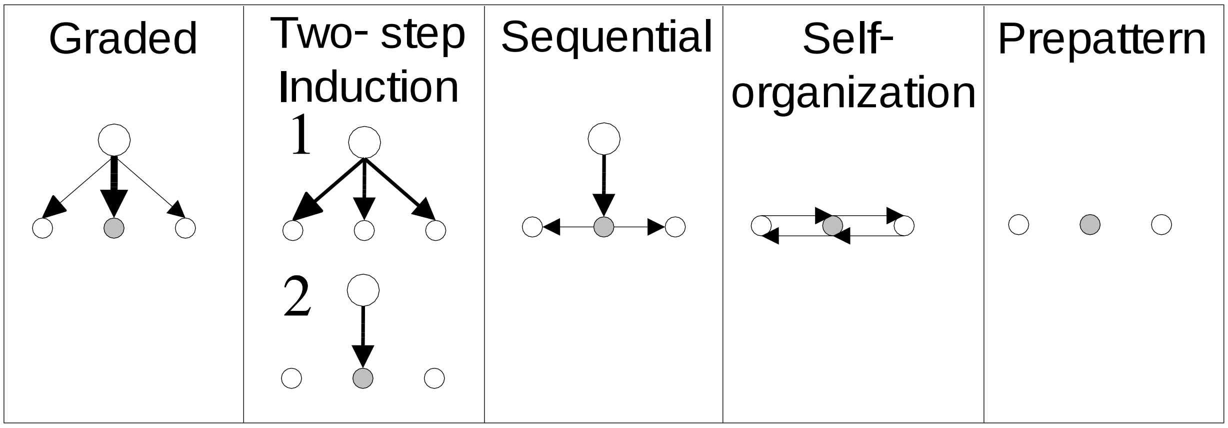

Figure 6: **Derivations of Cell Fates.** Five ways that cells can derive their roles are depicted. The large cell is an *organizer cell*, i.e. a cell that tells other cells what fates to assume. The smaller cells, which are initially undifferentiated, can become either cell type A (gray) or cell type B (white), through various mechanisms. A question for AE systems is which mechanisms lead to most efficient evolution? This figure is based on Wilkins (2002).

- **Graded Induction**: A signal is released from an organizer in a graded pattern. The precursor cell receiving the most signals assumes fate A, while those receiving fewer signals assume fate B.
- **Two Step Induction**: The organizer first releases a uniform signal to all the precursor cells to let them know that they form a precursor group. A secondary signal is subsequently released to a subset of the group in order to further distinguish fates.
- **Sequential**: The signal from the organizer reaches only a single precursor cell, which assumes fate A. It then sends secondary signals to adjacent precursor cells, giving them secondary identities.
- **Self-organization**: The precursor cells signal each other and the dynamic properties of the network of signals assigns fates, without any organizer cell necessary.
- **Prepattern**: Gene expression alone, deriving from each precursor cell's lineage, is responsible for fates. Signals are not utilized.

While in biology all of these strategies are utilized, any subset of them can be used to specify cell fate in an AE system. Grammatical systems tend to rely extensively on prepatterning, while cell chemistry approaches rely on a diverse set of signaling strategies. However, either approach can use *any* of the above strategies. For example, parameters or contexts in L-system rewrite rules could include signaling constraints, while cell chemistry approaches could rely extensively on prepatterning.

In addition to the above strategies, AE can also take advantage of positional information that is not explicitly available in nature. In embryos of many natural organisms, a gradient of protein expression along different axes forms a rudimentary Cartesian coordinate system. Because coordinate information is not explicitly available to developing cells, these gradients evolved to provide positional information. For example in vertebrates, the fates of cells along the anterior-posterior axis is based on proteins expressed by HOX genes (Curtis et al. 1995; Lall and Patel 2001) (see Section 3.5). However, in AE, coordinate information is known *prima facie* and therefore need not be evolved. Rules or genes can be made to explicitly activate based on the coordinates of their containing cells in a Cartesian system. In fact, Bentley and Kumar (1999) and Eggenberger (1997) used this idea by assuming a "gradient" present in every developing embryo that was really just a two-dimensional Cartesian coordinate. This way, positional information can be used directly to specify fate, or to regionalize the embryo into cell fate groups.

Because the information available computationally can differ from that available in nature, the design of AE systems should be based on a principled analysis of the pros and cons of the various natural and artificial strategies. For example, prepatterning is computationally inexpensive, yet used alone only allows a single rigid way of specifying identities. In contrast, two step induction and self-organization are dynamic approaches to fate determination, yet involve a great deal of computation. The extent to which the various strategies contribute to the development of highly complex systems is unknown.

Another question is how the cells arise in the first place, before they differentiate. In natural embryos, precursor cells must be created and distributed in an organized manner. The early proliferation of embryonic cells is called *cleavage* (Gilbert 2000, page 25), during which numerous cell divisions occur. When cell division begins to slow down, these early cells, called *blastomeres*, rearrange their positions by moving around the embryo in a process called *gastrulation* (Gilbert 2000, page 26). AE researchers must decide whether cleavage or gastrulation are the most computationally efficient approaches to start the process of embryogenesis. Proliferation may not be necessary at all in a computer because a canvas of undifferentiated cells can be created instantaneously with little computational cost. Thus, it may not be necessary for evolution to discover ways to initially spread the canvas through cleavage and gastrulation.

Similarly, even though natural nervous systems rely extensively on cell migration to place cells in their appropriate locations (Gans and Northcutt 1983), AE researchers must determine to what extent migration aids evolutionary search. One possibility is that migrating neurons enable evolution to express similar phenotypes in different ways through dynamically placing cells in their final positions. For example, because of migration, a phenotypic class (such as neurons) can be created all at once in a single location. Evolution can then produce spawning centers where a class of cells are created, and then place and connect those cells in different locations.

Even if cells are allowed to migrate, they can do so in many ways. In low-level implementations, cells would literally move through the developing structure as they do in natural embryos. However, AE cells could instead prespecify the ultimate positions of their progeny using either absolute or relative coordinates, in which case simulating migration would be unnecessary.

Much research remains to be done on distinguishing the factors in cell fate determination that evolved in response to physical constraints in nature from those that facilitate the evolutionary search process. We have surveyed the different mechanisms of cell fate determination in nature, and those that are additionally available computationally. Through implementing these mechanisms, experiments can be performed to determine how they contribute to developing complexity.

3.2 Targeting

Not only must cells reach their proper locations, but they must also accurately connect to other cells. Particularly in the nervous system, developing proper connectivity is crucial (Gans and Northcutt 1983). In order to create a working network of cells, cell extensions such as axons and dendrites must ultimately reach their intended targets.

Targeting of cell extensions is usually based on one of two primary strategies (Marin et al. 2002; Zigmond 1999, pages 527-528):

- **Specific Identity**: The identity of the target is specified directly in the genetic code and the target is located based on a chemical marker.

- **Relative Position**: The relative location of a target is specified in the genome, so that the extension grows a specific distance and angle in order to reach the target. Highly regular patterns of connectivity, such as neural self-organizing maps (SOMs), are likely to form this way.

Many variations of the two targeting strategies exist in biology, and can be utilized in AE. For example, a chemical gradient could be used to to identify a large target area and help processes find their goals. Location and identity could be used together, such that an extension grows towards a particular region and then uses a gradient to zero in on its target. Other cells can also emit path designating proteins, as in Dellaert and Beer's (1996) model.

Although many neural systems can conceivably be organized without specifying precise targets for individual extensions, at least some natural neural networks are based on very specific prespecified targeting. For example, dendritic targeting in the olfactory nervous system of *Drosophila* flies has been demonstrated to be prespecified (i.e. hardwired; Marin et al. 2002). Thus, it is reasonable for an AE system to be able to evolve both prespecified hardwired connection topologies and topologies that organize based on chemical gradients or relative locations.

The entire process of growing axons and dendrites need not be simulated in order to form neural connections in simulation. Connections can be formed instantaneously in AE systems with perfect accuracy. Whether low-level simulation of axon growth provides an evolutionary advantage remains to be demonstrated. In addition, as with cell migration, Cartesian coordinate information can be directly incorporated into AE cells in order to allow them to find appropriate target locations.

AE researchers also must consider the role of user-specified inputs and outputs in developing neural networks. How can a developing embryo connect its cells to user-specified cells that do not result from development? Some researchers avoid this problem by coevolving body morphology along with neural networks, so that inputs and outputs arise naturally from the body itself (Bongard and Paul 2000; Bongard and Pfeifer 2001; Hornby and Pollack 2001b; Komosinski and Rotaru-Varga 2001; Sims 1994). However, such coevolution may not always be possible in real-world domains such as the evolution of controllers for robots that have already been manufactured. Natural evolution did not need to address the problem of connecting user specified inputs and outputs to a developing embryo. Thus, a special facility to ensure that the final topology is connected may need to be implemented, and may affect the kinds of targeting allowed by an AE system.

The way that targets can be expressed through genetic encoding can profoundly bias evolutionary search towards certain kinds of topologies. Thus, significant research is necessary to determine both the most computationally efficient way to simulate targeting, as well as the most expressive encoding.

3.3 Heterochrony

Changes in the timing of developmental events over generations is called *heterochrony* (Gilbert 2000, page 554). In natural embryogeny, the path to the final product is surprisingly flexible. Entire phases of development can be eliminated without sacrificing the end result. For example, many frog species have evolved away their tadpole stage, yet still grow into mature, sexually functioning frogs (Ellinson 1987). Their limb buds develop early, and a little frog, rather than a tadpole, ultimately emerges from the egg. Conversely, the Mexican axolotl, a salamander, has lost its *adult* stage, and develops mature gonads as a tadpole-like creature (Voss and Shaffer 1997). Their development of gonads has been greatly accelerated so that they become sexually functional as tadpoles. This dramatic flexibility in development suggests that significant modularity underlies the genetic encoding. Because the timing of the development of different modules is so flexible, mutations can safely modify timing, allowing evolution to explore a variety of developmental

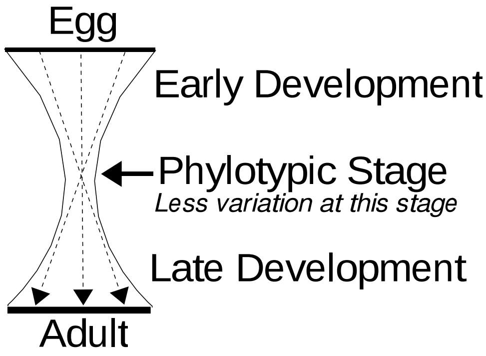

Figure 7: **The Developmental Hourglass (Raff 1996).** The hourglass represents the constraints on trajectories through developmental space. It illustrates that even in complex organisms, a great deal of change in developmental pathways is possible not only in late development, when established body parts are being refined, but also early in development, when the master body-plan is still being established. The phylotypic stage, where interactions between different developing components increase, is least amenable to change. Because global interaction between key components are critical to interconnecting the entire organism, timing changes during this stage could severely disrupt development. The three dotted lines depict different potential trajectories through the space, all of which cross identical phylotypic stages, even with different start and endpoints. This crossing illustrates that the phenotype at the phylotypic stage can remain constant even as early and late development vary in their timing and structure.

plans. Heterochrony allows developing components to come into contact with different components , so that evolution can explore many points of synthesis between components. Thus, AE researchers may wish to consider whether their encodings support heterochrony.

The ordering of developmental events is strikingly flexible. In *Drosophila* flies, all body segments form in parallel. However, in other flies, body segments form sequentially (Lawrence 1992). In both cases, the individual segments develop in the same way regardless of the order of their appearance. Thus, many components can appear at different relative times. Only when different components must communicate, such as when one component uses messages from another to guide its development, can timing changes have a deleterious effect.

Both early and late development exhibit developmental variation in nature. While changes in late development can be explained by local variations in mature components, the question remains how early development can change without altering the entire development of the organism. One explanation for the great deal of flexibility in early development is that global interactions among differentiated components of an embryo do not begin to occur until around the middle of embryogenesis. This pivotal middle period is called the *phylotypic stage* (Raff 1996, page 208) . After the phylotypic stage, most development is highly localized, giving precise definition to specific body parts. The cessation of global interaction allows a great deal of plasticity. This way, a picture of a *developmental hourglass* emerges, wherein trajectories of development can vary most early and late in development, but less in the middle (figure 7). In fact, Raff et al. (1999) demonstrated that early development is so flexible that they were able to mate two separate species of sea urchins, only one of which had a larval stage, and produce a viable hybrid offspring! The offspring reinstated the larval stage that was absent from its maternal parent, but developed differently than either parent. Thus, early developmental modules can be very flexible in their interactions.

There are several ways heterochrony can positively affect the final phenotype. For example, body (or neural) parts could be moved or manipulated in relation to each other by altering the timing and sequence of developmental. Parts can also be extended or compressed. Indeed, heterochrony has been credited with the evolution of single-toed horses from five toed horses (Wolpert 1987) and the movement of the nose of the whale to the top of its skull (Slijper 1962). In the evolution of the single-toed horse, the growth rate of the central toe of the five toed horse sped up so much relative to the other toes that the other toes ultimately became irrelevant and disappeared. In the case of the whale, an extension in the duration of jaw growth caused the nose to be pushed upwards to form a blowhole. Thus, heterochrony has led to important phenotypic discoveries.

In addition, heterochrony increases the number of successful genotypes by offering a variety of paths to each successful phenotype. Mutations that can cause timing changes without disrupting the final product are *neutral* mutations. There exist neutral networks of genotypes, connected by neutral mutations, that can be explored without fitness cost, increasing the space that can easily be explored by evolution. In other words, changes in timing can affect the developing parts that come in contact with each other, and the places of contact, without altering the final product. Thus, given both the positive effect heterochrony can have on final phenotypes, and the increased potential for neutral evolution, it is worthwhile to consider whether temporal changes are possible in a given AE system.

What kinds of artificial encodings allow heterochrony? Transcriptional factors that regulate genes control gene expression during development. The overall timing of development results from the precise sequence of genes expressed and from the the combinatorial properties of transcription factors that regulate genes (Ambros 2002; Reinhart et al. 2000). Whether AE systems approximate the combinatorial properties of natural gene regulatory networks may determine the extent to which they support heterochrony. Some cell chemistry researchers have already begun to explore this area. For example, Bongard (2002) showed neural and morphological development in evolved artificial organisms could each vary independently.

In addition, computational abstractions can be utilized that are not available in nature. Like Cartesian coordinates being used in lieu of chemical gradients, time itself can be used to regulate genes. Since time is directly available, it can be exploited as a ”growth hormone” that explicitly activates and terminates events in embryogeny. Grammatical encodings can be parameterized by time, while cell chemistry approaches may implicitly or explicitly take time into account regulating genes. Thus, either kind of encoding can potentially use time to regulate steps in development.

Biologists continue to uncover the mechanisms that make heterochrony possible. It remains to be seen how difficult it is to achieve similar flexibility with either grammatical or cell chemistry approaches.

3.4 Canalization

Genetic algorithm encodings are notoriously brittle: A significant proportion of mutation leads to infeasible offspring. In contrast, natural organisms are more robust to mutation. Natural development seems to be buffered so that slight changes in critical components do not cause related or connected components to fail. This robustness to mutations is called *canalization* (Waddington 1942)[3]. Canalization may be an important property for an AE system, facilitating safe exploration of genotypic space. How does canalization work, and what might be the mechanisms that allow development to succeed even when under the stress of unpredictable changes among dependent parts?

We will discuss three important ways that nature buffers against changes in expression:

- **Stochasticity**: Key events in development can be nondeterministic at the single cell level, instead relying on self-organizing mechanisms over groups of cells. Thus, the loss or gain of cells does not

[3]The term *canalization* was chosen by Waddington (1942) to form an analogy between the way water running down a hill eventually carves out regular streams in the surface, and the way development slowly settles on a set of conventions that become ingrained in the genome. Although changes to the geological environment or vegetation may cause water to take different paths, the set of available paths is very difficult to alter once canalization has occurred.

throw off the process of development.

- **Resource Allocation** The availability of resources to cells or groups of cells impacts the final phenotype. Body parts share finite available resources. Thus, if a mutation causes one component to consume less resources than it used to, other components can automatically consume more resources (e.g. by growing) to make room for the change, without requiring complementary mutations of their own.
- **Overproduction** A large amount of cells can be produced and later eliminated if they turn out unnecessary.

3.4.1 Stochasticity

After a cell first appears, its fate is not always realized deterministically (Alberch 1987). In many cases, cell fates cannot be predicted even with complete genetic knowledge. For example, several cells in early nematode embryogenesis are candidates for vulva formation. The cell that is ultimately chosen to be the central cell around which the vulva forms is the cell that produces the most of a special ligand. Receiving cells register the ligand production of their neighbors and create more ligand receptors the more ligand they receive. This creation decreases their own ligand production. Thus, a single cell ultimately emerges as the primary cell around which vulva formation will center (Sigrist and Sommer 1999; Wilkins 2002).

It does not matter if mutations shift the ligand production one way or another, since there is a mechanism around which the entire process will organize in *any* case. Such robustness may turn out to benecessary or useful to AE systems as well. Therefore, experimentation with stochastic grammatical rules and artificial regulatory networks may give insight into how AE can adjust to mutations in similar ways.

3.4.2 Resource Allocation

Recent research into developmental allocation of resources has confirmed a surprising result: Body parts, at the level of appendages, can reallocate the distribution of resources. Nijhout and Emlen (1998) manipulated the body parts of developing butterflies and beetles using two techniques: amputation, where a limb was removed, and hormone treatment and selection, where limbs were only reduced. In all cases they found that when one appendage consumed less resources, another appendage became larger by consuming the remaining resources. This result implies that compensatory mechanisms are genetically encoded and can react and reallocate resources properly if a mutation leads to a change in body proportions. Thus, a change in the size of one appendage does not require evolving compensatory changes in other appendages at the same time.

It is possible that a resource allocation scheme could be used in AE for similar flexibility. One major difference between AE and natural embryogeny is that the analog of nutrient resources in AE are essentially unlimited. However, a resource allocation scheme could still help maintain reasonable topological constraints on phenotypes after significant mutational changes.

3.4.3 Overproduction

Neural connectivity can potentially be disrupted by mutations affecting only some neurons in a network. If a group of neurons A connects to another group B, then a mutation only affecting A may be enough to break the entire circuit involving A and B. Such changes are called *non-concordant*. In contrast, mutations that effect both A and B in complementary ways are called *concordant* changes. While non-concordant changes are potentially devastating, the problem can be avoided if B has a buffering mechanism that enables it to react to a wide variety of possible configurations of A.

One such buffering mechanism is overproducing neurons in B. If the number of connections extending from A to B is reduced or increased through mutation, the extraneous cells produced by B can easily be disposed of through apoptosis, or planned cell death. In either case, the functional connection between A and B will develop.

In fact, overproduction as a buffering mechanism has been documented in the connections between a cat's eyes and brain (Williams et al. 1986, 1993). European wildcats evolved to live in cold environments and hunted during the day, while North African cats, from which domestic cats descend, hunted at night. Because requirements are different for daylight hunting, wildcats have significantly more light sensors, or cones, in the center of their visual field, resulting in an optic nerve with more neurons sending signals back to the lateral geniculate nucleus (LGN), which is the part of the brain receiving the connections. Interestingly, fetal wildcats have the same number of axons as domestic cats in their optic nerve in early development. However, in domestic cats, many of the cells that are initially produced end up dying, while significantly fewer cells die in the wildcat (Williams et al. 1993). Furthermore, studies of hybrid cats revealed that the number of axons in their optic nerve significantly varies. Despite this variance, the number of cells in the LGN always matches the number of neurons in the optic nerve, indicating that cell death is being used as a buffering mechanism by the genetic system that produces the optic nerve. Thus, whether the size of the optic nerve increases or decreases from one generation to the next, the LGN can always provide the right number of connections for the system to function.

How could overproduction work in AE? In section 3.2, we discussed targeting by either specific identity or relative position. If specific identity alone were used in an AE system, overproduction could not be utilized. However, if targets that might not exist can be specified, connections to such targets would be removed, utilizing overproduction. In contrast, overproduction with relative positional targeting is relatively easy to implement. If an axon targets an empty location, it can be discarded. Some AE systems already remove unused axons that result from overproduction in order to allow relative targeting to be imprecise (Cangelosi et al. 1993; Nolfi and Parisi 1991).

All three imprecision-based strategies, that is stochasticity, competition, and overproduction, are related and overlapping in their application in nature. Perhaps they would allow a safer search through genotypic space in AE as well.

3.5 Complexification

While the preceding discussion of natural mechanisms in embryogeny has focused on the process of development itself, it is important also to consider how genetic recombination and mutation are leveraged in the evolution of developing systems. In particular, mutation in nature goes beyond optimization. New genes are occasionally added to the genome, allowing evolution to perform a *complexifying* function over and above optimization. Complexification allows evolution to begin with simple systems and elaborate on them incrementally, as opposed to evolving elaborate systems from the start. Furthermore, elaboration is protected in nature in that interspecific mating is prohibited. Such speciation creates important dynamics differing from standard GAs. In this section, we discuss how these important characteristics of natural evolution bear on embryogeny.

Gene duplication is a special kind of mutation in which one or more parental genes are copied into an offspring's genome more than once. The offspring then has redundant genes expressing the same proteins (figure 8). Gene duplication has been responsible for key innovations in overall body morphology over the course of natural evolution (Amores et al. 1998; Carroll 1995; Force et al. 1999; Martin 1999).

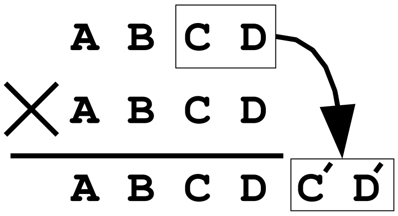

Figure 8: **Gene Duplication.** Two genomes of equal length are crossed over. The letters represent the trait expressed by each gene. The offspring has two additional redundant genes, resulting from the duplication of genes C and D from the first parent.

A major gene duplication event occurred around the time that vertebrates separated from invertebrates. The evidence for this duplication centers around *HOX genes*, which determine the fate of cells along the anterior-posterior axis of embryos. HOX genes are crucial in shaping the overall pattern of developmental in embryos. In fact, differences in HOX gene regulation explain a great deal of arthropod and tetrapod diversity (Carroll 1995). Amores et al. (1998) argue that since invertebrates have a single HOX cluster while vertebrates have four, cluster duplication must have significantly contributed to elaborations in vertebrate body-plans. The additional HOX genes took on new regulatory roles in vertebrate anterior-posterior axis development, considerably increasing body-plan complexity. Although Martin (1999) argues that the additional clusters can be explained by many single gene duplications accumulating over generations, as opposed to massive whole-genome duplications, researchers agree that gene duplication has contributed to important body-plan elaborations.

A detailed account of how duplicate genes can take on novel roles was given by Force et al. (1999): Base pair mutations in the generations following duplication *partition* the initially redundant regulatory roles of genes into separate classes. Thus, the embryo develops in the same way, but the genes that determine overall body-plan are confined to more specific roles, since there are more of them. The partitioning phase completes when redundant clusters of genes are separated enough that they no longer produce identical proteins at the same time. After partitioning, mutations within the duplicated cluster of genes alter different steps in development than mutations within the original cluster. In other words, the opportunities for mutation increase through duplication because duplication creates more points at which mutations can occur. In this way, developmental processes elaborate.

Because major biological shifts in body-plan complexity have resulted from duplication, AE systems should be able to utilize this kind of mutation as well. However, duplication is difficult to implement for two reasons:

1. If genes are allowed to duplicate over the course of evolution, the population will contain genomes of variable length. Variable length genomes can cause problems for crossover in GAs.

2. Two identical sets of genes must be capable of diverging into separate roles in a given AE system. It is difficult to ensure that new genes remain useful after their first appearance.

We now discuss how these obstacles have been overcome by nature, and how AE can employ similar solutions.

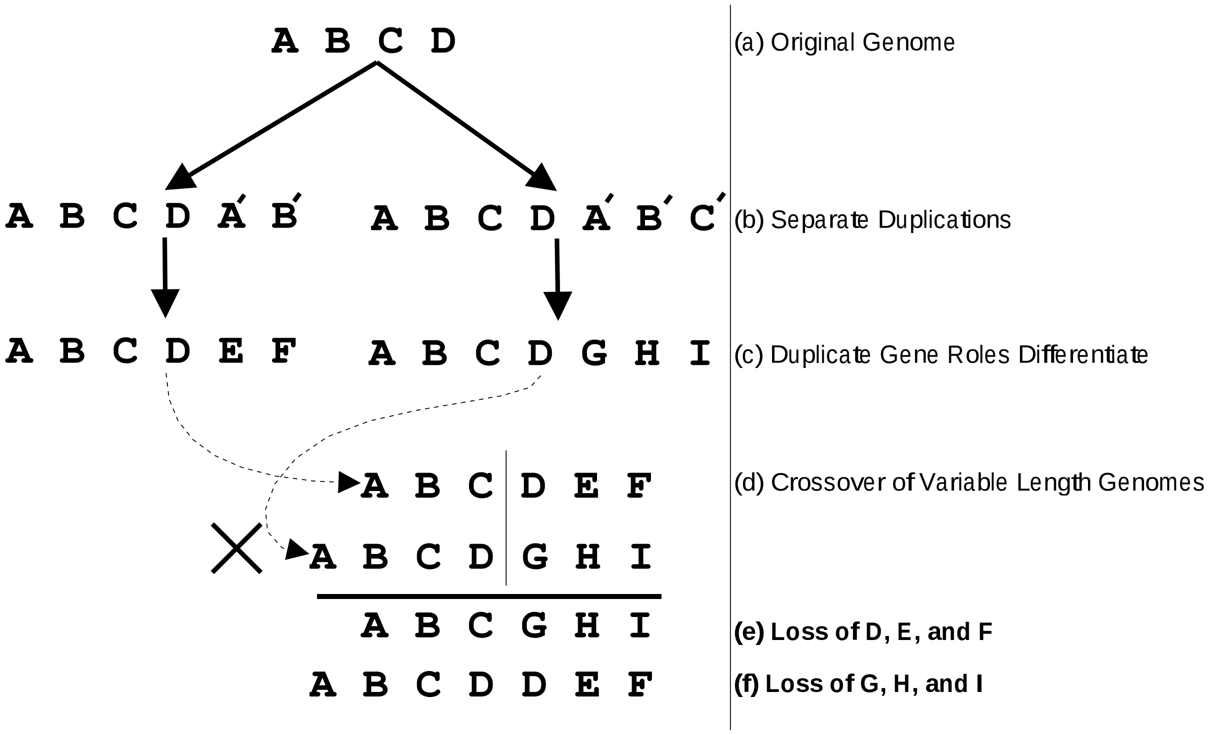

Figure 9: **The Variable Length Genome Problem.** Variable length genomes are necessary for complexification. The diagram shows how critical genes can be lost in the crossover of such genomes. A sequence of events is depicted from top to bottom (a through f). (a) The original genome contains four genes, A, B, C, and D. (b) In separate instances of reproduction, the original genome undergoes two different duplications. In one case a cluster of two of its genes, A and B, are duplicated. In the other case, three genes, A, B, and C, are duplicated. The resultant genomes now have differing lengths. (c) Over generations, the roles of the duplicated genes differentiate from their redundant original roles. This functional divergence is represented using different letters, E and F, and G, H, and I. The newly differentiated genes now may serve important new roles in their respective lineages. (d) The two genomes of differing length are crossed over. (e) In one possible offspring, maintaining the length of the smaller genome, genes D, E, and F are lost. Particularly troublesome is the loss of D, which was in the original genome. (f) Another potential crossover, preserving the length of the larger genome, loses G, H, and I, along with duplicating D. Loss of information in crossover may disrupt previously stable phenotypic development. Moreover, there is no way to ensure that all the necessary genes are included without a mechanism for checking which genes from one genome correspond to those from another. See figure 10 for a solution to this problem.

3.5.1 Variable Length Genomes in AE

A gene in AE might be a rewrite rule or a gene product with a regulatory region. Whatever the encoding, when duplication is allowed, the number of genes is variable, which can cause loss of information during crossover. Figure 9 shows that as new genes are added in different lineages through different duplications, the same gene may exist at different positions in different genomes. Conversely, different genes may exist at the same position. Thus, crossover may lose essential genes through misalignment.

Interestingly, this problem does not occur in nature, and therefore it should be possible to avoid it also in AE. First, organisms with significantly different genomes never mate because they are in different species.

Second, nature has a mechanism for aligning genes with their proper counterparts during crossover, so that data is not lost nor obscured. This alignment process has been most clearly observed in *E. coli* (Radding

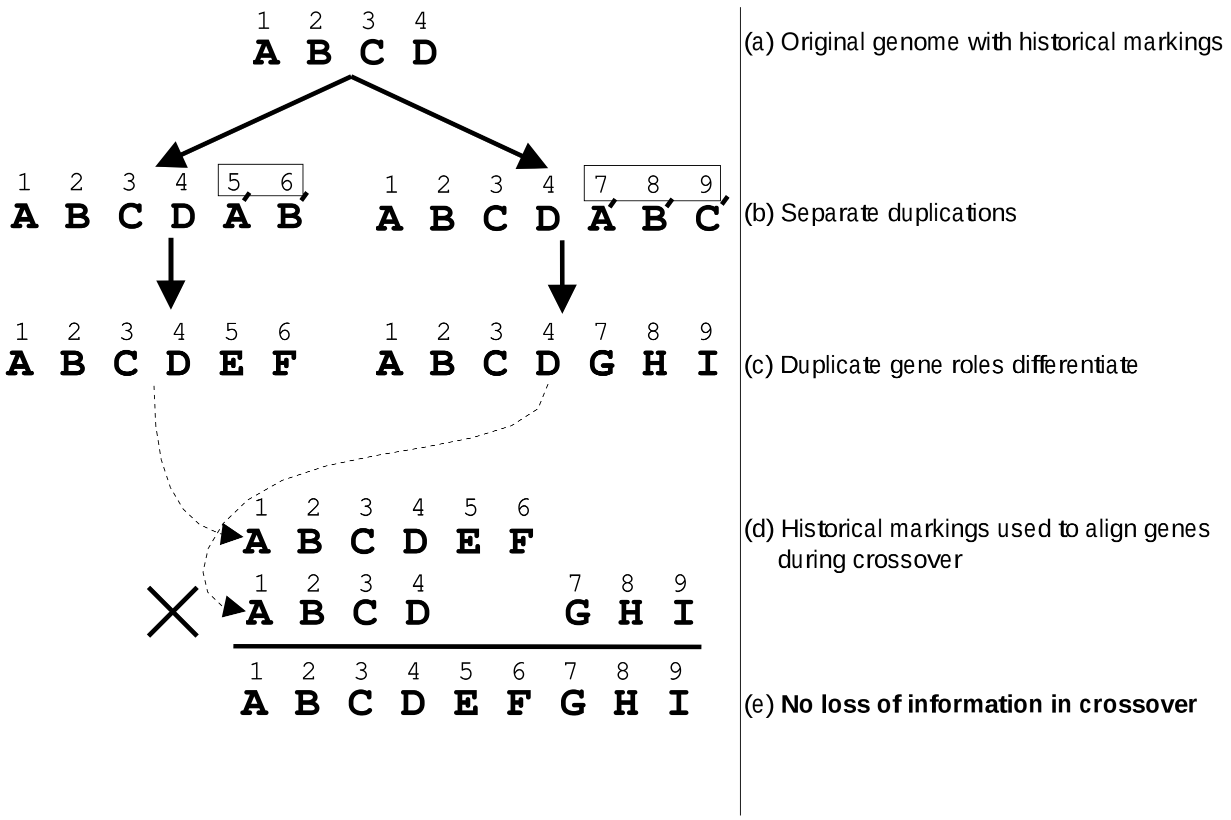

Figure 10: **Solving the Variable Length Genome Problem with Historical Markings.** Historical markings are numbers assigned to each gene that represent the order in which new genes appeared over evolution. (a) The original genome contains four genes, A, B, C, and D, assigned historical markings 1 through 4. (b) When new genes appear through duplication, they are assigned numbers in the order in which they appear. Assuming the duplication on the left happened before the one on the right, the new genes, A' and B', and A', B', and C', are assigned the numbers 5 through 9. (c) As the products of the duplicate genes differentiate, their historical markings continue to serve as a record of their origins. (d) During crossover, those genes that have matching historical markings are aligned, while those that are disjoint are purposefully not aligned. (e) The result is that any kind of crossover can preserve the information and relationships between all the genes in variable length genomes by utilizing the historical markings. Historical markings are an abstraction of synapsis, the process used in nature to match up alleles of the same trait during crossover (Radding 1982; Sigal and Alberts 1972).

1982; Sigal and Alberts 1972). A special protein called *RecA* takes a single strand of DNA and aligns it with another strand by attaching the strands at genes that express the same traits, which are called *homologous genes*. The process by which RecA protein aligns homologous genes is called *synapsis*. In experiments in vitro, researchers have found that RecA protein does not complete the process of synapsis on fragments of DNA that are not homologous (Radding 1982).

It turns out that speciation and synapsis can both be utilized in genetic algorithms, including AE, by using features of evolution that are available only through computational means. Stanley and Miikkulainen (2002a,b,c) showed that the ancestral *history* of genes in an evolving population can be used to tell which genes should line up with which during crossover. If a counter assigns increasing integers to new genes every time they appear through a mutation, and if those integers are preserved every time genes are subsequently inherited, then the origin of every gene is known throughout evolution. The numbers assigned to each gene are called *historical markings*. Since two genes with the same origin must express the same trait, it is possible to know exactly which genes line up using the historical markings (figure 10).

Although the system developed by Stanley and Miikkulainen was applied to a direct encoding of neural networks that did not include a developmental stage, in principle historical markings can be applied to any genetic encoding, including those used in AE, since gene history is a genetic property in both developmental and nondevelopmental encodings. Thus, using the NeuroEvolution of Augmenting Topologies (NEAT) methodology, the variable length genome problem can be overcome in AE.

Stanley and Miikkulainen (2002b,c) also showed that historical markings can be used to speciate the population, thereby protecting incompatible organisms in different niches. The extent to which two genomes have different genetic histories is a measure of their incompatibility. Therefore, historical markings allow a simple way to test whether two genomes belong in the same species. The number of historical markings only present in one of the two genomes is a measure of their incompatibility. This measure can be used to cluster genomes into compatibility groups, or species.

Historical markings provide another example of a computational abstraction that is potentially more efficient than the natural process: In this case, historical markings are an abstraction of synapsis, which is the process through which homologous genes in natural genomes are aligned (Radding 1982; Sigal and Alberts 1972). Historical markings enhance evolutionary search, making it possible for evolution to utilize gene duplication effectively.

3.5.2 Divergence of Gene Clusters

Even if variable length genomes are possible in AE, the question remains how clusters of genes can be duplicated such that the new cluster eventually takes on a new role. This goal is important because major mutations in new duplicate clusters could permanently disabled them. They might never activate, or their products could become so different from those of the original cluster that they disrupt development. As Force et al. (1999) explained, after a cluster of genes is duplicated in nature, subsequent mutations repartition the roles of both the original genes and the duplicated genes without significantly altering the overall developmental plan. Once duplicate genes have undergone sufficient mutation to be activated at different points during development than their original counterparts, subsequent mutations can begin to alter development at these new points. Thus, because of the duplicate genes, evolution has the flexibility to alter the developmental process at additional points.

If duplicate genes are to take on new roles in development, they must be carefully integrated into the already-existing developmental plan of the organism. If mutations that change the conditions under which duplicate genes are activated are too severe, the genes will become disconnected from the existing developmental plan, and subsequent mutations will likely have little effect. Thus, in order to allow duplicate genes to gradually take on new roles, the conditions under which they activate should lie on a continuum. In other words, a slight mutation in one duplicate should cause it to be activated in some but not all cases where its counterpart was formerly always activated. Care must be taken to make such changes possible in a particular AE encoding. Preconditions for activating grammatical rules or regulatory requirements for genes in cell chemistry systems should not allow slight mutations to significantly alter the conditions under which a particular gene is activated.

AE systems that implement both synapsis and gradual divergence of duplicate genes will allow researchers to experiment with gene duplication.[4] Because duplication is the primary means of complexification, genomes should start out small and be allowed to become larger through duplication. Each duplication lifts the genome into a higher dimensional space where more genes are being optimized, increasing the

[4] *Gene deletion* is also possible. However, deletion is potentially more deleterious than duplication. Duplication creates redundancy, which does not cause any loss of functionality. In contrast, deletion may cause important steps in development to be removed, short-circuiting embryogeny.

potential complexity of the phenotype. Thus, complexity builds up incrementally from a minimal starting point. This approach allows evolution to optimize the lowest possible number of dimensions, since organisms with new genes only do better if those genes are useful. Speciation allows different species to complexify at different rates, so that the population can explore different spaces simultaneously (Stanley and Miikkulainen 2002a,c). AE systems that complexify genomes in different species from a minimal starting point will allow researchers to address an important question: How should clusters of genes be chosen for duplication? Everything from copying single genes to duplicating whole genomes is possible. While biologists continue to debate this issue (Amores et al. 1998; Martin 1999), AE can also begin to address it through experimentation.

A complexifying AE system that starts with small, simple genomes will first evolve basic structures, such as bilateral symmetry, and then elaborate on them in future generations by adding new genes. This approach is more likely to discover highly complex phenotypes than an approach that begins searching in the intractably large search space of complete solutions.

We have now surveyed all five major dimensions of development in biological systems: Cell Fate, Targeting, Heterochrony, Canalization, and Complexification. Since each dimension can be implemented in different ways in AE, the dimensions together represent the space of possible AE systems. Thus, the distinction between cell chemistry and grammatical approaches reveals less about the underlying properties of AE systems than how they operate on each dimension.

Another important theme from this section was computational abstraction of processes in natural development that may be more efficient than nature's own methods. The following resources that are not directly available to nature but useful in computational systems were identified:

- Cartesian Coordinates to represent position (Sections 3.1 and 3.2)
- Instantaneous spreading of a canvas of cells (Section 3.1)
- Real time as a regulator of gene expression (Section 3.3)
- Historical markings as a mechanism for artificial synapsis (Section 3.5)

These artificial information sources can boost computational performance in various dimensions, and should also be considered in experimentation.

The next section uses the five dimensions of development to compose a taxonomy of existing and future AE systems, and suggests how the taxonomy can help focus future research efforts.

4 A New Taxonomy for Artificial Embryogeny

The goal of a taxonomy is to make classification and comparison of different systems possible. To achieve this goal, a good taxonomy must reflect the underlying properties of the space being classified. Thus, the new AE taxonomy will replace the older grammatical/cell chemistry distinction by employing each dimension of development as an axis in a multidimensional classification space. In order for dimensions of development to serve as axes, they must be characterized as continuums. That way, it will be possible to describe a particular AE system, as well as natural evolution, as *points* in a five-dimensional space of possible evolvable developmental systems. To better understand this space, we begin with a summary of the major dimensions and their properties:

1. **Cell Fate**: An AE system can range from having a single method for determining the fate of a cell (for example, prepatterning) to having a large variety of determination methods (described in Section 3.1).

2. **Targeting**: At one extreme, only specific targeting is used, whereas only relative targeting is utilized at the other extreme. In between, systems use some combination of the two strategies (Section 3.2).

3. **Heterochrony**: In simple systems, development may be inflexible, such that slight changes in relative timing are either not possible or would destabilize the entire process. More flexible systems approach the development hourglass, with many possible paths from early to late development (Section 3.3). [5]

4. **Canalization**: AE systems can range between requiring precise genotypic instructions to those that tolerate a high degree of imprecision or mutation, utilizing such strategies as stochasticity, resource allocation, and overproduction (Section 3.4).

5. **Complexification**: Classical GA encodings employ fixed-size genomes. On the other extreme, genomes in nature have variable length and sometimes new genes are added through duplications. Synapsis and speciation facilitate variable length genomes (Section 3.5).[6]

Figure 11 classifies AE systems from Section 2 along these dimensions. Only systems for which evolutionary results have been reported are classified. Natural evolution is also graphed (depicted as a tree) for comparison. The classification is *not* meant as a rating; There is no implied superiority of one end of a dimension over another. Rather, the classification is meant to show which areas of the space are currently being explored, and which remain uncharted.

The taxonomy makes it possible to ask design questions in a unified context. For example, can a system with high heterochrony and low canalization evolve complex phenotypes? Or, does specific targeting interfere with canalization? The answers to such questions will come from exploring the taxonomic space. Then, it will be possible to systematically construct better AE systems.

Already, several important conclusions can be drawn from the current classification. First, a large amount of space on many dimensions has not been explored. Unexplored space includes systems that employ many ways to determine cell fates, mix relative and specific targeting, have high potential for heterochrony, high potential for canalization, and realistic complexification. Interestingly, natural evolution occupies exactly this part of the AE space. This observation suggests that an extremely important point in this vast space remains to be explored, along with many other untested points.

Second, figure 11 shows that cell chemistry and grammatical approaches are more distinguishable on some dimensions than others. Cell chemistry approaches tend to use relative targeting while grammatical approaches prefer specific targeting, since rewrite rules and graph instructions tend to encode both sources and targets directly. Grammatical approaches also have low potential for canalization because of their use of specific targeting; Cell chemistry systems can utilize imprecise targeting since physical location is a standard property of their implementation. However, both methods are theoretically capable of implementing any particular kind of targeting or imprecision-based strategy, and such a division exists primarily for historical reasons. In contrast, the grammatical and cell chemistry approaches do not differ significantly in the cell fate,

[5] Ideally, empirical results would measure the potential of an AE system to support heterochrony. Because few such results are currently available, we estimate heterochronic potential by considering whether mechanisms of timing (e.g. counters, parameters in L-systems, or dynamic regulatory networks) are implemented in each system.

[6] No AE systems currently implement synapsis and speciation, or any replacement thereof. Therefore, to get distinctions in this dimension among existing systems, we considered whether genomes could be variable in length, and if so, whether their crossover operator differed from standard point crossover used in fixed-length genome GAs.

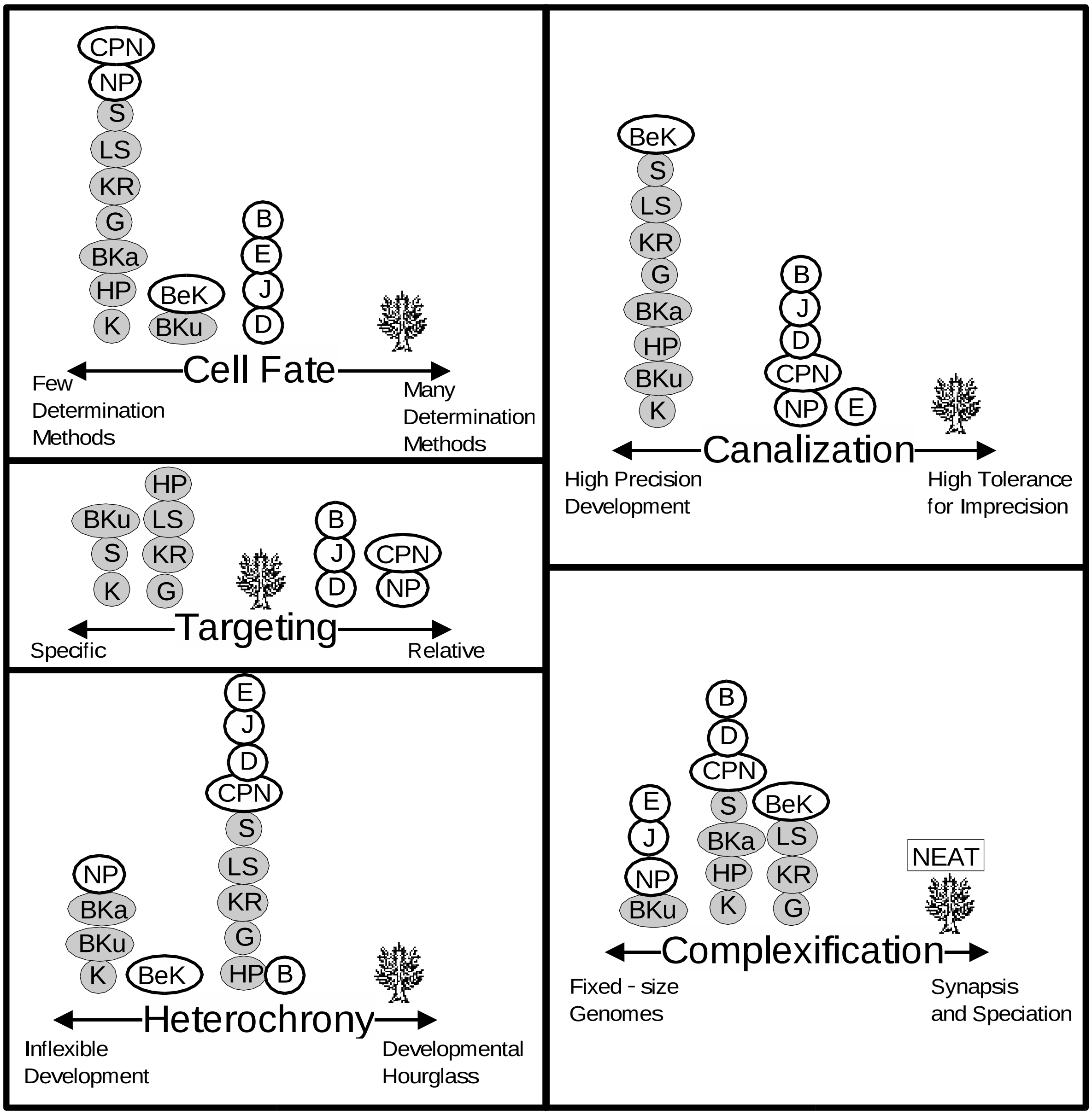

Figure 11: **The Space of Existing AE Systems.** AE systems, in addition to natural evolution (depicted as a tree), are graphed on each of the five dimensions of development laid out horizontally in each subfigure. Systems depicted in gray are grammar-based, while those in white utilize cell chemistry techniques. Only those systems for which evolutionary results were reported are included. Even though it is not a developmental system, NeuroEvolution of Augmenting Topologies (NEAT; Stanley and Miikkulainen 2002a,c) is included on the complexification dimension since it is the only system that currently implements a version of synapsis and speciation. The letters are abbreviations for the authors who developed the specific AE systems, as described in Section 2: B - Bongard and Paul (2000), Bongard and Pfeifer (2001), and Bongard (2002); BeK - Bentley and Kumar (1999); BKa - Belew and Kammeyer (1993); BKu - Boers and Kuiper (1992); CPN - Cangelosi et al. (1993); D - Dellaert and Beer (1994a,b) and Dellaert (1995); E - Eggenberger (1997); G - Gruau (1993), and Gruau (1994), and Gruau et al. (1996); HP - Hornby and Pollack (2001a,b); J - Jakobi (1995); K - Kitano (1990); KR - Komosinski and Rotaru-Varga (2001); LS - Luke and Spector (1996); NP - Nolfi and Parisi (1991); S - Sims (1994); Those systems that did not include any kind of targeting are not graphed on that dimension. The figure shows the kinds of AE systems that have been implemented, and those areas that remain unexplored. This taxonomy makes possible principled exploration and comparison of future systems.

heterochrony, or complexification dimensions. Thus, when comparing one AE system to another, it makes more sense to ask questions about dimensions in the taxonomy than about cell chemistry vs. grammatical approaches.

Third, the points that have been explored have rarely if ever been compared experimentally. The goal is to predict the ramifications of implementation decisions. Thus, it is not sufficient to merely explore a point in the space through a single isolated experiment. Rather, it will be necessary to choose important benchmarks and compare their results across points covering much of the space. The resulting data will characterize the entire space, allowing us to to predict what the effects of particular implementation decisions are on evolving complex phenotypes. Thus, the new taxonomy provides a principled framework within which future research can take place.

Eventually, the taxonomy will make possible principled design of AE systems. Informed implementation decisions on each dimension can be combined to form effective systems.

5 Discussion and Future Work

Ultimately, the goal of AE is to be able to evolve phenotypes as complex as biological systems. This goal is still far in the future; At the current stage, we are still searching for constituents that make AE effective. The dimensions of the new taxonomy for AE indicate what some of these constituents might be. The space of possible systems along these dimensions is still largely unexplored. There may not be one best solution: Different parameterizations may be good for evolving different kinds of complex systems. For example, an AE system suited for evolving million-neuron neural networks may not be the same as one suited for evolving vehicle architectures.

Thus, the challenge for future research is to comprehensively explore this massive space. We must find out what the trade-offs are between flexibility and simplicity, and we must question whether the mechanisms that biologists have identified as instrumental to natural development are equally viable in AE. As a first step in this exploration, we conclude this paper with several suggested benchmarks that may serve as a starting point for understanding the AE space. Each benchmark can be applied at many points in the taxonomic space, providing many possible avenues for future work.

- **Evolution of pure symmetry**: Symmetry is a significant means of reuse. When the same structures exist on both sides of an organism, discovering them only once in the genome as opposed to twice reduces search effort. Therefore, it is important to understand how the various dimensions of development can enhance the evolution of symmetrical patterns without any goal other than symmetry itself. If evolving k-fold symmetrical patterns is made the only goal, we may isolate those dimensions that facilitate gene reuse. Another interesting question is whether k-fold symmetrical patterns can easily be evolved into $k + 1$-fold symmetrical patterns, providing insight into the power of reuse in a particular implementation. Eggenberger's (1997) experiments in evolving bilateral symmetry represents a first step in this direction.

- **Evolving a specific shape**: How hard is it for various AE systems to evolve specific shapes that are known to be useful in nature or engineering? Understanding why systems at different points in AE space succeed or fail to evolve specific shapes will aid in making future implementation decisions.

- **Evolving specific connectivity patterns**: The targeting dimension (section 3.2) is crucial in neuroevolution and experiments should be devised to understand it thoroughly. It is of such importance that it may be useful to study it in isolation. How hard is it to get topologies such as feedforward, recurrent, or self-organizing maps to arise in AE? When is specific identity targeting most useful as opposed to relative location targeting? How important is neuron positioning during development when final connectivity is the only fitness criteria?

- **Evolving a simple controller**: While much AE research has focused on ambitious artificial creatures complete with body and brain, at some point it will be necessary to compare AE systems with other reinforcement learning systems (including non-developmental neuroevolution systems) on standard benchmark problems in order to asses whether particular AE approaches can evolve solutions to problems that non-developmental systems cannot. For example, pole balancing has been used as a reinforcement learning benchmark for over 30 years (Anderson 1989; Barto et al. 1983; Gomez and Miikkulainen 1999; Gruau et al. 1996; Michie and Chambers 1968; Moriarty and Miikkulainen 1996; Saravanan and Fogel 1995; Stanley and Miikkulainen 2002c; Watkins and Dayan 1992; Whitley et al. 1993; Wieland 1991, 1990), making it convenient for comparison with other methods. If a particular AE methodology cannot compete in a relatively simple domain, it may not be appropriate for evolving *more* complex artificial organisms. Using performance benchmarks will allow researchers to determine which AE methodologies are actually efficient enough to potentially achieve the long term goal of evolving highly complex structures.

6 Conclusion

The goal of AE is to eventually evolve systems that rival the complexity seen in natural organisms. While current AE systems represent a step in this direction, no artificial system has yet come close to the power of natural evolution. Thus, a principled approach to building AE systems is needed. As a first step, a framework is needed in which different implementations can be compared and contrasted along different dimensions. In this paper, we provided such a comparative framework. We proposed a new taxonomy for AE systems based on the dimensions of development seen in nature. This taxonomy suggests that the existing distinction between grammatical and cell chemistry approaches is superficial. Rather, the dimensions of development define the capabilities of an AE system. Using the new taxonomy, it will be possible to make principled design decisions in any kind of encoding, and to compare and contrast systems in the same context. Ultimately, we hope researchers can use this taxonomy to predict how varying the settings of different dimensions affects the capabilities of different implementations, and therefore build better AE systems.

Acknowledgments

This research was supported in part by the National Science Foundation under grant IIS-0083776 and by the Texas Higher Education Coordinating Board under grant ARP-003658-476-2001. Special thanks to Dr. Gary Freeman and Dr. David Parichy for discussions on biological evolution and development.

References

Alberch, P. (1987). Evolution of a developmental process: Irreversibility and redundancy in amphibian metamorphosis. In Raff, R. A., and Raff, E. C., editors, *Development as an Evolutionary Process*, 23–40. New York: Alan R. Liss.

Ambros, V. (2002). A hierarchy of regulatory genes controls a larva-to-adult developmental switch in C. elegans. *Cell*, 57:40–57.

Amores, A., Force, A., Yan, Y.-L., Joly, L., Amemiya, C., Fritz, A., Ho, R. K., Langeland, J., Prince, V., Wang, Y.-L., Westerfield, M., Ekker, M., and Postlethwait, J. H. (1998). Zebrafish HOX clusters and vertebrate genome evolution. *Science*, 282:1711–1784.

Anderson, C. W. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9:31–37.

Astor, J., and Adami, C. (2000). A developmental model for the evolution of artificial neural networks. *Artificial Life*, 6(3):189–218.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:834–846.

Belew, R. K., and Kammeyer, T. E. (1993). Evolving aesthetic sorting networks using developmental grammars. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Francisco, CA: Morgan Kaufmann.

Bentley, P. J., and Kumar, S. (1999). The ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, 35–43. San Francisco, CA: Morgan Kaufmann.

Boers, E. J., and Kuiper, H. (1992). *Biological Metaphors and the Design of Modular Artificial Neural Networks*. Master's thesis, Departments of Computer Science and Experimental and Theoretical Psychology at Leiden University, The Netherlands.

Bongard, J. C. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the 2002 Congress on Evolutionary Computation*.

Bongard, J. C., and Paul, C. (2000). Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In *Proceedings of the Sixth Internation Conference on Simulation of Adaptive Behavior*, 420–429. MIT Press.

Bongard, J. C., and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, 829–836. San Francisco, CA: Morgan Kaufmann.

Cangelosi, A., Parisi, D., and Nolfi, S. (1993). Cell division and migration in a genotype for neural networks. Technical Report PCIA-93, Institute of Psychology, C.N.R., Rome.

Carroll, S. B. (1995). Homeotic genes and the evolution of arthropods and chordates. *Nature*, 376:479–485.

Cohn, M., Patel, K., Krumlauf, R., Wilkinson, D., Clarke, J., and Tickle, C. (1997). HOX9 genes and vertebrate limb specification. *Nature*, 387:97–101.

Curtis, D., Apfeld, J., and Lehmann, R. (1995). Nanos is an evolutionarily conserved organizer of anterior-posterior polarity. *Development*, 121:1899–1910.

Dellaert, F. (1995). *Toward a Biologically Defensible Model of Development*. Master's thesis, Case Western Reserve University, Clevekand, OH.

Dellaert, F., and Beer, R. (1994a). Co-evolving body and brain in autonomous agents using a developmental model. Technical Report CES-94-16, Dept. of Computer Engineering and Science, Case Western Reserve University, Cleveland, OH.

Dellaert, F., and Beer, R. (1994b). Toward an evolvable model of development for autonomous agent synthesis. In Brooks, R. A., and Maes, P., editors, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*. Cambridge, MA: MIT Press.

Dellaert, F., and Beer, R. (1996). A developmental model for the evolution of complete autonomous agents. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.

Deloukas, P., Schuler, G., Gyapay, G., Beasley, E., Soderlund, C., Rodriguez-Tome, P., Hui, L., Matise, T., McKusick, K., Beckmann, J., Bentolila, S., Bihoreau, M., Birren, B., Browne, J., Butler, A., Castle, A., Chiannilkulchai, N., Clee, C., Day, P., Dehejia, A., Dibling, T., Drouot, N., Duprat, S., Fizames, C., and Bentley, D. (1998). A physical map of 30,000 human genes. *Science*, 282(5389):744–746.

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In Husbands, P., and Harvey, I., editors, *Proceedings of the Fourth European Conference on Artificial Life*, 205–213. Cambridge, MA: MIT Press.

Ellinson, R. P. (1987). Change in developmental patterns : Embryos of amphibians with large eggs. In Raff, R. A., and Raff, E. C., editors, *Development as an Evolutionary Process*, 1–21. New York: Alan R. Liss.

Fleischer, K., and Barr, A. H. (1993). A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In Langton, C. G., editor, *Artificial Life III*, 389–416. Addison-Wesley.

Force, A., Lynch, M., Pickett, F. B., Amores, A., lin Yan, Y., and Postlethwait, J. (1999). Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151:1531–1545.

Gans, C., and Northcutt, R. G. (1983). Neural crest and the origin of vertebrates: A new head. *Science*, 220(4594):268–274.

Gilbert, C. D., and Wiesel, T. N. (1992). Receptive field dynamics in adult primary visual cortex. *Nature*, 356:150–152.

Gilbert, S. F., editor (2000). *Developmental Biology*. Sunderland, MA: Sinauer Associates. Sixth edition.

Gomez, F., and Miikkulainen, R. (1999). Solving non-Markovian control tasks with neuroevolution. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*. Denver, CO: Morgan Kaufmann.

Gruau, F. (1993). Genetic synthesis of modular neural networks. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 318–325. San Francisco, CA: Morgan Kaufmann.

Gruau, F. (1994). *Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm*. PhD thesis, Ecole Normale Superieure de Lyon, France.

Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In Koza, J. R., Goldberg, D. E., Fogel, D. B., and Riolo, R. L., editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, 81–89. Cambridge, MA: MIT Press.

Hart, W. E., Kammeyer, T. E., and Belew, R. K. (1994). The role of development in genetic algorithms. Technical Report CS94-394, University of California San Diego, San Diego.

Hornby, G. S., and Pollack, J. B. (2001a). The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2002 Congress on Evolutionary Computation*.

Hornby, G. S., and Pollack, J. B. (2001b). Body-brain co-evolution using L-systems as a generative encoding. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA: Morgan Kaufmann.

Hubel, D. H., and Wiesel, T. N. (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28:229–289.

Jakobi, N. (1995). Harnessing morphogenesis. In *Proceedings of Information Processing in Cells and Tissues*, 29–41. University of Liverpool.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M., editors (1991). *Principles of Neural Science*. New York: Elsevier. Third edition.

Kauffman, S. A. (1993). *The Origins of Order*. New York: Oxford University Press.

Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476.

Komosinski, M., and Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated 3D agents. *Artificial Life*, 7(4):395–418.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Lall, S., and Patel, N. (2001). Conservation and divergence in molecular mechanisms of axis formation. *Annual Review of Genetics*, 35:407–447.

Lawrence, P. (1992). *The Making of a Fly*. Oxford: Blackwell Science Publishing.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development parts I and II. *Journal of Theoretical Biology*, 18:280–299 and 300–315.

Lindenmayer, A. (1974). Adding continuous components to L-systems. In Rozenberg, G., and Salomaa, A., editors, *L Systems, Lecture Notes in Computer Science 15*, 53–68. Heidelberg, Germany: Springer-Verlag.

Lipson, H., and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.

Luke, S., and Spector, L. (1996). Evolving graphs and networks with edge encoding: Preliminary report. In Koza, J. R., editor, *Late-breaking Papers of Genetic Programming 1996*. Stanford Bookstore.

Marin, E., Jeffries, G. S. X. E., Komiyama, T., Zhu, H., and Luo, L. (2002). Representation of the glomerular olfactory map in the Drosophila brain. *Cell*, 109(2):243–255.

Martin, A. P. (1999). Increasing genomic complexity by gene duplication and the origin of vertebrates. *The American Naturalist*, 154(2):111–128.

Michie, D., and Chambers, R. A. (1968). BOXES: An experiment in adaptive control. In Dale, E., and Michie, D., editors, *Machine Intelligence*. Edinburgh, UK: Oliver and Boyd.

Mjolsness, E., Sharp, D. H., and Reinitz, J. (1991). A connectionist model of development. *Journal of Theoretical Biology*, 152:429–453.

Moriarty, D. E., and Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32.

Nijhout, H. F., and Emlen, D. J. (1998). Competition among body parts in the development and evolution of insect morphology. *Proceedings of the National Academy of Sciences of the USA*, 95:3685–3689.

Nolfi, S., and Parisi, D. (1991). Growing neural networks. Technical Report PCIA-91- 15, Institute of Psychology, C.N.R., Rome.

O'Reilly, U.-M. (2000). Emergent design: Artificial life for architecture design. In *7th International Conference on Artificial Life (ALIFE-00)*. Cambridge, MA: MIT Press.

Prusinkiewicz, P., and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Heidelberg, Germany: Springer-Verlag.

Radding, C. M. (1982). Homologous pairing and strand exchange in genetic recombination. *Annual Review of Genetics*, 16:405–437.

Raff, E. C., Popodi, E. M., Sly, B. J., Turner, F. R., Villinski, J. T., and Raff, R. A. (1999). A novel ontogenetic pathway in hybrid embryos between species with different modes of development. *Development*, 126:1937–1945.

Raff, R. A. (1996). *The Shape of Life: Genes, Development, and the Evolution of Animal Form.* Chicago: The University of Chicago Press.

Reinhart, B. J., Slack, F. J., Basson, M., Pasquinelli, A. E., Bettinger, J. C., Rougvie, A. E., Horvitz, H. R., and Ruvkun, G. (2000). The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. *Nature*, 403:901–905.

Saravanan, N., and Fogel, D. B. (1995). Evolving neural control systems. *IEEE Expert*, 23–27.

Siddiqi, A., and Lucas, S. (1999). A comparison of matrix rewriting versus direct encoding for evolving neural networks. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC'98)*, 392–397. IEEE.

Sigal, N., and Alberts, B. (1972). Genetic recombination: The nature of a crossed strand-exchange between two homologous DNA molecules. *Journal of Molecular Biology*, 71(3):789–793.

Sigrist, C. B., and Sommer, R. J. (1999). Vulva formation in Pristionchus pacificusrelies relies on continuous gonadal induction. *Development Genes and Evolution*, 209:451–459.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. In Brooks, R. A., and Maes, P., editors, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, 28–39. Cambridge, MA: MIT Press.

Slijper, E. J. (1962). *Whales.* New York: Basic Books.

Stanley, K. O., and Miikkulainen, R. (2002a). Continual coevolution through complexification. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002).* San Francisco, CA: Morgan Kaufmann.

Stanley, K. O., and Miikkulainen, R. (2002b). Efficient reinforcement learning through evolving neural network topologies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002).* San Francisco, CA: Morgan Kaufmann.

Stanley, K. O., and Miikkulainen, R. (2002c). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B*, 237:37–72.

Vaario, J. (1994). From evolutionary computation to computational evolution. *Informatica*, 18(4):417–434.

Voss, S. R., and Shaffer, H. B. (1997). Adaptive evolution via a major gene effect : Paedomorphosis in the Mexican axolotl. *Proceedings of the National Academy of Sciences of the USA*, 94:14185–14189.

Waddington, C. H. (1942). Canalization of development and the inheritance of acquired characters. *Nature*, 150:563.

Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Whitley, D., Dominic, S., Das, R., and Anderson, C. W. (1993). Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284.

Wieland, A. (1991). Evolving neural network controllers for unstable systems. In *Proceedings of the International Joint Conference on Neural Networks* (Seattle, WA), 667–673. Piscataway, NJ: IEEE.

Wieland, A. P. (1990). Evolving controls for unstable systems. In Touretzky, D. S., Elman, J. L., Sejnowski, T. J., and Hinton, G. E., editors, *Connectionist Models: Proceedings of the 1990 Summer School*, 91–102. San Francisco, CA: Morgan Kaufmann.

Wilkins, A. (2002). *The Evolution of Developmental Pathways*. Sunderland, MA: Sinauer Associates.

Williams, R., Bastiani, M., Lia, B., and Chulupa, L. (1986). Growth cones, dying axons, developmental fluctuations in the fiber population of the cats optic nerve. *Journal of Computational Neurology*, 246:32–69.

Williams, R., Cavada, C., and Reinoso-Saurez, F. (1993). Rapid evolution of the visual system : A cellular assay of the retina and dorsal lateral geniculate nucleus of the Spanish wildcat and the domestic cat. *Journal of Neuroscience*, 13(1):208–228.

Wolpert, L. (1987). Constancy and change in the development and evolution of pattern. In Goodwin, B. C., Holder, N., and Wylie, C. C., editors, *Development and Evolution*, 47–57. Cambridge: Cambridge University Press.

Zigmond, M. J., Bloom, F. E., Landis, S. C., Roberts, J. L., and Squire, L. R., editors (1999). *Fundamental Neuroscience*. London: Academic Press.