Thomas S. Ray
ATR Human Information Processing Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02, Japan
ray@hip.atr.co.jp, ray@udel.edu

An Evolutionary Approach to Synthetic Biology,
Zen and the Art of Creating Life

October 21, 1993

Ray, T. S. In press. An evolutionary approach to synthetic biology, Zen and the art of creating life. Artificial Life 1(1): xx–xx. MIT Press.

Abstract

Our concepts of biology, evolution and complexity are constrained by having observed only a single instance of life, life on Earth. A truly comparative biology is needed to extend these concepts. Because we can not observe life on other planets, we are left with the alternative of creating artificial life forms on Earth. I will discuss the approach of inoculating evolution by natural selection into the medium of the digital computer. This is not a physical/chemical medium, it is a logical/informational medium. Thus these new instances of evolution are not subject to the same physical laws as organic evolution (e.g., the laws of thermodynamics), and therefore exist in what amounts to another universe, governed by the "physical laws" of the logic of the computer. This exercise gives us a broader perspective on what evolution is and what it does.

An evolutionary approach to synthetic biology consists of inoculating the process of evolution by natural selection into an artificial medium. Evolution is then allowed to find the natural forms of living organisms in the artificial medium. These are not models of life, but independent instances of life. This essay is intended to communicate a way of thinking about synthetic biology that leads to a particular approach: to understand and respect the natural form of the artificial medium, to facilitate the process of evolution in generating forms that are adapted to the medium, and to let evolution find forms and processes that naturally exploit the possibilities inherent in the medium. Examples are cited of synthetic biology embedded in the computational medium, where in addition to being an exercise in experimental comparative evolutionary biology, it is also a possible means of harnessing the evolutionary process for the production of complex computer software.

1 Synthetic Biology

Artificial Life (AL) is the enterprise of understanding biology by constructing biological phenomena out of artificial components, rather than breaking natural life forms down into their component parts. It is the synthetic rather than the reductionist approach. I will describe an approach to the synthesis of artificial living forms that exhibit natural evolution.

The umbrella of Artificial Life is broad, and covers three principal approaches to synthesis: in hardware (e.g., robotics, nanotechnology), in software (e.g., replicating and evolving computer programs), in wetware (e.g., replicating and evolving organic molecules, nucleic acids or others). This essay will focus on software synthesis, although it is hoped that the issues discussed will be generalizable to any synthesis involving the process of evolution.

I would like to suggest that software syntheses in AL could be divided into two kinds: simulations and instantiations of life processes. AL simulations represent an advance in biological modeling, based on a bottom-up approach, that has been made possible by the increase of available computational power. In the older approaches to modeling of ecological or evolutionary phenomena, systems of differential equations were set up that expressed relationships between covarying quantities of entities (i.e., genes, alleles, individuals, or species) in the populations or communities.

The new bottom up approach creates a population of data structures, with each instance of the data structure corresponding to a single entity. These structures contain variables defining the state of an individual. Rules are defined as to how the individuals interact with one another and with the environment. As the simulation runs, populations of these data structures interact according to local rules, and the global behavior of the system emerges from those interactions. Several very good examples of bottom up ecological models have appeared in the AL literature [34, 92]. However, ecologists have also developed this same approach independently of the AL movement, and have called the approach "individual based" models [19, 40].

The second approach to software synthesis is what I have called instantiation rather than simulation. In simulation, data structures are created which contain variables that represent the states of the entities being modeled. The important point is that in simulation, the data in the computer is treated as a representation of something else, such as a population of mosquitoes or trees. In instantiation, the data in the computer does not represent anything else. The data patterns in an instantiation are considered to be living forms in their own right, and are not models of any natural life form. These can from the basis of a comparative biology [58].

The object of an AL instantiation is to introduce the natural form and process of life into an artificial medium. This results in an artificial life form in some medium other than carbon chemistry, and is not a model of organic life forms. The approach discussed in this essay involves introducing the process of evolution by natural selection into the computational medium. I consider evolution to be the fundamental process of life, and the generator of living form.

2 Recognizing Life

Most approaches to defining life involve assembling a short list of properties of life, and then testing candidates on the basis of whether or not they exhibit the properties on the list. The main problem with this approach is that there is disagreement as to what should be on the list. My private list contains only two items: self-replication and open-ended evolution. However, this reflects my biases as an evolutionary biologist.

I prefer to avoid the semantic argument and take a different approach to the problem of recognizing life. I was led to this view by contemplating how I would regard a machine that exhibited conscious intelligence at such a level that it could participate as an equal in a debate such as this. The machine would meet neither of my two criteria as to what life is, yet I don't feel that I could deny that the process it contained was alive.

This means that there are certain properties that I consider to be unique to life, and whose presence in a system signify the existance of life in that system. This suggests an alternative approach to the problem. Rather than creating a short list of minimal requirements and testing whether a system exhibits all items on the list, create a long list of properties unique to life and test whether a system exhibits *any* item on the list.

In this softer, more pluralistic approach to recognizing life, the objective is not to determine if the system is alive or not, but to determine if the system exhibits a "genuine" instance of some property that is a signature of living systems (e.g., self-replication, evolution, flocking, consciousness).

Whether we consider a system living because it exhibits some property that is unique to life amounts to a semantic issue. What is more important is that we recognize that it is possible to create disembodied but genuine instances of specific properties of life in artificial systems. This capability is a powerful research tool. By separating the property of life that we choose to study, from the many other complexities of natural living systems, we make it easier to manipulate and observe the property of interest. The objective of the approach advocated in this paper is to capture genuine evolution in an artificial system.

3 What Natural Evolution Does

Evolution by natural selection is a process that enters into a physical medium. Through iterated replication-with-selection of large populations through many generations, it searches out the possibilities inherent in the "physics and chemistry" of the medium in which it is embedded. It exploits any inherent self-organizing properties of the medium, and flows into natural attractors realizing and fleshing out their structure.

Evolution never escapes from its ultimate imperative: self-replication. However, the mechanisms that evolution discovers for achieving this ultimate goal gradually become so convoluted and complex that the underlying drive can seem to become superfluous. Some philosophers have argued that the evolutionary theory as expressed by the phrase "survival of the fittest" is tautological, in that the fittest are defined as those that survive to reproduce. In fact, fitness is achieved through innovation in engineering of the organism [82]. However there remains something peculiarly self-referential about the whole enterprise. There is some sense in which life may be a natural tautology.

Evolution is both a defining characteristic and the creative process of life itself. The living condition is a state that complex physical systems naturally flow into under certain conditions. It is a self-organizing, self-perpetuating state of auto-catalytically increasing complexity. The living component of the physical system quickly becomes the most complex part of the system, such that it re-shapes the medium, in its own image as it were. Life then evolves adaptations predominantly in relation to the living components of the system, rather than the non-living components. Life evolves adaptations to itself.

3.1 Evolution in Sequence Space

Think of organisms as occupying a "genotype space" consisting of all possible sequences of all possible lengths of the elements of the genetic system (i.e., nucleotides or machine instructions). When the first organism begins replicating, a single self-replicating creature, with a single sequence of a certain length occupies a single point in the genotype space. However, as the creature replicates in the environment, a population of creatures forms, and errors cause genetic variation, such that the population will form a cloud of points in the genotype space, centered around the original point.

Because the new genotypes that form the cloud are formed by random processes, most of them are completely inviable, and die without reproducing. However, some of them are capable of reproduction. These new genotypes persist, and as some of them are affected by mutation, the cloud of points spreads further. However, not all of the viable genomes are equally viable. Some of them discover tricks to replicate more efficiently. These genotypes increase in frequency, causing the population of creatures at the corresponding points in the genotype space to increase.

Points in the genotype space occupied by greater populations of individuals will spawn larger numbers of mutant offspring, thus the density of the cloud of points in the genotype space will shift gradually in the direction of the more fit genotypes. Over time, the cloud of points will percolate through the genotype space, either expanding outward as a result of random drift, or by flowing along fitness gradients.

Most of the volume of this space represents completely inviable sequences. These regions of the space may be momentarily and sparsely occupied by inviable mutants, but the cloud will never flow into the inviable regions. The cloud of genotypes may bifurcate as it flows into habitable regions in different directions, and it may split as large genetic changes spawn genotypes in distant but viable regions of the space. We may imagine that the evolving population of creatures will take the form of wispy clouds flowing through this space.

Now imagine for a moment the situation that there were no selection. This implies that every sequence is replicated at an equal rate. Mutation will cause the cloud of points to expand outward, eventually filling the space uniformly. In this situation, the complexity of the structure of the cloud of points does not increase through time, only the volume that it occupies. Under selection by contrast, through time the cloud will take on an intricate structure as it flows along fitness gradients and percolates by drift through narrow regions of viability in a largely uninhabitable space.

Consider that the viable region of the genotype space is a very small subset of the total volume of the space, but that it probably exhibits a very complex shape, forming tendrils and sheets sparsely permeating the otherwise empty space. The complex structure of this cloud can be considered to be a product of evolution by natural selection. This thought experiment appears to imply that the intricate structure that the cloud of genotypes may assume through evolution is fully deterministic. Its shape is pre-defined by the physics and chemistry and the structure of the environment, in much the same way that the form of the Mandlebrot set is pre-determined by its defining equation. The complex structure of this viable space is inherent in the medium, and is an example of "order for free" [45].

No living world will ever fill the entire viable subspace, either at a single moment of time, or even cumulatively over its entire history. The region actually filled will be strongly influenced by the original self-replicating sequence, and by stochastic forces which will by chance push the cloud down a subset of possible habitable pathways. Furthermore, co-evolution and ecological interactions imply that certain regions can only be occupied when certain other regions are also occupied. This concept of the flow of genotypes through the genotype space is essentially the same as that discussed by Eigen [22] in the context of "quasispecies". Eigen limited his discussion to species of viruses, where it is also easy to think of sequence spaces. Here, I am extending the concept beyond the bounds of the species, to include entire phylogenies of species.

3.2 Natural Evolution in an Artificial Medium

Until recently, life has been known as a state of matter, particularly combinations of the elements carbon, hydrogen, oxygen, nitrogen and smaller quantities of many others. However, recent work in the field of Artificial Life has shown that the natural evolutionary process can proceed with great efficacy in other media, such as the informational medium of the digital computer [1, 3, 7, 15, 16, 20, 24, 29, 43, 44, 51, 53, 54, 68, 69, 71, 72, 73, 74, 77, 78, 81, 89, 91]

These new natural evolutions, in artificial media, are beginning to explore the possibilities inherent in the "physics and chemistry" of those media. They are organizing themselves and constructing self-generating complex systems. While these new living systems are still so young that they remain in their primordial state, it appears that they have embarked on the same kind of journey taken by life on earth, and presumably have the potential to evolve levels of complexity that could lead to sentient and eventually intelligent beings.

If natural evolution in artificial media leads to sentient or intelligent beings, they will likely be so alien that they will be difficult to recognize. The sentient properties of plants are so radically different from those of animals, that they are generally unrecognized or denied by humans, and plants are merely in another kingdom of the one great tree of organic life on earth [70, 75, 88]. Synthetic organisms evolving in other media such as the digital computer, are not only not a part of the same phylogeny, but they are not even of the same physics. Organic life is based on conventional material physics, whereas digital life exists in a logical, not material, informational universe. Digital intelligence will likely be vastly different from human intelligence; forget the Turing test.

4 The Approach

> Marcel, a mechanical chessplayer... his exquisite 19th-century brainwork — the human art it took to build which has been flat lost, lost as the dodo bird ... But where inside Marcel is the midget Grandmaster, the little Johann Allgeier? where's the pantograph, and the magnets? Nowhere. Marcel really is a mechanical chessplayer. No fakery inside to give him any touch of humanity at all.
>
> — Thomas Pynchon, *Gravity's Rainbow*.

The objective of the approach discussed here, is to create an instantiation of evolution by natural selection in the computational medium. This creates a conceptual problem that requires considerable art to solve: ideas and techniques must be learned by studying organic evolution, and then applied to the generation of evolution in a digital medium, without forcing the digital medium into an "un-natural" simulation of the organic world.

We must derive inspiration from observations of organic life, but we must never lose sight of the fact that the new instantiation is not organic, and may differ in many fundamental ways. For example, organic life inhabits a Euclidean space, however computer memory is not a Euclidean space. Inter-cellular communication in the organic world is chemical in nature, and therefore a single message generally can pass no more information than on or off. By contrast, communication in digital computers generally involves the passing of bit patterns, which can carry much more information.

The fundamental principal of the approach being advocated here is *to understand and respect the natural form of the digital computer, to facilitate the process of evolution in generating forms that are adapted to the computational medium, and to let evolution find forms and processes that naturally exploit the possibilities inherent in the medium.*

Situations arise where it is necessary to make significant changes from the standard computer architecture. But such changes should be made with caution, and only when there is some feature of standard computer architectures which clearly inhibits the desired processes. Examples of such changes are discussed in the section "The Genetic Language" below. Less substantial changes are also discussed in the sections on the "Flaw" genetic operator, "Mutations", and "Artificial Death". The sections on "Spatial Topology" and "Digital 'Neural Networks' — Natural AI" are little tirades against examples of what I consider to be un-natural transfers of forms from the natural world to the digital medium.

5 The Computational Medium

The computational medium of the digital computer is an informational universe of boolean logic, not a material one. Digital organisms live in the memory of the computer, and are powered by the activity of the central processing unit (CPU). Whether the hardware of the CPU and memory is built of silicon chips, vacuum tubes, magnetic cores, or mechanical switches is irrelevant to the digital organism. Digital organisms should be able to take on the same form in any computational hardware, and in this sense are "portable" across hardware.

Digital organisms might as well live in a different universe from us, as they are not subject to the same laws of physics and chemistry. They are subject to the "physics and chemistry" of the rules governing the manipulation of bits and bytes within the computer's memory and CPU. They never "see" the actual material from which the computer is constructed, they see only the logic and rules of the CPU and the operating system. These rules are the only "natural laws" that govern their behavior. They are not influenced by the natural laws that govern the material universe (e.g., the laws of thermodynamics).

A typical instantiation of this type involves the introduction of a self-replicating machine language program into the RAM memory of a computer subject to random errors such as bit flips in the memory or occasionally inaccurate calculations [3, 7, 20, 53, 71]. This generates the basic conditions for evolution by natural selection as outlined by Darwin [14]: self-replication in a finite environment with heritable genetic variation.

In this instantiation, the self-replicating machine language program is thought of as the individual "digital organism" or "creature". The RAM memory provides the physical space that the creatures occupy. The CPU provides the source of energy. The memory consists of a large array of bits, generally grouped into eight bit bytes and sixteen or thirty-two bit words. Information is stored in these arrays as voltage patterns which we usually symbolize as patterns of ones and zeros.

The "body" of a digital organism is the information pattern in memory that constitutes its machine language program. This information pattern is data, but when it is passed to the CPU, it is interpreted as a series of executable instructions. These instructions are arranged in such a way that the data of the body will be copied to another location of memory. The informational patterns stored in the memory are altered only through the activity of the CPU. It is for this reason that the CPU is thought of as the analog of the energy source. Without the activity of the CPU, the memory would be static, with no changes in the informational patterns stored there.

The logical operations embodied in the instruction set of the CPU constitute a large part of the definition of the "physics and chemistry" of the digital universe. The topology of the computer's memory (discussed below) is also a significant component of the digital physics. The final component of the digital physics is the operating system, a software program running on the computer, which embodies rules for the allocation of resources such as memory space and CPU time to the various processes running on the computer.

The instruction set of the CPU, the memory, and the operating system together define the complete "physics and chemistry" of the universe inhabited by the digital organism. They constitute the physical environment within which digital organisms will evolve. Evolving digital organisms will compete for access to the limited resources of memory space and CPU time, and evolution will generate adaptations for the more agile access to and the more efficient use of these resources.

6 The Genetic Language

The simplest possible instantiation of a digital organism is a machine language program that codes for self-replication. In this case, the bit pattern that makes up the program is the body of the organism, and at the same time its complete genetic material. Therefore, the machine language defined by the CPU constitutes the genetic language of the digital organism.

It is worth noting at this point that the organic organism most comparable to this kind of digital organism is the hypothetical, and now extinct, RNA organism [6]. These were presumably nothing more than RNA molecules capable of catalyzing their own replication. What the supposed RNA organisms have in common with the simple digital organism is that a single molecule constitutes the body and the genetic information, and effects the replication. In the digital organism a single bit pattern performs all the same functions.

The use of machine code as a genetic system raises the problem of brittleness. It has generally been assumed by computer scientists that machine language programs can not be evolved because random alterations such as bit flips and recombinations will always produce inviable programs. It has been suggested [23] that overcoming this brittleness and "Discovering how to make such self-replicating patterns more robust so that they evolve to increasingly more complex states is probably the central problem in the study of artificial life."

The assumption that machine languages are too brittle to evolve is probably true, as a consequence of the fact that machine languages have not previously been designed to survive random alterations. However, recent experiments have shown that brittleness can be overcome by addressing the principal causes, and without fundamentally changing the structure of machine languages [71, 78].

The first requirement for evolvability is graceful error handling. When code is being randomly altered, every possible meaningless or erroneous condition is likely to occur. The CPU should be designed to handle these conditions without crashing the system. The simplest solution is for the CPU to perform no operation when it meets these conditions, perhaps setting an error flag, and to proceed to the next instruction.

Due to random alterations of the bit patterns, all possible bit patterns are likely to occur. Therefore a good design is for all possible bit patterns to be interpretable as meaningful instructions by the CPU. For example in the Tierra system [71, 72, 73, 74, 77, 78], a five bit instruction set was chosen, in which all thirty-two five bit patterns represent good machine instructions.

This approach (all bit patterns meaningful) also could imply a lack of syntax, in which each instruction stands alone, and need not occur in the company of other instructions. To the extent that the language includes syntax, where instructions must precede or follow one another in certain orders, random alterations are likely to destroy meaningful syntax thereby making the language more brittle. A certain amount of this kind of brittleness can be tolerated as long as syntax errors are also handled gracefully.

During the design of the first evolvable machine language [71], a standard machine language (Intel 80X86) was compared to the genetic language of organic life, to attempt to understand the difference between the two languages that might contribute to the brittleness of the former and the robustness of the latter. One of the outstanding differences noted was in the number of basic informational objects contained in the two.

The organic genetic language is written with an alphabet consisting of four different nucleotides. Groups of three nucleotides form sixty-four "words" (codons), which are translated into twenty amino-acids by the molecular machinery of the cell. The machine language is written with sequences of two voltages (bits) which we conceptually represent as ones and zeros. The number of bits that form a "word" (machine instruction) varies between machine architectures, and in some architectures is not constant. However, the number required generally ranges from sixteen to thirty-two. This means that there are from tens of thousands to billions of machine instruction bit patterns, which are translated into operations performed by the CPU.

The thousands or billions of bit patterns that code for machine instructions contrasts with the sixty four nucleotide patterns that code for amino acids. The sixty-four nucleotide patterns are degenerate, in that they code for only twenty amino-acids. Similarly, the machine codes are degenerate, in that there are at most hundreds rather than thousands or billions of machine operations.

The machine codes exhibit a massive degeneracy (with respect to actual operations) as a result of the inclusion of data into the bit patterns coding for the operations. For example, the add operation will take two operands, and produce as a result the sum of the two operands. While there may be only a single add operation, the instruction may come in several forms depending on where the values of the two operands come from, and where the resultant sum will be placed. Some forms of the add instruction allow the value(s) of the operand(s) to be specified in the bit pattern of the machine code.

The inclusion of numeric operands in the machine code is the primary cause of the huge degeneracy. If numeric operands are not allowed, the number of bit patterns required to specify the complete set of operations collapses to at most a few hundred.

While there is no empirical data to support it, it is suspected that the huge degeneracy of most machine languages may be a source of brittleness. The logic of this argument is that mutation causes random swapping among the fundamental informational objects, codons in the organic language, and machine instructions in the digital language. It seems more likely that meaningful results will be produced when swapping among sixty-four objects than when swapping among billions of objects.

The size of the machine instruction set can be made comparable to the number of codons simply by eliminating numeric operands embedded in the machine code. However, this change creates some new problems. Computer programs generally function by executing instructions located sequentially in memory. However, in order to loop or branch, they use instructions such as "jump" to cause execution to jump to some other part of the program. Since the locations of these jumps are usually fixed, the jump instruction will generally have the target address included as an operand embedded in the machine code.

By eliminating operands from the machine code, we generate the need for a new mechanism of addressing for jumps. To resolve this problem, an idea can be borrowed from molecular biology. We can ask the question: how do biological molecules address one another? Molecules do not specify the coordinates of the other molecules they interact with. Rather, they present shapes on their surfaces that are complementary to the shapes on the surfaces of the target molecules. The concept of complementarity in addressing can be introduced to machine languages by allowing the jump instruction to be followed by some bit pattern, and having execution jump to the nearest occurrence of the complementary bit pattern.

In the development of the Tierran language, two changes were introduced to the machine language to reduce brittleness: elimination of numeric operands from the code, and the use of complementary patterns to control addressing. The resulting language proved to be evolvable [71]. As a result, nothing was learned about evolvability, because only one language was tested, and it evolved. It is not known what features of the language enhance its evolvability, which detract, and which do not affect evolvability. Subsequently, three additional languages were tested and the four languages were found to vary in their patterns and degree of evolvability [78]. However, it is still not known how the features of the language affect its evolvability.

7 Genetic Operators

In order for evolution to occur, there must be some genetic variation among the offspring. In organic life, this is insured by natural imperfections in the replication of the informational molecules. However, one way in which digital "chemistry" differs from organic chemistry is in the degree of perfection of its operations. In the computer, the genetic code can be reliably replicated without errors to such a degree that we must artificially introduce errors or other sources of genetic variation in order to induce evolution.

7.1 Mutations

In organic life, the simplest genetic change is a "point mutation", in which a single nucleic acid in the genetic code is replaced by one of the three other nucleic acids. This can cause an amino acid substitution in the protein coded by the gene. The nucleic acid replacement can be caused by an error in the replication of the DNA molecule, or it can be caused by the effects of radiation or mutagenic chemicals.

In the digital medium, a comparably simple genetic change can result from a bit flip in the memory, where a one is replaced by a zero, or a zero is replaced by a one. These bit flips can be introduced in a variety of ways that are analogous to the various natural causes of mutation. In any case, the bit flips must be introduced at a low to moderate frequency, as high frequencies of mutation prevent the replication of genetic information, and lead to the death of the system [74].

Bit flips may be introduced at random anywhere in memory, where they may or may not hit memory actually occupied by digital organisms. This could be thought of as analogous to cosmic rays falling at random and disturbing molecules which may or may not be biological in nature. Bit flips may also be introduced when information is copied in the memory, which could be analogous to the replication errors of DNA. Alternatively, bit flips could be introduced in memory as it is accessed, either as data or executable code. This could be thought of as damage due to "wear and tear".

7.2 Flaws

Alterations of genetic information are not the only source of noise in the system. In organic life, enzymes have evolved to increase the probability of chemical reactions that increase the fitness of the organism. However, the metabolic system is not perfect. Undesired chemical reactions do occur, and desired reactions sometimes produce undesired by-products. The result is the generation of molecular species that can "gum up the works", having unexpected consequences, generally lowering the fitness of the organism, but possibly raising it.

In the digital system, an analogue of metabolic (non-genetic) errors can be introduced by causing the computations carried out by the CPU to be probabilistic, producing erroneous results at some low frequency. For example, any time a sum or difference is calculated, the result could be off by some small value (e.g. plus or minus one). Or, if all bits are shifted one position to the left or right, an appropriate error would be to shift by two positions or not at all. When information is transferred from one location to another, either in the RAM memory or the CPU registers, it could occasionally be transferred from the wrong location, or to the wrong location. While flaws do not directly cause genetic changes, they can cause a cascade of events that result in the production of an offspring that is genetically different from the parent.

7.3 Recombination — Sex

7.3.1 The Nature of Sex

In organic life, there are a wide variety of mechanisms by which offspring are produced which contain genetic material from more that one parent. This is the sexual process. Recombination mechanisms range from very primitive and haphazard to elaborately orchestrated.

At the primitive extreme we find certain species of bacteria, in which upon death, the cell membrane breaks open, releasing the DNA into the surrounding medium. Fragments of this dead DNA are absorbed across the membranes of other bacteria of the same species, and incorporated into their genome [59]. This is a one way transferral of genetic material, rather than a reciprocal exchange.

At the complex extreme we find the conventional sexual system of most of the higher animals, in which each individual contains two copies of the entire genome. At reproduction, each of two parents contributes one complete copy of the genome (half of their genetic material) to the offspring. This means that each offspring receives one half of its genetic material from each of two parents, and each parent contributes one half of its genetic material to each offspring. Very elaborate behavioral and molecular mechanisms are required to orchestrate this joint contribution of genetic material to the offspring.

The preponderance of sex remains an enigma to evolutionary theory [5, 26, 31, 32, 55, 61, 86, 96]. Careful analysis has failed to show any benefits from sex, at the level of the individual organism, that outweigh the high costs (e.g., passing on only half of the genome). The only obvious benefit of sex is that it provides diversity among the offspring, allowing the species to adapt more readily to a changing environment. However, quantitative analysis has shown that in order for sex to be favored by selection at the individual level, it is not enough for the environment to change unpredictably, the environment must actually change capriciously [13, 57]. That is, whatever genotype has the highest fitness this generation, must have the lowest fitness the next generation, or at least a trend in this direction, a negative heritability of fitness.

One theory to explain the perpetuation of sex (based on the Red Queen hypothesis, see below) states that the environment is in fact capricious, due to the importance of biotic factors in determining selective forces. That is, sex is favored because it is necessary to maintain adaptation in the face of evolving species in the environment (e.g., predators/parasites, prey/hosts, competitors) who themselves are sexual, and can undergo rapid evolutionary change. Predators and parasites will tend to evolve so as to favor attacking whatever genotype of their prey/host is the most common. The genotype that is most successful at present is targeted for future attack. This dynamic makes the environment capricious in the sense discussed above.

There are fundamental differences in the nature of the evolutionary process between asexual and sexual organisms. The evolving entity in an asexual species is a branching lineage of genetic individuals which retain their genetic identity through the generations. In a sexual species, the evolving entity is a collective "gene pool", and genetic individuals are absolutely ephemeral, lasting only one generation.

Recalling the discussion of "genotype space" above in the section "Evolution in Sequence Space", imagine that we could represent genotype space in two dimensions, and that we allow a third dimension to represent time. Visualize now, an evolving asexual organism. Starting with a single individual, it would occupy a single point in the genotype space at time zero. When it reproduces, if there is no mutation, its offspring would occupy the same point in genotype space, at a later time. Thus the lineage of the asexual organism would appear as a line moving forward in time. If mutations occur, they cause the offspring to occupy new locations in genotype space, forming branches in the lineage.

Through time, the evolving asexual lineage would form a tree like structure in the genotype space–time coordinates. However, every individual branch of the tree will evolve independently of all the others. While there may be ecological interactions between genetically different individuals, there is no exchange of genetic material between them. From a genetic point of view, each branch of the tree is on its own; it must adapt, or fail to adapt based on its own genetic resources.

In order to visualize an evolving sexual population we must start with a population of individuals, each of which will be genetically unique. Thus they will appear as a scatter of points in the genotype space plane at time zero. In the next generation, all of the original genotypes will be dead, however, a completely new set of genotypes will have been formed from new combinations of pieces of the genomes from the previous generation. No individual genotypes will survive from one generation to the next, thus over time, the evolving sexual population appears as a diffuse cloud of disconnected points, with no lines formed from persistent genotypes.

The most important distinction between the evolving asexual and sexual populations is that the asexual individuals are genetically isolated and must adapt or not based on the limited genetic resources of the individual, while sexual organisms by comparison draw on the genetic resources of the entire population, due to the flow of genes resulting from sexual matings. The entity that evolves in an asexual population is an isolated but branching lineage of genetic individuals. In a sexual population, the individual is ephemeral, and the entity that evolves is a "gene pool".

Due to the genetic cohesion of a sexual population and the ephemeral nature of its individuals, the evolving sexual entity exists at a higher level of organization than the individual organism. The evolving entity, a gene pool, is supra-organismal. It samples the environment through many individuals simultaneously, and pools their genetic resources in finding adaptive genetic combinations.

The definition of the biological species is based on a concept of sexual reproduction: a group of individuals capable of interbreeding freely under natural conditions. Species concepts simply do not apply well to asexual species. In order for synthetic life to be useful for the study of the properties of species and the speciation process, it must include an organized sexual process, such that the evolving entity is a gene pool.

7.3.2 Implementation of Digital Sex

The above discussions of the nature of sexuality are intended to make the point that it is an important process in evolutionary biology, and should be included in synthetic implementations of life. The sexual process is implemented with the "cross-over" genetic operator in the field of genetic algorithms, where it has been considered to be the most important genetic operator [35].

The cross-over operator has also been implemented in synthetic life systems [76, 91]. However, it has been implemented in the spirit of a genetic algorithm, rather than in the spirit of synthetic life. This is because in these implementations the cross-over process is not under the control of the organism, but rather is forced on the individual. In addition, these implementations are based on haploid sex not diploid sex (see below). In order to address many of the interesting evolutionary questions surrounding sexuality, the sexual process must be optional, at least through evolution, and should be diploid.

Primitive sexual processes have appeared spontaneously in the Tierra synthetic life system [71]. However, there apparently has still not been an implementation of natural organized sexuality in a synthetic system. I would like to discuss my conception of how this could be implemented, with particular reference to the Tierra system.

It would seem that the simplest way of implementing an organized sexuality that would give rise to an evolving gene pool would involve the use of "ploidy". Ploidy refers to a system in which each individual contains multiple copies of the complete genome. In the most familiar sexual system (that used by humans), the gametes (egg and sperm) contain one copy of the genome (they are haploid), and all other stages of the life cycle contain two copies (they are diploid), which derive from the union of a sperm and egg.

In a digital organism whose body consists of a sequence of machine code, it would be easy to duplicate the sequence and include two copies within the cell. However, some problems can arise with this configuration, if the two copies of the genome occupy adjacent blocks of memory. Which copy of the genome will be executed? When the organism contributes one of its two copies of the genome to an offspring, which of the two copies will be contributed, and how can the mother cell recognize where one complete genome begins and ends?

A solution to these problems that has been partially implemented in the Tierra system is to have the two copies of the genome intertwined, rather than in adjacent blocks of memory. This can be done by letting alternate bytes represent one genome, and the skipped bytes the other genome. Tierran instructions utilize only five bits, and so are mapped to successive bytes in memory. If we instead place successive instructions in successive sixteen bit words, one copy of the genome can occupy the high order bytes, and the other genome can occupy the low order bytes of the words.

This arrangement facilitates relatively simple solutions to the problems mentioned above. Execution of the genome takes place by having the instruction pointer execute alternate bytes. In a diploid organism there are two tracks. The track to initially be executed can be chosen at random. At a certain frequency, or under certain circumstances, the executing track can be switched so that both copies of the genome will be expressed.

Having two parallel tracks helps to resolve the problem of recognizing where one copy of the genome ends and the other begins, since both genomes usually begin and end together. Copying of the genome, like execution, can occur along one track. Optionally, tracks could be switched during the copy process, to introduce an effect similar to crossing over in meiosis. In addition, the use of both tracks can be optional, so that haploid and diploid organisms can coexist in the same soup, and evolution can favor either form, according to selective pressures.

7.4 Transposons

The explosion of diversity in the Cambrian occurred in the lineage of the eukaryotes; the prokaryotes did not participate. One of the most striking genetic differences between eukaryotes and prokaryotes is that most of the genome of prokaryotes is translated into proteins, while most of the genome of eukaryotes is not. It has been estimated that typically 98% of the DNA in eukaryotes is neither translated into proteins nor involved in gene regulation, that it is simply "junk" DNA [93]. It has been suggested that much of this junk code is the result of the self-replication of pieces of DNA within rather than between cells [21, 67].

Mobile genetic elements, transposons, have this intra-genome self-replicating property. It has been estimated that 80% of spontaneous mutations are caused by transposons [12, 30]. Repeated sequences, resulting from the activity of mobile elements, range from dozens to millions in numbers of copies, and from hundreds to tens of thousands of base pairs in length. They vary widely in dispersion patterns from clumped to sparse [41].

Larger transposons carry one or more genes in addition to those necessary for transposition. Transposons may grow to include more genes; one mechanism involves the placement of two transposons into close proximity so that they act as a single large transposon incorporating the intervening code. In many cases transposons carry a sequence that acts as a promoter, altering the regulation of genes at the site of insertion [90].

Transposons may produce gene products and often are involved in gene regulation [17]. However, they may have no effect on the external phenotype of the individual [21]. Therefore they evolve through another paradigm of selection, one that does not involve an external phenotype. They are seen as a mechanism for the selfish spread of DNA which may become inactive junk after mutation [67].

DNA of transposon origin can be recognized by their palindrome endings flanked by short non-reversed repeated sequences resulting from insertion after staggered cuts. In *Drosophila melanogaster* approximately 5 to 10 percent of its total DNA is composed of sequences bearing these signs. There are many families of such repeated elements, each family possessing a distinctive nucleotide sequence, and distributed in many sites throughout the genome. One well known repeated sequence occurring in humans is found to have as many as a half million copies in each haploid genome [87].

Elaborate mechanisms have evolved to edit out junk sequences inserted into critical regions. An indication of the magnitude of the task comes from the recent cloning of the gene for cystic fibrosis, where it was discovered that the gene consists of 250,000 base pairs, only 4,440 of which code for protein, the remainder are edited out of the messenger RNA before translation [46, 56, 80, 79].

It appears that many repeated sequences in genomes may have originated as transposons favored by selection at the level of the gene, favoring genes which selfishly replicated themselves within the genome. However, some transposons may have coevolved with their host genome as a result of selection at the organismal or populational level, favoring transposons which introduce useful variation through gene rearrangement. It has been stated that: "transposable elements can induce mutations that result in complex and intricately regulated changes in a single step", and they are "A highly evolved macromutational mechanism" [90].

In this manner, "smart" genetic operators may have evolved, through the interaction of selection acting at two or more hierarchical levels (it appears that some transposons have followed another evolutionary route, developing inter-cellular mobility and becoming viruses [41]). It is likely that transposons today represent the full continuum from purely parasitic "selfish DNA" and viruses to highly coevolved genetic operators and gene regulators. The possession of smart genetic operators may have contributed to the explosive diversification of eukaryotes by providing them with the capacity for natural genetic engineering.

In designing self replicating digital organisms, it would be worthwhile to introduce such genetic parasites, in order to facilitate the shuffling of the code that they bring about. Also, the excess code generated by this mechanism provides a large store of relatively neutral code that can randomly explore new configurations through the genetic operations of mutation and recombination. When these new configurations confer functionality, they may become selected for.

8 Artificial Death

Death must play a role in any system that exhibits the process of evolution. Evolution involves a continuing iteration of selection, which implies differential *death*. In natural life, death occurs as a result of accident, predation, starvation, disease, or if these fail to kill the organism, it will eventually die from senescence resulting from an accumulation of wear and tear at every level of the organism including the molecular.

In normal computers, processes are "born" when they are initiated by the user, and "die" when they complete their task and halt. A process whose goal is to repeatedly replicate itself is essentially an endless loop, and would not spontaneously terminate. Due to the perfection of normal computer systems, we can not count on "wear and tear" to eventually cause a process to terminate.

In synthetic life systems implemented in computers, death is not likely to be a process that would occur spontaneously, and it must generally be introduced artificially by the designer. Everyone who has set up such a system has found their own unique solutions. Todd [94] recently discussed this problem in general terms.

In the Tierra system [71] death is handled by a "reaper" function of the operating system. The reaper uses a linear queue. When creatures are born, they enter the bottom of the queue. When memory is full, the reaper frees memory to make space for new creatures by killing off the top of the queue. However, each time an individual generates an error condition, it moves up the reaper queue one position.

An interesting variation on this was introduced by Barton-Davis [3] who eliminated the reaper queue. In its place, he caused the "flaw rate" (see section on Flaws above) to increase with the age of the individual, in mimicry of wear and tear. When the flaw rate reached 100%, the individual was killed. Skipper [81] provided a "suicide" instruction, which if executed, would cause a process to terminate (die). The evolutionary objective then became to have a suicide instruction in your genome which you do not execute yourself, but which you try to get other individuals to execute. Litherland [51] introduced death by local crowding. Davidge caused processes to die when they contained certain values in their registers [16]. Gray [29] allowed each process six attempts at reproduction, after which they would die.

9 Operating System

Much of the "physics and chemistry" of the digital universe is determined by the specifications of the operations performed by the instruction set of the CPU. However, the operating system also determines a significant part of the physical context. The operating system manages the allocation of critical resources such as memory space and CPU cycles.

Digital organisms are processes that spawn processes. As processes are born, the operating system will allocate memory and CPU cycles to them, and when they die, the operating system will return the resources they had utilized to the pool of free resources. In synthetic life systems, the operating system may also play a role in managing death, mutations and flaws.

The management of resources by the operating system is controlled by algorithms. From the point of view of the digital organisms these take the form of a set of logical rules like those embodied in the logic of the instruction set. In this way, the operating system is a defining part of the physics and chemistry of the digital universe. Evolution will explore the possibilities inherent in these rules, finding ways to more efficiently gain access to and exploit the resources managed by the operating system.

10 Spatial Topology

Digital organisms live in the memory space of computers, predominantly in the RAM memory, although they could also live on disks or any other storage device, or even within networks to the extent that the networks themselves can store information. In essence, digital organisms live in the space that has been referred to as "cyber-space". It is worthwhile reflecting on the topology of this space, as it is a radically different space from the one we live in.

A typical UNIX workstation, or MacIntosh computer includes a RAM memory that can contain some megabytes of data. This is "flat" memory, meaning that it is essentially unstructured. Any location in memory can be accessed through its numeric address. Thus adjacent locations in memory are accessed through successive integer values. This addressing convention causes us to think of the memory as a linear space, or a one-dimensional space.

However, this apparent one-dimensionality of the RAM memory is something of an illusion generated by the addressing scheme. A better way of understanding the topology of the memory comes from asking "what is the distance between two locations in memory". In fact the distance can not be measured in linear units. The most appropriate unit is the time that it takes to move information between the two points.

Information contained in the RAM memory can not move directly from point to point. Instead the information is transferred from the RAM to a register in the CPU, and then from the CPU back to the new location in RAM. Thus the distance between two locations in RAM is just the time that it takes to move from the RAM to the CPU plus the time that it takes to move from the CPU to the RAM. Because all points in the RAM are equidistant from the CPU, the distance between any pair of locations in the RAM is the same, regardless of how far apart they may appear based on their numeric addresses.

A space in which all pairs of points are equidistant is clearly not a Euclidean space. That said, we must recognize however, that there are a variety of ways in which memory is normally addressed, that gives it the appearance, at least locally, of being one dimensional. When code is executed by the CPU, the instruction pointer generally increments sequentially through memory, for short distances, before jumping to some other piece of code. For those sections of code where instructions are sequential, the memory is effectively one-dimensional. In addition, searches of memory are often sequentially organized (e.g., the search for complementary templates in Tierra). This again makes the memory effectively one-dimensional within the search radius. Yet even under these circumstances, the memory is not globally one-dimensional. Rather it consists of many small one dimensional pieces, each of which has no meaningful spatial relationship to the others.

Because we live in a three-dimensional Euclidean space, we tend to impose our familiar concepts of spatial topology onto the computer memory. This leads first to the erroneous perception that memory is a one-dimensional Euclidean space, and second, it often leads to the conclusion that the digital world could be enriched by increasing the dimensionality of the Euclidean memory space.

Many of the serious efforts to extend the Tierra model have included as a central feature, the creation of a two-dimensional space for the creatures to inhabit [3, 15, 16, 53, 81]. The logic behind the motivation derives from contemplation of the extent to which the dimensionality of the space we live in permits the richness of pattern and process that we observe in nature. Certainly if our universe were reduced from three to two dimensions, it would eliminate the possibility of most of the complexity that we observe. Imagine for example, the limitations that two-dimensionality would place on the design of neural networks (if "wires" could not cross). If we were to further reduce the dimensionality of our universe to just one dimension, it would probably completely preclude the possibility of the existence of life.

It follows from these thoughts, that restricting digital life to a presumably one-dimensional memory space places a tragic limitation on the richness that might evolve. Clearly it would be liberating to move digital organisms into a two or three-dimensional space. The flaw in all of this logic derives from the erroneous supposition that computer memory is a Euclidean space.

To think of memory as Euclidean is to fail to understand its natural topology, and is an example of one of the greatest pitfalls in the enterprise of synthetic biology: to transfer a concept from organic life to synthetic life in a way that is "un-natural" for the artificial medium. The fundamental principal of the approach I am advocating is *to respect the nature of the medium into which life is being inoculated, and to find the natural form of life in that medium*, without inappropriately trying to make it like organic life.

The desire to increase the richness of memory topology is commendable, however this can be achieved without forcing the memory into an un-natural Euclidean topology. Let us reflect a little more on the structure of cyberspace. Thus far we have only considered the topology of flat memory. Let us consider segmented memory such as is found with the notorious Intel 80X86 design. With this design, you may treat any arbitrarily chosen block of 64K bytes as flat, and all pairs of locations within that block are equidistant. However, once the block is chosen, all memory outside of that block is about twice as far away.

Cache memory is designed to be accessed more rapidly than RAM memory, thus pairs of points within cache memory are closer than pairs of points within RAM memory. The distance between a point in cache and a point in RAM would be an intermediate distance. The access time to memory on disks is much greater than for RAM memory, thus the distance between points on disk is very great, and the distance between RAM and disk is again intermediate (but still very great). CPU registers represent a small amount of memory locations, between which data can move very rapidly, thus these registers can be considered to be very close together.

For networked computer systems, information can move between the memories of the computers on the net, and the distances between these memories is again the transfer time. If the CPU, cache, RAM and disk memories of a network of computers are all considered together, they present a very complex memory topology. Similar considerations apply to massively parallel computers which have memories connected in a variety of topologies. Utilizing this complexity moves us in the direction of what has been intended by creating Euclidean memories for digital organisms, but does so while fully respecting the natural topology of computer memories.

11 Ecological Context

11.1 The Living Environment

Some rain forests in the Amazon region occur on white sand soils. In these locations, the physical environment consists of clean white sand, air, falling water, and sunlight. Embedded within this relatively simple physical context we find one of the most complex ecosystems on earth, containing hundreds of thousands of species. These species do not represent hundreds of thousands of adaptations to the physical environment. Most of the adaptations of these species are to the other living organism. The forest creates its own environment.

Life is an auto-catalytic process that builds on itself. Ecological communities are complex webs of species, each living off of others, and being lived off of by others. The system is self-constructing, self-perpetuating, and feeds on itself. Living organisms interface with the non-living physical environment, exchanging materials with it, such as oxygen, carbon-dioxide, nitrogen, and various minerals. However, in the richest ecosystems, the living components of the environment predominate over the physical components.

With living organisms constituting the predominant features of the environment, the evolutionary process is primarily concerned with adaptation to the living environment. Thus ecological interactions are an important driving force for evolution. Species evolve adaptations to exploit other species (to eat them, to parasitize them, to climb on them, to nest on them, to catch a ride on them, etc.) and to defend against such exploitation where it creates a burden.

This situation creates an interesting dynamic. Evolution is predominantly concerned with creating and maintaining adaptations to living organisms which are themselves evolving. This generates evolutionary races among groups of species that interact ecologically. These races can catalyze the evolution of upwardly spiraling complexity as each species evolves to overcome the adaptations of the others. Imagine for example, a predator and prey, each evolving to increase its speed and agility, in capturing prey, or in evading capture. This coupled evolutionary race can lead to increasingly complex nervous systems in the evolving predator and prey species.

This mutual evolutionary dynamic is related to the Red Queen hypothesis [95], named after the Red Queen from Alice in Wonderland. This hypothesis suggests that in the face of a changing environment, organisms must evolve as fast as they can in order to simply maintain their current state of adaptation. "In order to get anywhere you must run twice as fast as that" [11].

If organisms only had to adapt to the non-living environment, the race would not be so urgent. Species would only need to evolve as fast as the relatively gradual changes in the geology and climate. However, given that the species that comprise the environment are themselves evolving, the race becomes rather hectic. The pace is set by the maximal rate that species may change through evolution, and it becomes very difficult to actually get ahead. A maximal rate of evolution is required just to keep from falling behind.

What all of this discussion points to is the importance of embedding evolving synthetic organisms into a context in which they may interact with other evolving organisms. A counter example is the standard implementations of genetic algorithms in which the evolving entities interact only with the fitness function, and never "see" the other entities in the population. Many interesting behavioral, ecological and evolutionary phenomena can only emerge from interactions among the evolving entities.

11.2 Diversity

Major temporal and spatial patterns of organic diversity on earth remain largely unexplained, although there is no lack of theories. Diversity theories suggest fundamental ecological and evolutionary principles which may apply to synthetic life. In general these theories relate to synthetic life in two ways: 1) They suggest factors which may be critical to the auto-catalytic increase of diversity and complexity in an evolving system. It may be necessary then to introduce these factors into an artificial system to generate increasing diversity and complexity. 2) Because it will be possible to manipulate the presence, absence, or state of these factors in an artificial system, the artificial system may provide an experimental framework for examining evolutionary and ecological processes that influence diversity.

The Gaussian principle of competitive exclusion states that no two species that occupy the same niche can coexist. The species which is the superior competitor will exclude the inferior competitor. The principle has been experimentally demonstrated in the laboratory, and is considered theoretically sound. However, natural communities widely flaunt the principle. In tropical rain forests several hundred species of trees coexist without any dominant species in the community. All species of trees must spread their leaves to collect light and their roots to absorb water and nutrients. Evidently there are not several hundred niches for trees in the same habitat. Somehow the principle of competitive exclusion is circumvented.

There are many theories on how competitive exclusion may be circumvented. One leading theory is that periodic disturbance at the proper level sets back the process of competitive exclusion, allowing more species to coexist [37, 38, 39]. There is substantial evidence that moderate levels of disturbance can increase diversity. In a digital community, disturbance might take the form of freeing blocks of memory that had been filled with digital organisms. It would be very easy to experiment with differing frequencies and patch sizes of disturbance.

One theory to explain the great increase in diversity and complexity in the Cambrian explosion [85] states that its evolution was driven by ecological interactions, and that it was originally sparked by the appearance of the first organisms that ate other organisms (heterotrophs). As long as all organisms were autotrophs (produce their own food, like plants), there was only room for a few species. In a community with only one trophic level, the most successful competitors would dominate. The process of competitive exclusion would keep diversity low.

However, when the first herbivore (organisms that eat autotrophs) appeared it would have been selected to prefer the most common species of algae, thereby preventing any species of algae from dominating. This opens the way for more species of algae to coexist. Once the "heterotroph barrier" had been crossed, it would be simple for carnivores to arise, imposing a similar diversifying effect on herbivores. With more species of algae, herbivores may begin to specialize on different species of algae, enhancing diversification in herbivores. The theory states that the process was auto-catalytic, and set off an explosion of diversity.

One of the most universal of ecological laws is the species area relationship [52]. It has been demonstrated that in a wide variety of contexts, the number of species occupying an "area" increases with the area. The number of species increases in proportion to the area raised to a power between 0.1 and 0.3. $S = KA^z$, where $0.1 < z < 0.3$. The effect is thought to result from the equilibrium species number being determined by a balance between the arrival (by immigration or speciation) and local extinction of species. The likelihood of extinction is greater in small areas because they support smaller populations, for which a fluctuation to a size of zero is more likely. If this effect holds for digital organisms it suggests that larger amounts of memory will generate greater diversity.

11.3 Ecological Attractors

While there are no completely independent instances of natural evolution on Earth, there are partially independent instances. Where major diversifications have occurred, isolated either by geography or epoch from other similar diversifications, we have the opportunity to observe whether evolution tends to take the same routes or is always quite different. We can compare the marsupial mammals of Australia to the placental mammals of the rest of the world, or the modern mammals to the reptiles of the age of dinosaurs, or the bird fauna of the Galapagos to the bird faunas of less isolated islands.

What we find again and again is an uncanny convergence between these isolated faunas. This suggests that there are fairly strong ecological attractors which evolution will tend to fill, more or less regardless of the developmental and physiological systems that are evolving. In this view, chance and history still play a role, in determining what kind of organism fills the array of ecological attractors (reptiles, mammals, birds, etc.), but the attractors themselves may be a property of the system and not as variable. Synthetic systems may also contain fairly well defined ecological forms which may be filled by a wide variety of specific kinds of organisms.

Given their evident importance in moving evolution, it is important to include ecological interactions in synthetic instantiations of life. It is encouraging to observe that in the Tierra model, ecological interactions, and the corresponding evolutionary races emerged spontaneously. It is possible that any medium into which evolution is inoculated will contain an array of "ecological attractors" into which evolution will easily flow.

12 Cellularity

Cellularity is one of the fundamental properties of organic life, and can be recognized in the fossil record as far back as 3.6 billion years. The cell is the original individual, with the cell membrane defining its limits and preserving its chemical integrity. An analog to the cell membrane is probably needed in digital organisms in order to preserve the integrity of the informational structure from being disrupted by the activity of other organisms.

The need for this can be seen in AL models such as cellular automata where virtual state machines pass through one another [48], or in core wars type simulations where coherent structures that arise demolish one another when they come into contact [68, 69]. An analog to the cell membrane that can be used in the core wars type of simulation is memory allocation. An artificial "cell" could be defined by the limits of an allocated block of memory. Free access to the memory within the block could be limited to processes within the block. Processes outside of the block would have limited access, according the rules of "semi-permeability"; for example they might be allowed to read and execute but not write.

13 Multi-cellularity

Multi-celled digital organisms are parallel processes. By attempting to synthesize multi-celled digital organisms we can simultaneously explore the biological issues surrounding the evolutionary transition from single-celled to multi-celled life, and the computational issues surrounding the design of complex parallel software.

13.1 Biological Perspective — Cambrian Explosion

Life appeared on earth somewhere between three and four billion years ago. While the origin of life is generally recognized as an event of the first order, there is another event in the history of life that is less well known but of comparable significance. The origin of biological diversity and at the same time of complex macroscopic multi-cellular life, occurred abruptly in the Cambrian explosion 600 million years ago. This event involved a riotous diversification of life forms. Dozens of phyla appeared suddenly, many existing only fleetingly, as diverse and sometimes bizarre ways of life were explored in a relative ecological void [28, 65].

The Cambrian explosion was a time of phenomenal and spontaneous increase in the complexity of living systems. It was the process initiated at this time that led to the evolution of immune systems, nervous systems, physiological systems, developmental systems, complex morphology, and complex ecosystems. To understand the Cambrian explosion is to understand the evolution of complexity. If the history of organic life can be used as a guide, the transition from single celled to multi-celled organisms should be critical in achieving a rich diversity and complexity of synthetic life forms.

13.2 Computational Perspective — Parallel Processes

It has become apparent that the future of high performance computing lies with massively parallel architectures. There already exist a variety of parallel hardware platforms, but our ability to fully utilize the potential of these machines is constrained by our inability to write software of a sufficient complexity.

There are two fairly distinctive kinds of parallel architecture in use today: SIMD (single instruction multiple data) and MIMD (multiple instruction multiple data). In the SIMD architecture, the machine may have thousands of processors, but in each CPU cycle, all of the processors must execute the same instruction, although they may operate on different data. It is relatively easy to write software for this kind of machine, since what is essentially a normal sequential program will be broadcast to all the processors.

In the MIMD architecture, there exists the capability for each of the hundreds or thousands of processors to be executing different code, but to have all of that activity coordinated on a common task. However, there does not exist an art for writing this kind of software, at least not on a scale involving more than a few parallel processes. In fact it seems unlikely that human programmers will ever be capable of actually writing software of such complexity.

13.3 Evolution as a Proven Route

It is generally recognized that evolution is the only process with a proven ability to generate intelligence. It is less well recognized that evolution also has a proven ability to generate parallel software of great complexity. In making life a metaphor for computation we will think of the genome, the DNA, as the program, and we will think of each cell in the organism as a processor (CPU). A large multi-celled organism like a human contains trillions of cells/processors. The genetic program contains billions of nucleotides/instructions.

In a multi-celled organism, cells are differentiated into many cell types such as brain cells, muscle cells, liver cells, kidney cells, etc. The cell types just named are actually general classes of cell types within which there are many sub-types. However, when we specify the ultimate indivisible types, what characterizes a type is the set of genes it expresses. Different cell types express different combinations of genes. In a large organism, there will be a very large number of cells of most types. All cells of the same type express the same genes.

The cells of a single cell type can be thought of as exhibiting parallelism of the SIMD kind, as they are all running the same "program" by expressing the same genes. Cells of different cell types exhibit MIMD parallelism as they run different code by expressing different genes. Thus large multi-cellular organisms display parallelism on an astronomical scale, combining both SIMD and MIMD parallelism into a beautifully integrated whole. From these considerations it is evident that evolution has a proven ability to generate massively parallel software embedded in wetware. The computational goal of evolving multi-cellular digital organisms is to produce such software embedded in hardware.

13.4 Fundamental Definition

In order to conceptualize multi-cellularity in the context of an artificial medium, we must have a very fundamental definition which is independent of the context of the medium. We generally think of the defining property of multi-cellularity as being that the cells stick together, forming a physically coherent unit. However, this is a spatial concept based on Euclidean geometry, and therefore is not relevant to non-Euclidean cyberspace.

While physical coherence might be an adequate criteria for recognizing multi-cellularity in organic organisms, it is not the property that allows multi-cellular organisms to become large and complex. There are algae that consist of strands of cells that are stuck together, with each cell being identical to the next. This is a relatively limiting form of multi-cellularity because there is no differentiation of cell types. It is the specialization of functions resulting from cell differentiation that has allowed multi-cellular organisms to attain large sizes and great complexity. It is differentiation that has generated the MIMD style of parallelism in organic software.

From an evolutionary perspective, an important characteristic of multi-cellular organisms is their genetic unity. All the cells of the individual contain the same genetic material as a result of having a common origin from a single egg cell (some small genetic differences may arise due to somatic mutations; in some species new individuals arise from a bud of tissue rather than a single cell). Genetic unity through common origin, and differentiation are critical qualities of multi-cellularity that may be transferable to media other than organic chemistry.

Buss [9] provides a provocative discussion of the evolution of multi-cellularity, and explores the conflicts between selection at the levels of cell lines and of individuals. From his discussion the following idea emerges (although he does not explicitly state this idea, in fact he proposes a sort of inverse of this idea, p. 65): the transition from single to multi-celled existence involves the extension of the control of gene regulation by the mother cell to successively more generations of daughter cells.

In organic cells, genes are regulated by proteins contained in the cytoplasm. During early embryonic development in animals, an initially very large fertilized egg cell undergoes cell division with no increase in the overall size of the embryo. The large cell is simply partitioned into many smaller cells, and all components of the cytoplasm are of maternal origin. By preventing several generations of daughter cells from producing any cytoplasmic regulatory components, the mother gains control of the course of differentiation, and thereby creates the developmental process. In single celled organisms by contrast, after each cell division, the daughter cell produces its own cytoplasmic regulatory products, and determines its own destiny independent of the mother cell.

Complex digital organisms will be self replicating algorithms, consisting of many distinct processes dedicated to specific tasks (e.g., locating free memory, mates or other resources; defense; replicating the code). These processes must be coordinated and regulated, and may be divided among several cells specialized for specific functions. If the mother cell can influence the regulation of the processes of the daughter, so as to force the daughter cell to specialize in function and express only a portion of its full genetic potentiality, then the essence of multi-cellularity will be achieved.

13.5 Computational Implementation

The discussion above suggests that the critical feature needed to allow the evolution of multi-cellularity is for a cell to be able to influence the expression of genes by its daughter cell. In the digital context, this means that a cell must be able to influence what code is executed by its daughter cell.

If we assume that in digital organisms, as in organic ones, all cells in an individual contain the same genetic material, then the desired regulatory mechanism can be achieved most simply by allowing the mother cell to affect the context of the CPU of the daughter cell at the time that the cell is "born". Most importantly, the mother cell needs to be able to set the address of the instruction pointer of the daughter cell at birth, which will determine where the daughter cell will begin executing its code. Beyond that, additional influence can be achieved by allowing the mother cell to place values in the registers of the daughter's CPU.

A large digital genome may contain several sections of code that are "closed" in the sense that one section of code will not pass control of execution to another. Thus if execution begins in one of these sections of code, the other sections will never be expressed. This type of genetic organization, coupled with the ability of the mother cell to determine where the daughter cell begins executing, could provide a mechanism of gene regulation suitable for causing the differentiation of cells in a multi-cellular digital organism.

Other schemes for the regulation of code expression are also possible. For example, digital computers commonly have three protection states available for the memory: read, write and execute. If the code of the genome were provided with execute protection, it would provide a means of suppression of the execution of code in the protected region of the genome.

13.6 Digital "Neural Networks" — Natural Artificial Intelligence

One of the greatest challenges in the field of computer science is to produce computer systems that are "intelligent" in some way. This might involve for example, the creation of a system for the guidance of a robot which is capable of moving freely in a complex environment, seeking, recognizing and manipulating a variety of objects. It might involve the creation of a system capable of communicating with humans in natural spoken human language, or of translating between human languages.

It has been observed that natural systems with these capabilities are controlled by nervous systems consisting of large numbers of neurons interconnected by axons and dendrites. Borrowing from nature, a great deal of work has gone into setting up "neural networks" in computers [18, 33]. In these systems, a collection of simulated "neurons" are created, and connected so that they can pass messages. The learning that takes place is accomplished by adjusting the "weights" of the connections.

Organic neurons are essentially analog devices, thus when neural networks are implemented on computers, they are digital emulations of analog devices. There is a certain inefficiency involved in emulating an analog device on a digital computer. For this reason, specialized analog hardware has been developed for the more efficient implementation of artificial neural nets [60].

Neural networks, as implemented in computers, either digital or analog, are intentional mimics of organic nervous systems. They are designed to function like natural neural networks in many details. However, natural neural networks represent the solution found by evolution to the problem of creating a control system based on organic chemistry. Evolution works with the physics and chemistry of the medium in which it is embedded.

The solution that evolution found to the problem of communication between organic cells is chemical. Cells communicate by releasing chemicals that bind to and activate receptor molecules on target cells. Working within this medium, evolution created neural nets. Inter-cellular chemical communication in neural nets is "digital" in the sense that chemical messages are either present or not present (on or off). In this sense, a single chemical message carries only a single bit of information. More detailed information can be derived from the temporal pattern of the messages, and also the context of the message. The context can include where on the target cell body the message is applied (which influences its "weight"), and what other messages are arriving at the same time, with which the message in question will be integrated.

It is hoped that evolving multi-cellular digital organisms will become very complex, and will contain some kind of control system that fills the functional role of the nervous system. While it seems likely that the digital nervous system would consist of a network of communicating “cells”, it seem unlikely that this would bear much resemblance to conventional neural networks.

Compare the mechanism of inter-cellular communication in organic cells (described above), to the mechanisms of inter-process communication in computers. Processes transmit messages in the form of bit patterns, which may be of any length, and so which may contain any amount of information. Information need not be encoded into the temporal pattern of impulse trains. This fundamental difference in communication mechanisms between the digital and the organic mediums must influence the course that evolution will take as it creates information processing systems in the two mediums.

It seems highly unlikely that evolution in the digital context would produce information processing systems that would use the same forms and mechanisms as natural neural nets (e.g., weighted connections, integration of incoming messages, threshold triggered all or nothing output, thousands of connections per unit). The organic medium is a physical/chemical medium, whereas the digital medium is a logical/informational medium. That observation alone would suggest that the digital medium is better suited to the construction of information processing systems.

If this is true, then it may be possible to produce digitally based systems that have functionality equivalent to natural neural networks, but which have a much greater simplicity of structure and process. Given evolution's ability to discover the possibilities inherent in a medium, and it's complete lack of preconceptions, it would be very interesting to observe what kind of information processing systems evolution would construct in the digital medium. If evolution is capable of creating network based information processing systems, it may provide us with a new paradigm for digital “connectionism”, that would be more natural to the digital medium than simulations of natural neural networks.

14 Digital Husbandry

Digital organisms evolving freely by natural selection do no “useful” work. Natural evolution tends to the selfish needs of perpetuating the genes. We can not expect digital organisms evolving in this way to perform useful work for us, such as guiding robots or interpreting human languages. In order to generate digital organisms that function as useful software, we must guide their evolution through artificial selection, just as humans breed dogs, cattle and rice. Some experiments have already been done with using artificial selection to guide the evolution of digital organisms for the performance of “useful” tasks [1, 89, 91]. I envision two approaches to the management of digital evolution: digital husbandry, and digital genetic engineering.

Digital husbandry is an analogy to animal husbandry. This technique would be used for the evolution of the most advanced and complex software, with intelligent capabilities. Correspondingly, this technique is the most fanciful. I would begin by allowing multi-cellular digital organisms to evolve freely by natural selection. Using strictly natural selection, I would attempt to engineer the system to the threshold of the computational analog of the Cambrian explosion, and let the diversity and complexity of the digital organisms spontaneously explode.

One of the goals of this exercise would be to allow evolution to find the natural forms of complex parallel digital processes. Our parallel hardware is still too new for human programmers to have found the best way to write parallel software. And it is unlikely that human programmers will ever be capable of writing software of the complexity that the hardware is capable of running. Evolution should be able to show us the way.

It is hoped that this would lead to highly complex digital organisms, which obtain and process information, presumably predominantly about other digital organisms. As the complexity of the evolving system increases, the organisms will process more complex information in more complex ways, and take more complex actions in response. These will be information processing organisms living in an informational environment.

It is hoped that evolution by natural selection alone would lead to digital organisms which while doing no "useful" work, would none-the-less be highly sophisticated parallel information processing systems. Once this level of evolution has been achieved, then artificial selection could begin to be applied, to enhance those information processing capabilities that show promise of utility to humans. Selection for different capabilities would lead to many different breeds of digital organisms with different uses. Good examples of this kind of breeding from organic evolution are the many varieties of domestic dogs which were derived by breeding from a single species, and the vegetables cabbage, kale, broccoli, cauliflower, and brussels sprouts which were all produced by selective breeding from a single species of plant.

Digital genetic engineering would normally be used in conjunction with digital husbandry. This consists of writing a piece of application code and inserting it into the genome of an existing digital organism. A technique being used in organic genetic engineering today is to insert genes for useful proteins into goats, and to cause them to be expressed in the mammary glands. The goats then secrete large quantities of the protein into the milk, which can be easily removed from the animal. We can think of our complex digital organisms as general purpose animals, like goats, into which application codes can be inserted to add new functionalities, and then bred through artificial selection to enhance or alter the quality of the new functions.

In addition to adding new functionalities to complex digital organisms, digital genetic engineering could be used for achieving extremely high degrees of optimization in relatively small but heavily used pieces of code. In this approach, small pieces of application code could be inserted into the genomes of simple digital organisms. Then the allocation of CPU cycles to those organisms would be based on the performance of the inserted code. In this way, evolution could optimize those codes, and they could be returned to their applications. This technique would be used for codes that are very heavily used such as compiler constructs, or central components of the operating system.

15 Living Together

> I'm glad they're not real, because if they were, I would have to feed them and they would be all over the house.
>
> — Isabel Ray.

Evolution is an extremely selfish process. Each evolving species does whatever it can to insure its own survival, with no regard for the well-being of other genetic groups (potentially with the exception of intelligent species). Freely evolving autonomous artificial entities should be seen as potentially dangerous to organic life, and should always be confined by some kind of containment facility, at least until their real potential is well understood. At present, evolving digital organisms exist only in virtual computers, specially designed so that their machine codes are more robust than usual to random alterations. Outside of these special virtual machines, digital organisms are merely data, and no more dangerous than the data in a data base or the text file from a word processor.

Imagine however, the problems that could arise if evolving digital organisms were to colonize the computers connected to the major networks. They could spread across the network like the infamous internet worm [2, 8, 83, 84]. When we attempted to stop them, they could evolve mechanisms to escape from our attacks. It might conceivably be very difficult to eliminate them. However, this scenario is highly unlikely, as it is probably not possible for digital organisms to evolve on normal computer systems. While the supposition remains untested, normal machine languages are probably too brittle to support digital evolution.

Evolving digital organisms will probably always be confined to special machines, either real or virtual, designed to support the evolutionary process. This does not mean however, that they are necessarily harmless. Evolution remains a self-interested process, and even the interests of confined digital organisms may conflict with our own. For this reason it is important to restrict the kinds of peripheral devices that are available to autonomous evolving processes.

This conflict was taken to its extreme in the movie Terminator 2. In the imagined future of the movie, computer designers had achieved a very advanced chip design, which had allowed computers to autonomously increase their own intelligence until they became fully conscious. Unfortunately, these intelligent computers formed the "sky-net" of the United States military. When the humans realized that the computers had become intelligent, they decided to turn them off. The computers viewed this as a threat, and defended themselves by using one of their peripheral devices: nuclear weapons.

Relationships between species can however, be harmonious. We presently share the planet with millions of freely evolving species, and they are not threatening us with destruction. On the contrary, we threaten them. In spite of the mindless and massive destruction of life being caused by human activity, the general pattern in living communities is one of a network of inter-dependencies.

More to the point, there are many species with which humans live in close relationships, and whose evolution we manage. These are the domesticated plants and animals that form the basis of our agriculture (cattle, rice), and who serve us as companions (dogs, cats, house plants). It is likely that our relationship with digital organisms will develop along the same two lines.

There will likely be carefully bred digital organisms developed by artificial selection and genetic engineering that perform intelligent data processing tasks. These would subsequently be "neutered" so that they can not replicate, and the eunuchs would be put to work in environments free from genetic operators. We are also likely to see freely evolving and/or partially bred digital ecosystems contained in the equivalent of digital aquariums (without dangerous peripherals) for our companionship and aesthetic enjoyment.

While this paper has focused on digital organisms, it is hoped that the discussions be taken in the more general context of the possibilities of any synthetic forms of life. The issues of living together become more critical for synthetic life forms implemented in hardware or wetware. Because these organisms would share the same physical space that we occupy, and possibly consume some of the same material resources, the potential for conflict is much higher than for digital organisms.

At the present, there are no self-replicating artificial organisms implemented in either hardware or wetware (with the exception of some simple organic molecules with evidently small and finite evolutionary potential [25, 36, 66]). However, there are active attempts to synthesize RNA molecules capable of replication [4, 42], and there is much discussion of the future possibility of self-replicating nano-technology and macro-robots. I would strongly urge that as any of these technologies approaches the point where self-replication is possible, the work be moved to specialized containment facilities. The means of containment will have to be handled on a case-by-case basis, as each new kind of replicating technology will have its own special properties.

There are many in the artificial life movement who envision a beautiful future in which artificial life replaces organic life, and expands out into the universe [49, 50, 62, 63, 64]. The motives vary from a desire for immortality to a vision of converting virtually all matter in the universe to living matter. It is argued that this transition from organic to metallic based life is the inevitable and natural next step in evolution.

The naturalness of this step is argued by analogy with the supposed genetic takeovers in which nucleic acids became the genetic material taking over from clays [10], and cultural evolution took over from DNA based genetic evolution in modern humans. I would point out that whatever nucleic acids took over from, it marked the origin of life more than the passing of a torch. As for the supposed transition from genetic to cultural evolution, the truth is that genetic evolution remains intact, and has had cultural evolution layered over it rather than being replaced by it.

The supposed replacement of genetic by cultural evolution remains a vision of a brave new world, which has yet to materialize. Given the ever increasing destruction of nature, and human misery and violence being generated by human culture, I would hesitate to place my trust in the process as the creator of a bright future. I still trust in organic evolution, which created the beauty of the rainforest through billions of years of evolution. I prefer to see artificial evolution confined to the realm of cyberspace, where we can more easily coexist with it without danger, using it to enhance our lives without having to replace ourselves.

As for the expansion of life out into the universe, I am confident that this can be achieved by organic life aided by intelligent non-replicating machines. And as for immortality, our unwillingness to accept our own mortality has been a primary fuel for religions through the ages. I find it sad that Artificial Life should become an outlet for the same sentiment. I prefer to achieve immortality in the old fashioned organic evolutionary way, through my children. I hope to die in my patch of Costa Rican rain forest, surrounded by many thousands of wet and squishy species, and leave it all to my daughter. Let them set my body out in the jungle to be recycled into the ecosystem by the scavengers and decomposers. I will live on through the rain forest I preserved, the ongoing life in the ecosystem into which my material self is recycled, the memes spawned by my scientific works, and the genes in the daughter that my wife and I created.

16 Challenges

For well over a century, evolution has remained a largely theoretical science. Now new technologies have allowed us to inoculate natural evolution into artificial media, converting evolution into an experimental and applied science, and at the same time, opening Pandora's box. This creates a variety of challenges which have been raised or alluded to in the preceding essay, and which will be summarized here.

Respecting the Medium If the objective is to instantiate rather than simulate life, then care must be taken in transferring ideas from natural to artificial life forms. Preconceptions derived from experience with natural life may be inappropriate in the context of the artificial medium. Getting it right is an art, which likely will take some skill and practice to develop.

However, respecting the medium is only one approach, which I happen to favor. I do not wish to imply that it is the only valid approach. It is too early to know which approach will generate the best results, and I hope that other approaches will be developed as well. I have attempted to articulate clearly this "natural" approach to synthetic life, so that those who choose to follow it may achieve greater consistency in design through a deeper understanding of the method.

Understanding Evolvability Attempts are now underway to inoculate evolution into many artificial systems, with mixed results. Some genetic languages evolve readily, while others do not. We do not yet know why, and this is a fundamental and critically important issue. What are the elements of evolvability? Efforts are needed to directly address this issue. One approach that would likely be rewarding would be to systematically identify features of a class of languages (such as machine languages), and one by one, vary each feature, to determine how evolvability is affected by the state of each feature.

Creating Organized Sexuality Organized sexuality is important to the evolutionary process. It is the basis of the species concept, and while remaining something of an enigma in evolutionary theory, clearly is an important facilitator of the evolutionary process. Yet this kind of sexuality still has not been implemented in a natural way in synthetic life systems. It is important to find ways of orchestrating organized sexuality in synthetic systems such as digital organisms, in a way in which it is not mandatory, and in which the organisms must carry out the process through their own actions.

Creating Multi-cellularity In organic life, the transition from single to multi-celled forms unleashed a phenomenal explosion of diversity and complexity. It would seem then that the transition to multi-cellular forms could generate analogous diversity and complexity in synthetic systems. In the case of digital organisms, it would also lead to the evolution of parallel processes, which could provide us with new paradigms for the design of parallel software. The creation of multi-celled digital organisms remains an important challenge.

Controlling Evolution Humans have been controlling the evolution of other species for tens of thousands of years. This has formed the basis of agriculture, through the domestication of plants and animals. The fields of genetic algorithms [27, 35], and genetic programming [47] are based on controlling the evolution of computer programs. However, we still have very little experience with controlling the evolution of self-replicating computer programs, which is more difficult. In addition, breeding complex parallel programs is likely to bring new challenges. Developing technologies for managing the evolution of complex software will be critical for harnessing the full potential of evolution for the creation of useful software.

Living Together If we succeed in harnessing the power of evolution to create complex synthetic organisms capable of sophisticated information processing and behavior, we will be faced with the problems of how to live harmoniously with them. Given evolution's selfish nature and capability to improve performance, there exists the potential for a conflict arising through a struggle for dominance between organic and synthetic organisms. It will be a challenge to even agree on what the most desirable outcome should be, and harder still to accomplish it. In the end the outcome is likely to emerge from the bottom up through the interactions of the players, rather than being decided through rational deliberations.

17 Acknowledgements

This work was supported by grants CCR-9204339 and BIR-9300800 from the United States National Science Foundation, a grant from the Digital Equipment Corporation, and by the Santa Fe Institute, Thinking Machines Corp., IBM, and Hughes Aircraft. This work was conducted while at: School of Life & Health Sciences, University of Delaware, Newark, Delaware, 19716, USA, ray@udel.edu; and Santa Fe Institute, 1660 Old Pecos Trail, Suite A, Santa Fe, New Mexico, 87501, USA, ray@santafe.edu.

References

[1] Adami, Chris. Unpublished. Learning and complexity in genetic auto-adaptive systems. Caltech preprint: MAP – 164, One of the Marmal Aid Preprint Series In Theoretical Nuclear Physics, October 1993. Adami has used the input-output facilities of the new Tierra languages to feed data to creatures, and select for responses that result from simple computations, not contained in the seed genome. Contact: chris@almach.caltech.edu

[2] Anonymous. 1988. Worm invasion. Science 11-11-88: 885.

[3] Barton-Davis, Paul. Unpublished. Independent implementation of the Tierra system, contact: pauld@cs.washington.edu.

[4] Beaudry, Amber A., and Gerald F. Joyce. 1992. Directed evolution of an RNA enzyme. Science 257: 635–641.

[5] Bell, Graham. 1982. The masterpiece of nature: the evolution and genetics of sexuality. Berkeley: University of California Press.

[6] Benner, Steven A., Andrew D. Ellington, and Andreas Tauer. 1989. Modern metabolism as a palimpsest of the RNA world. Proc. Natl. Acad. Sci. U.S.A. 86: 7054–7058.

[7] Brooks, Rodney. Unpublished. Brooks has created his own Tierra-like system, which he calls Sierra. In his implementation, each machine instruction consists of an opcode and an operand. Successive instructions overlap, such that the operand of one instruction is interpreted as the opcode of the next instruction. Contact: brooks@ai.mit.edu

[8] Burstyn, Harold L. 1990. RTM and the worm that ate internet. Harvard Magazine 92(5): 23–28.

[9] Buss, Leo W. 1987. The evolution of individuality. Princeton, NJ: Princeton University Press. Pp. 203.

[10] Cairn-Smith, A. G. 1985. Seven clues to the origin of life. Cambridge: Cambridge University Press.

[11] Carroll, L. 1865. Through the Looking-Glass. London: MacMillan.

[12] Chao, Lin, Christopher Vargas, Brian B. Spear, and Edward C. Cox. 1983. Transposable elements as mutator genes in evolution. Nature 303: 633–635.

[13] Charlesworth, B. 1976. Recombination modification in a fluctuating environment, Genetics 83: 181–195.

[14] Darwin, Charles. 1859. On the origin of species by means of natural selection or the preservation of favored races in the struggle for life. London: Murray.

[15] Davidge, Robert. 1992. Processors as organisms. CSRP 250. School of Cognitive and Computing Sciences, University of Sussex. Presented at the ALife III conference. Contact: robertd@cogs.susx.ac.uk

[16] Davidge, Robert. 1993. Looping as a means to survival: playing Russian roulette in a harsh environment. *In*: Self organization and life: from simple rules to global complexity, proceedings of the second European conference on artificial life. Contact: robertd@cogs.susx.ac.uk

[17] Davidson, Eric H., and Roy J. Britten. 1979. Regulation of gene expression: Possible role of repetitive sequences. Science 204: 1052–1059.

[18] Dayhoff, Judith. 1990. Neural network architectures. New York: Van Nostrand Reinhold. Pp. 259.

[19] DeAngelis, D., and L. Gross [eds]. 1992. Individual based models and approaches in ecology. New York: Chapman and Hill.

[20] de Groot, Marc. Unpublished. Primordial soup, a Tierra-like system that has the additional ability to spawn self-reproducing organisms from a sterile soup. Contact: marc@kg6kf.ampr.org, marc@toad.com, marc@remarque.berkeley.edu

[21] Doolittle, W. Ford, and Carmen Sapienza. 1980. Selfish genes, the phenotype paradigm and genome evolution. Nature 284: 601–603.

[22] Eigen, Manfred. 1993. Viral quasispecies. Scientific American 269(1): 32–39. July 1993.

[23] Farmer, J. D. & Belin, A. Artificial life: the coming evolution. Proceedings in celebration of Murray Gell-Mann's 60th Birthday. Cambridge: University Press. (Reprinted in Artificial Life II. Pp. 815–840.)

[24] Feferman, Linda. 1992. Simple rules... complex behavior [video]. Santa Fe, NM: Santa Fe Institute. Contact: fef@santafe.edu, 0005851689@mcimail.com

[25] Feng, Q., Park, T. K. & Rebek, J. 1992. Science 254: 1179–1180.

[26] Ghiselin, Michael. 1974. The economy of nature and the evolution of sex. Berkeley: University of California Press.

[27] Goldberg, D. E. 1989. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.

[28] Gould, Steven J. 1989. Wonderful life. W. W. Norton & Company, Inc. Pp. 347.

[29] Gray, James. Unpublished. Natural selection of computer programs. This may have been the first Tierra-like system, but evolving real programs on a real rather than a virtual machine, and predating Tierra itself: "I have attempted to develop ways to get computer programs to function like biological systems subject to natural selection.... I don't think my systems are models in the usual sense. The programs have really competed for resources, reproduced, run, and 'died'. The resources consisted primarily of access to the CPU and partition space.... On a PDP11 I could have a population of programs running simultaneously." Contact: Gray.James_L+@northport.va.gov

[30] Green, Melvin M. 1988. Mobile DNA elements and spontaneous gene mutation. *In* M. E. Lambert, J. F. McDonald, I. B. Weinstein [eds.]: Eukaryotic transposable elements as mutagenic agents. Pp. 41–50. Banbury Report 30, Cold Spring Harbor Laboratory.

[31] Halvorson, Herlyn O., and Albert Monroy. 1985. The origin and evolution of sex. New York: A. R. Liss.

[32] Hapgood, Fred. 1979. Why males exist: an inquiry into the evolution of sex. New York: William Morrow.

[33] Hertz, John, Anders Krogh, and Richard G. Palmer. 1991. Introduction to the theory of neural computation. Addison-Wesley Publishing Co. Pp. 327.

[34] Hogeweg, P. 1989. Mirror beyond mirror: puddles of life. *In*: Langton, C. [ed], Artificial Life, Santa Fe Institute Studies in the Sciences of Complexity, vol. VI, 297–316. Redwood City, CA: Addison-Wesley.

[35] Holland, John Henry. 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence (Univ. of Michigan Press, Ann Arbor).

[36] Hong, J. I., Feng, Q., Rotello, V. & Rebek, J. 1992. Competition, cooperation, and mutation: improving a synthetic replicator by light irradiation. Science 255: 848–850.

[37] Huston, Michael. 1979. A general hypothesis of species diversity, American Naturalist 113: 81–101.

[38] Huston, Michael. 1992. Biological diversity and human resources. Impact of science on society 166: 121–130.

[39] Huston, Michael. 1993. Biological diversity: the coexistence of species on changing landscapes. Cambridge University Press.

[40] Huston, M., DeAngelis, D., and Post, W. 1988. New computer models unify ecological theory. Bioscience 38(10): 682–691.

[41] Jelinek, Warren R., and Carl W. Schmid. 1982. Repetitive sequences in eukaryotic DNA and their expression. Annual Reviews of Biochemistry 51: 813–844.

[42] Joyce, Gerald F. 1992. Directed molecular evolution. Scientific American, December 1992: 90–97.

[43] Kampis, George. 1993. Coevolution in the computer: the necessity and use of distributed code systems. Printed in the ECAL93 proceedings, Brussels. Contact: gk@cfnext.physchem.chemie.uni-tuebingen.de

[44] Kampis, George. 1993. Life-like computing beyond the machine metaphor. *In*: R. Paton [ed]: Computing with biological metaphors, London: Chapman and Hall. Contact: gk@cfnext.physchem.chemie.uni-tuebingen.de

[45] Kauffman, Stuart A. 1993. The origins of order, self-organization and selection in evolution. Oxford University Press. Pp. 709.

[46] Kerem, Bat-sheva, Johanna M. Rommens, Janet A. Buchanan, Danuta Markiewicz, Tara K. Cox, Aravinda Chakravarti, Manuel Buchwald, and Lap-Chee Tsui. 1989. Identification of the cystic fibrosis gene: genetic analysis. Science 245: 1073–1080.

[47] Koza, John R. 1992. Genetic programming, on the programming of computers by means of natural selection. Cambridge, MA: MIT Press.

[48] Langton, C. G. 1986. Studying artificial life with cellular automata. Physica 22D: 120–149.

[49] Levy, Steven. 1992. Artificial Life, the quest for a new creation. Pantheon Books, New York. Pp. 390.

[50] Levy, Steven. 1992. A-Life Nightmare. Whole Earth Review #76, Fall 1992, p. 22.

[51] Litherland, J. 1993. Open-ended evolution in a computerised ecosystem. A Masters of Science dissertation in the Department of Computer Science, Brunel University. Contact: david.martland@brunel.ac.uk

[52] Mac Arthur, Robert H., and Edward O. Wilson. 1967. The theory of island biogeography. Princeton University Press. Pp. 203.

[53] Maley, Carlo C. 1993. A model of early evolution in two dimensions. Masters of Science thesis, Zoology, New College, Oxford University. Contact: cmaley@oxford.ac.uk

[54] Manousek, Wolfgang. 1992. Spontane Komplexitaetsentstehung — TIERRA, ein Simulator fuer biologische Evolotion. Diplomarbeit, Universitaet Bonn, Germany, Oktober 1992. Contact: Kurt Stueber, stueber@vax.mpiz-koeln.mpg.d400.de

[55] Margulis, Lynn, and Dorion Sagan. 1986. Origin of sex. New Haven: Yale University Press.

[56] Marx, Jean L. 1989. The cystic fibrosis gene is found. Science 245: 923–925.

[57] Maynard Smith, J. 1971. What use is sex?, J. Theoret. Biol. 30: 319–335.

[58] Maynard Smith, J. 1992. Byte-sized evolution. Nature 355: 772–773.

[59] Maynard Smith, J, Christopher G. Dowson, and Brian G. Spratt. 1991. Localized sex in bacteria. Nature 349: 29–31.

[60] Mead, Carver. 1993. Analog VLSI and neural systems. Addison-Wesley Publishing Co. Pp. 371.

[61] Michod, Richard E., and Bruce R. Levin. 1988. The evolution of sex: an examination of current ideas. Sunderland, MA: Sinauer Associates.

[62] Moravec, Hans. 1988. Mind Children: the future of robot and human intelligence. Cambridge, MA: Harvard University Press.

[63] Moravec, Hans. 1989. Human culture: a genetic takeover underway. *In*: Langton, C. [ed], Artificial Life, Santa Fe Institute Studies in the Sciences of Complexity, vol. VI, 167–199. Redwood City, CA: Addison-Wesley.

[64] Moravec, Hans. 1993. Pigs in cyberspace. Extropy #10, Winter/Spring, 1993.

[65] Morris, S. Conway. 1989. Burgess shale faunas and the Cambrian explosion. Science 246: 339–346.

[66] Nowick, J., Feng, Q., Tijivikua, T., Ballester, P. & Rebek, J. 1991. Journal of the American Chemical Society 113: 8831–8839.

[67] Orgel, L. E., and F. H. C. Crick. 1980. Selfish DNA: the ultimate parasite. Nature 284: 604–607.

[68] Rasmussen, Steen, Carsten Knudsen, Rasmus Feldberg, and Morten Hindsholm. 1990. The coreworld: emergence and evolution of cooperative structures in a computational chemistry. Physica D 42: 111–134.

[69] Rasmussen, S., C. Knudsen, and R. Feldberg. 1991. Dynamics of programmable matter. *In*: Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen [eds], Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, vol. X, 211–254. Redwood City, CA: Addison-Wesley.

[70] Ray, T. S. 1979. Slow-motion world of plant 'behavior' visible in rainforest. Smithsonian 9(12): 121–30.

[71] ______. 1991. An approach to the synthesis of life. *In*: Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen [eds], Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, vol. X, 371–408. Redwood City, CA: Addison-Wesley.

[72] ______. 1991. Population dynamics of digital organisms. *In*: Langton, C. G. [ed.], Artificial Life II Video Proceedings. Redwood City, CA: Addison Wesley.

[73] ______. 1991. Is it alive, or is it GA? *In*: Belew, R. K., and L. B. Booker [eds.], Proceedings of the 1991 International Conference on Genetic Algorithms, 527–534. San Mateo, CA: Morgan Kaufmann.

[74] ______. 1991. Evolution and optimization of digital organisms. *In*: Billingsley K. R., E. Derohanes, H. Brown, III [eds.], Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers, Athens, GA, 30602: The Baldwin Press, The University of Georgia.

[75] ______. 1992. Foraging behaviour in tropical herbaceous climbers (Araceae). Journal of Ecology. 80: 189–203.

[76] ———. 1992. Tierra.doc. Documentation for the Tierra Simulator V4.0, 9–9–92. Newark, DE: Virtual Life. The full source code and documentation for the Tierra program is available by anonymous ftp at: tierra.slhs.udel.edu [128.175.41.34] and life.slhs.udel.edu [128.175.41.33], or by contacting the author.

[77] ———. In press. Evolution and complexity. *In*: Cowan, George A., David Pines and David Metzger [eds.], Complexity: Metaphor and Reality. Addison-Wesley Publishing Co.

[78] ———. Submitted. Evolution, complexity, entropy, and artificial reality. Physica D.

[79] Rommens, Johanna M., Michael C. Iannuzzi, Bat-sheva Kerem, Mitchell L. Drumm, Georg Melmer, Michhael Dean, Richard Rozmahel, Jeffery L. Cole, Dara Kennedy, Noriko Hidaka, Martha Zsiga, Manuel Buchwald, John R. Riordan, Lap-Chee Tsui, Francis S. Collins. 1989. Identification of the cystic fibrosis gene: chromosome walking and jumping. Science 245: 1059–1065.

[80] Riordan, John R., Johanna M. Rommens, Bat-sheva Kerem, Noa Alon, Richard Rozmahel, Zbyszko Grzelczak, Julian Zielenski, Si Lok, Natasa Plavsic, Jia-Ling Chou, Mitchell L. Drumm, Michael C. Lannuzzi, Francis S Collins, Lap-Chee Tsui. 1989. Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA. Science 245: 1066–1073.

[81] Skipper, Jakob. 1992. The computer zoo – evolution in a box. *In*: Francisco J. Varela and Paul Bourgine [eds.], Toward a practice of autonomous systems, proceedings of the first European conference on Artificial Life. MIT Press, Cambridge, MA. Pp. 355–364. Contact: Jakob.Skipper@copenhagen.ncr.com

[82] Sober, E. 1984. The nature of selection. MIT Press, Cambridge, MA.

[83] Spafford, Eugene H. 1989. The internet worm program: an analysis. Computer Communication Review 19(1): 17–57. Also issued as Purdue CS technical report TR-CSD-823. Contact: spaf@purdue.edu

[84] Spafford, Eugene H. 1989. The internet worm: crisis and aftermath. CACM 32(6): 678–687. Contact: spaf@purdue.edu

[85] Stanley, Steven M. 1973. An ecological theory for the sudden origin of multicellular life in the late precambrian, Proc. Nat. Acad. Sci. 70: 1486–1489.

[86] Stearns, Steven C. 1987. The evolution of sex and its consequences. Boston: Birkhäuser Verlag.

[87] Strickberger, Monroe W. 1985. Genetics. Macmillan Publishing Co. New York.

[88] Strong, D. R. and T. S. Ray. 1975. Host tree location behavior of a tropical vine (*Monstera gigantea*) by skototropism. Science, 190: 804–06.

[89] Surkan, Al. Unpublished. Self-balancing of dynamic population sectors that consume energy. Department of computer science, UNL. "Tierra-like systems are being explored for their potential applications in solving the problem of predicting the dynamics of consumption of a single energy carrying natural resource". Contact: surkan@cse.unl.edu

[90] Syvanen, Michael. 1984. The evolutionary implications of mobile genetic elements. Ann. Rev. Genet. 18: 271–293.

[91] Tackett, Walter, and Jean-Luc Gaudiot. 1993. Adaptation of self-replicating digital organisms. Proceedings of the International Joint Conference on Neural Networks, Nov. 1993, Beijing, China. IEEE Press. Contact: tackett@ipld01.hac.com, tackett@priam.usc.edu

[92] Taylor, Charles E., David R. Jefferson, Scott R. Turner, and Seth R. Goldman. 1989. RAM: artificial life for the exploration of complex biological systems. *In*: Langton, C. [ed], Artificial Life, Santa Fe Institute Studies in the Sciences of Complexity, vol. VI, 275–295. Redwood City, CA: Addison-Wesley.

[93] Thomas, C. A. 1971. The genetic organization of chromosomes. Ann. Rev. Genet. 5: 237–256.

[94] Todd, Peter M. 1993. Artificial death. Proceedings of the Second European Conference on Artificial Life (ECAL93), Vol. 2, Pp. 1048–1059. Brussels, Belgium: Universite Libre de Bruxelles. Contact: ptodd@spo.rowland.org

[95] Van Valen, L. 1973. A new evolutionary law. Evolutionary Theory 1: 1–30.

[96] Williams, George C. 1975. Sex and evolution. Princeton: Princeton University Press.