ARTIFICIAL LIFE CHRISTOPHER G. LANGTON

"Art" + "Life" = Artificial Life: Life made by Man rather than by Nature. Our technological capabilities have brought us to the point where we are on the verge of creating "living" artifacts. The field of Artificial Life is devoted to studying the scientific, technological, artistic, philosophical, and social implications of such an accomplishment.

1 THE BIOLOGY OF POSSIBLE LIFE

Biology is the scientific study of life — in principle anyway. In practice, biology is the scientific study of life on Earth based on carbon-chain chemistry. There is nothing in its charter that restricts biology to carbon-based life; it is simply that this is the only kind of life that has been available to study. Thus, theoretical biology has long faced the fundamental obstacle that it is impossible to derive general principles from single examples.

Without other examples, it is difficult to distinguish essential properties of life-properties that would be shared by any living system — from properties that may be incidental to life in principle, but which happen to be universal to life on Earth due solely to a combination of local historical accident and common genetic descent. In order to derive general theories about life, we need an ensemble of instances to generalize over. Since it is quite unlikely that alien life forms will present themselves to us for study in the near future, our only option is to try to create alternative life-forms ourselves — Artificial Life — literally "life made by Man rather than by Nature".

Artificial Life ("AL" or "A-Life") is the name given to a new discipline that studies "natural" life by attempting to recreate biological phenomena from scratch within computers and other "artificial" media. A-life complements the analytic approach of traditional biology with a synthetic approach: rather than studying biological phenomena by taking living organisms apart to see how they work, we attempt to put together systems that behave like living organisms.

The process of synthesis has been an extremely important tool in many disciplines. Synthetic chemistry — the ability to put together new chemical compounds not found in nature — has not only contributed enormously to our theoretical understanding of chemical phenomena, but has also allowed us to fabricate new materials and chemicals that are of great practical use for industry and technology. Artificial life amounts to the practice of "synthetic biology," and, by analogy with synthetic chemistry, the attempt to recreate biological phenomena in alternative media will result in not only better theoretical understanding of the phenomena under study, but also in practical applications of biological principles in industry and technology. By extending the horizons of empirical research in biology beyond the territory currently circumscribed by life-as-we-know-it, the study of Artificial Life gives us access to the domain of life-as-it-could-be, and it is within this vastly larger domain that we must ground general theories of biology and in which we will discover novel and practical applications of biology in our engineering endeavors.

1.1 AI AND THE BEHAVIOR GENERATION PROBLEM

Artificial Life is concerned with generating life-like behavior. Thus, it focuses on the problem of creating behavior generators. A good place to start is to identify the mechanisms by which behavior is generated and controlled in natural systems, and to recreate these mechanisms in artificial systems. This is the course we will take later in this paper.

The related field of Artificial Intelligence is concerned with generating intelligent behavior. It, too, focuses on the problem of creating behavior generators. However, although it initially looked to natural intelligence to identify its underlying mechanisms, these mechanisms were not known, nor are they today. Therefore, following an initial flirt with neural nets, Al became wedded to the only other known vehicle for the generation of complex behavior: the technology of serial computer programming. As a consequence, from the very beginning artificial intelligence embraced an underlying methodology for the generation of intelligent behavior that bore no demonstrable relationship to the method by which intelligence is generated in natural systems. In fact, AI has focused primarily on the production of intelligent solutions rather than on the production of intelligent behavior. There is a world of difference between these two possible foci.

By contrast, Artificial Life has the great good fortune that many of the mechanisms by which behavior arises in natural living systems are known. There are still many holes in our knowledge, but the general picture is in place. Therefore, Artificial Life can start by recapturing natural life, and has no need to resort to the sort of initial infidelity that is now coming back to haunt Al. Furthermore, Artificial Life is not primarily concerned with building systems that reach some sort of solution. For AL systems, the ongoing dynamic is the behavior of interest, not necessarily the state ultimately reached by that dynamic.

The key insight into the natural method of behavior generation is gained by noting that nature is fundamentally paralIel. This is reflected in the "architecture" of natural living organisms, which consist of many millions of parts, each one of which has its own behavioral repertoire. Living systems are highly distributed, and quite massively parallel. If our models are to be true to life, they must also be highly distributed and quite massively parallel. Indeed, it is unlikely that any other approach will prove viable.

2 HISTORICAL ROOTS OF ARTIFICIAL LIFE

Mankind has a long history of attempting to map the mechanics of his contemporary technology onto the workings of nature, trying to understand the latter in terms of the former. It is not surprising, therefore, that models of life have also reflected the principal technology of the era, from pneumatics in the time of the Greeks to computation in the current "information age".

2.1 EARLY HISTORY

The earliest models were simple statuettes and paintings - works of art which captured the static form of living things. Later, these statues were provided with articulated arms and legs in the attempt to capture the dynamic form of living things. These simple statues incorporated no internal dynamics, requiring human operators to make them behave.

The earliest mechanical devices that were capable of generating their own behavior were based on the technology of water transport. These were the early Egyptian water clocks called Clepsydra. These devices made use of a rate-limited process — in this case the dripping of water through a fixed orifice — to indicate the progression of another process — the position of the sun. Ctesibius of Alexandria developed a water powered mechanical clock around 135 BC which employed a great deal of the available hydraulic technology including floats, a siphon, and a water-wheel driven train of gears.

In the first century AD, Hero of Alexandria produced a treatise on Pneumatics, which described, among other things, various simple gadgets in the shape of animals and humans that utilized pneumatic principles to generate simple movements.

However, it was really not until the age of mechanical clocks that artifacts exhibiting complicated internal dynamics became possible. Around 850 AD, the mechanical escapement was invented, which could be used to regulate the power provided by falling weights. This invention ushered in the great age of clockwork technology. Throughout the Middle Ages and the Renaissance, the history of technology is largely bound up with the technology of clocks. Clocks often constituted the most complicated and advanced application of the technology of an era.

Perhaps the earliest clockwork simulations of life were the so-called "Jacks", mechanical "men" incorporated in early clocks who would swing a hammer to strike the hour on a bell. The word "jack" is derived from "jaccomarchiadus," which means "the man in the suit of armor." These accessory figures retained their popularity even after the spread of clock dials and hands — to the extent that clocks were eventually developed in which the function of time-keeping was secondary to the control of large numbers of figures engaged in various activities, to the point of acting out entire plays.

Finally, clockwork mechanisms appeared which had done away altogether with any pretense at time-keeping. These "automata" were entirely devoted to imparting life-like motion to a mechanical figure or animal. These mechanic automation simulations of life included such things as elephants, peacocks, singing birds, musicians, and even fortune tellers.

This line of development reached its peak in the famous duck of Vaucanson, described as "an artificial duck made of gilded copper who drinks, eats, quacks, splashes about on the water, and digests his food like a living duck". Vaucanson's goal is captured neatly in the following description.[1]

In 1735 Jacques de Vaucanson arrived in Paris at the age of 26. Under the influence of contemporary philosophic ideas, he had tried, it seems, to reproduce life artificially.

Unfortunately, neither the duck itself nor any technical descriptions or diagrams remain that would give the details of construction of this duck. The complexity of the mechanism is attested to by the fact that one single wing contained over 400 articulated pieces.

One of those called upon to repair Vaucanson's duck in later years was a "mechanician" named Reichsteiner, who was so impressed with it that he went on to build a duck of his own — also now lost — which was exhibited in 1847.

1

Figure 1 shows two views of one of the ducks — there is some controversy as to whether it is Vaucanson's or Reichsteiner's. The mechanism inside the duck would have been completely covered with feathers and the controlling mechanism in the box below would have been covered up as well.

2.2 THE DEVELOPMENT OF CONTROL MECHANISMS

Out of the technology of the clockwork regulation of automata came the more general — and perhaps ultimately more important — technology of process control. As attested to in the descriptions of the mechanical ducks, some of the clockwork mechanisms had to control remarkably complicated actions on the part of the automata, not only powering them but sequencing them as well.

Control mechanisms evolved from early, simple devices such as a lever attached to a wheel which converted circular motion into linear motion — to later, more complicated devices, such as whole sets of cams upon which would ride many interlinked mechanical arms, giving rise to extremely complicated automaton behaviors. 2

Eventually programmable controllers appeared, which incorporated such devices as interchangeable cams, or drums with movable pegs, with which one could program arbitrary sequences of actions on the part of the automaton. The writing and picture drawing automata of Figure 2, built by the Jaquet-Droz family, are examples of programmable automata. The introduction of such programmable controllers was one of the primary developments on the road to general purpose computers.

2.3 ABSTRACTION OF THE LOGICAL "FORM" OF MACHINES

During the early part of the 20 century, the formal application of logic to the mechanical process of arithmetic lead to the abstract formulation of a "procedure". The work of Church, Kleene, Gödel, Turing, and Post formalized the notion of a logical sequence of steps, leading to the realization that the essence of a mechanical process — the "thing" responsible for its dynamic behavior — is not a thing at all, but an abstract control structure, or "program" — a sequence of simple actions selected from a finite repertoire. Furthermore, it was recognized that the essential features of this control structure could be captured within an abstract set of rules — a formal specification — without regard to the material out of which the machine was constructed.

The "logical form" of a machine was separated from its material basis of construction, and it was found that "machineness" was a property of the former, not of the latter. Today, the formal equivalent of a "machine" is an algorithm: the logic underlying the dynamics of an automaton, regardless of the details of its material construction. We now have many formal methods for the specification and operation of abstract machines: such as programming languages, formal language theory, automata theory, recursive function theory, etc. All of these have been shown to be logically equivalent.

Once we have learned to think of machines in terms of their abstract, formal specifications, we can turn around and view abstract, formal specifications as potential machines. In mapping the machines of our common experience to formal specifications, we have by no means exhausted the space of possible specifications. Indeed, most of our individual machines map to a very small subset of the space of specifications — a subset largely characterized by methodical, boring, uninteresting dynamics. When placed together in aggregates, however, even the simplest machines can participate in extremely complicated dynamics.

2.4 GENERAL PURPOSE COMPUTERS

Various threads of technological development — programmable controllers, calculating engines, and the formal theory of machines — have come together in the general purpose, stored program computer. Programmable computers are extremely general behavior generators. They have no intrinsic behavior of their own. Without programs, they are like formless matter. They must be told how to behave. By submitting a program to a computer — that is: by giving it a formal specification for a machine — we are telling it to behave as if it were the machine specified by the program. The computer then "emulates" that more specific machine in the performance of the desired task. Its great power lies in its plasticity of behavior. If we can provide a step-by-step specification for a specific kind of behavior, the chameleon-like computer will exhibit that behavior. Computers should be viewed as second-order machines — given the formal specification of a first-order machine, they will "become" that machine. Thus, the space of possible machines is directly available for study, at the cost of a mere formal description: computers "realize" abstract machines.

2.5 FORMAL LIMITS OF MACHINE BEHAVIORS

Although computers — and by extension other machines — are capable of exhibiting a bewilderingly wide variety of behaviors, we must face two fundamental limitations on the kinds of behaviors that we can expect of computers.

The first limitation is one of computability in principle. There are certain behaviors that are "uncomputable" — behaviors for which no formal specification can be given for a machine that will exhibit that behavior. The classic example of this sort of limitation is Turing's famous Halting Problem: can we give a formal specification for a machine which, when provided with the description of any other machine together with its initial state, will — by inspection alone — determine whether or not that machine will reach its halt state? Turing proved that no such machine can be specified. In particular, Turing showed that the best that such a proposed machine could do would be to emulate the given machine to see whether or not it halted. If the emulated machine halted, fine. However, the emulated machine might run forever before halting, and therefore the emulating machine could not answer whether or not it would halt. Rice and others have extended this indecisive result to the determination by inspection alone — of any non-trivial property of the future behavior of an arbitrary machine.[2]

The second limitation is one of computability in practice. There are many behaviors for which we do not know how to specify a sequence of steps that will cause a computer to exhibit that behavior. We can automate what we can explain how to do, but there is much that we can not explain how to do. Thus, although a formal specification for a machine that will exhibit a certain behavior may be possible in principle, we have no formal procedure for producing that formal specification in practice, short of a trial and error search through the space of possible descriptions.

We need to separate the notion of a formal specification of a machine — that is: a specification of the logical structure of the machine — from the notion of a formal specification of a machine's behavior — that is: a specification of the sequence of transitions that the machine will undergo. We have formal systems for the former, but not for the latter. In general, we cannot derive behaviors from specifications nor can we derive specifications from behaviors.

The moral is: in order to determine the behavior of some machines, there is no recourse but to run them and see how they behave! This has consequences for the methods by which we (or nature) go about generating behavior generators themselves, which we will take up in the section on evolution.

2.6 JOHN VON NEUMANN: FROM MECHANICS TO LOGIC

With the development of the general purpose computer, various researchers turned their attention from the mechanics of life to the logic of life.

The first computational approach to the generation of life-like behavior was due to the brilliant Hungarian mathematician John von Neumann. In the words of his colleague Arthur W. Burks, Von Neumann was interested in the general question:[3]

"What kind of logical organization is sufficient for an automaton to reproduce itself? This question is not precise and admits to trivial versions as well as interesting ones. Von Neumann had the familiar natural phenomenon of self-reproduction in mind when he posed it, but he was not trying to simulate the self-reproduction of a natural system at the level of genetics and biochemistry. He wished to abstract from the natural self-reproduction problem its logical form.

This approach is the first to capture the essence of the Artificial Life approach. To understand the field of Artificial Life, one need only replace references to "self-reproduction" in the above with references to any other biological phenomenon.

In von Neumann's initial thought experiment (his "kinematic model"), a machine floats around on the surface of a pond, together with lots of machine parts. The machine is a universal constructor: given the description of any machine, it will locate the proper parts and construct that machine. If given a description of itself, it will construct itself. This is not quite self-reproduction, however, because the offspring machine will not have a description of itself and hence could not go on to construct another copy. So, von Neumann's machine also contains a description copier: once the offspring machine has been constructed, the "parent" machine constructs a copy of the description that it worked from and attaches it to the offspring machine. This constitutes genuine self-reproduction.

Von Neumann decided that this model did not properly distinguish the logical form of the process from the material of the process, and looked about for a completely formal system within which to model self-reproduction. Stan Ularn — one of von Neumann's colleagues at Los Alamos[4] — suggested an appropriate formalism, which has come to be known as a cellular automaton (CA).

In brief, a CA consists of a regular lattice of finite automata, which are the simplest formal models of machines. A finite automaton can be in only one of a finite number of states at any given time, and its transitions between states from one time step to the next are governed by a state-transition table: given a certain input and a certain internal state, the state-transition table specifies the state to be adopted by the finite automaton at the next time step. In a CA, the necessary input is derived from the states of the automata at neighboring lattice points. Thus, the state of an automaton at time t+1 is a function of the states of the automaton itself and its immediate neighbors at time t. All of the automata in the lattice obey the same transition table and every automaton changes state at the same instant, time step after time step. CA's are good examples of the kind of computational paradigm sought after by Artificial Life: bottom-up, parallel, local-determination of behavior.

Von Neumann was able to embed the equivalent of his kinematic model as an initial pattern of state assignments within a large CA-lattice using 29 states per cell. Although von Neumann's work on self-reproducing automata was left incomplete at the time of his death, Arthur Burks organized what had been done, filled in the remaining details, and published it. Together with a transcription of von Neumann's 1949 lectures at the University of Illinois entitled "Theory and Organization of Complicated Automata," in which he gives his views on various problems related to the study of complex systems in general.[5] Figure 3 shows a schematic diagram of von Neumann's self-reproducing machine.

3

Von Neumann's CA model was a constructive proof that an essential characteristic behavior of living things — self-reproduction — was achievable by machines. Furthermore, he determined that any such method must make use of the information contained in the description of the machine in two fundamentally different ways:

Interpreted, as instructions to be executed in the construction of the offspring.

Uninterpreted, as passive data to be duplicated to form the description given to the offspring.

Of course, when Watson and Crick unveiled the structure of DNA, they discovered that the information contained therein was used in precisely these two ways in the processes of transcription/translation and replication.

In describing his model, von Neumann pointed out that:[6]

"By axiomatizing automata in this manner, one has thrown half of the problem out the window, and it may be the more important half. One has resigned oneself not to explain how these parts are made up of real things, specifically, how these parts are made up of actual elementary particles, or even of higher chemical molecules."

Whether or not the more important half of the question has been disposed of depends on the questions we are asking. If we are concerned with explaining how the life that we know emerges from the known laws of physics and organic-chemistry, then indeed the interesting part has been tossed out. However, if we are concerned with the more general problem of explaining how life-like behaviors emerge out of low-level interactions within a population of logical primitives, we have retained the more interesting portion of the question.

2.7 LESSONS FROM HISTORY

As stated at the beginning of this section, throughout history we have repeatedly tried to map our contemporary technology onto nature in an attempt to understand natural phenomena. Sometimes this has been successful, but in the case of both life and intelligence, such mappings have not provided satisfactory explanations. There is a lesson here: although Artificial Life uses computers as its primary tool for the synthesis of biological phenomena, we must not mistake the tool for the object under study and attempt to characterize life as a "computation." If we are able to bring life to computers, this will not be because life "is" a computation, at least as we now understand the term "computation." Rather, it will be because computers have some fundamental properties that will allow them to be organized in such a way that they can become alive. It is quite likely that we will learn more about computation by studying life than we will learn about life by studying computation. This will be taken up in more detail in the following section.

It is also important to note that in the control programs of early automata we see the roots of what Mitchell Resnick has called the "centralized mindset": the attribution to natural phenomena of a central controller that is ultimately responsible for their behavior. It is a mindset that has dominated most of our scientific, philosophical, and even religious thought for the last several centuries. In contrast, Resnick refers to the kind of approach advocated by Artificial Life as the "distributed mindset". The difference is crucial, since most of nature chugs along in the absence of any central controllers. In order to understand most of nature, therefore, we must abandon the centralized mindset and come to an understanding of the dynamics of distributed systems qua distributed systems.

3 THE ROLE OF COMPUTERS IN STUDYING LIFE AND OTHER COMPLEX SYSTEMS

Artificial Intelligence and Artificial Life are each concerned with the application of computers to the study of complex, natural phenomena. Both are concerned with generating complex behavior. However, the manner in which each field employs the technology of computation in the pursuit of its respective goals is strikingly different.

Al has based its underlying methodology for generating intelligent behavior on the computational paradigm. That is, Al has adopted the centralized control architecture of serial, "von Neumann" style computation as a model of intelligence. AL, on the other hand, is attempting to develop a new computational paradigm based on the distributed processes that support living organisms. That is, AL uses insights from biology to explore the dynamics of interacting information structures. AL has not adopted the computational paradigm as its underlying methodology of behavior generation, nor does it attempt to "explain" life as a "computation".

One way to pursue the study of artificial life would be to attempt to create life in-vitro, using the same kinds of organic chemicals out of which we are constituted. Indeed, there are numerous exciting efforts in this direction. This would certainly teach us a lot about the possibilities for alternative life-forms within the carbon-chain chemistry domain that could have (but didn't) evolve here.

However, biomolecules are extremely small and difficult to work with, requiring rooms full of special equipment, replete with dozens of "post-docs" and graduate students willing to devote the larger part of their professional careers to the perfection of electrophoretic gel techniques. Besides, although the creation of life in-vitro would certainly be a scientific feat worthy of note — and probably even a Nobel prize — it would not, in the long run, tell us much more about the space of possible life than we already know.

Computers provide an alternative medium within which to attempt to synthesize life. Modern computer technology has resulted in machinery with tremendous potential for the creation of life in-silico.

Computers should be thought of as an important laboratory tool for the study of life, substituting for the array of incubators, culture dishes, microscopes, electrophoretic gels, pipettes, centrifuges and other assorted wet-lab paraphernalia, one simple-to-master piece of experimental equipment devoted exclusively to the incubation of information structures.

The advantage of working with information structures is that information has no intrinsic size. The computer is the tool for the manipulation of information, whether that manipulation is a consequence of our actions or a consequence of the actions of the information structures themselves. Computers themselves will not be alive, rather they will support informational universes within which dynamic populations of informational "molecules" engage in informational "biochemistry." This view of computers as workstations for performing scientific experiments within artificial universes is fairly new, but it is rapidly becoming accepted as a legitimate — even necessary — way of pursuing science. In the days before computers, scientists worked primarily with systems whose defining equations could be solved analytically, and ignored those whose defining equations could not be so solved. This was largely the case because, in the absence of analytic solutions, the equations would have to be integrated over and over again — essentially simulating the time-behavior of the system. Without computers to handle the mundane details of these calculations, such an undertaking was unthinkable except in the simplest cases.

However, with the advent of computers, the necessary mundane calculations can be relegated to these idiot-savants, and the realm of numerical simulation is opened up for exploration. "Exploration" is an appropriate term for the process, because the numerical simulation of systems allows one to "explore" the system's behavior under a wide range of parameter settings and initial conditions. The heuristic value of this kind of experimentation cannot be over-estimated. One often gains tremendous insight for the essential dynamics of a system by observing its behavior under a wide range of initial conditions.

Most importantly, however, computers are beginning to provide scientists with a new paradigm for modeling the world. When dealing with essentially unsolvable governing equations, the primary reason for producing a formal mathematical model — the hope of reaching an analytic solution by symbolic manipulation — is lost. Systems of ordinary and partial differential equations are not very well suited for implementation as computer algorithms. One might expect that other modeling technologies would be more appropriate when the goal is the synthesis, rather than the analysis, of behavior.[7]

This expectation is easily borne out. With the precipitous drop in the cost of raw computing power, computers are now available that are capable of simulating physical systems from first principles. This means that it has become possible, for example, to model turbulent flow in a fluid by simulating the motions of its constituent particles — not just approximating changes in concentrations of particles within regions of the fluid, but actually computing their motions exactly.[8]

There is an extremely important point here, one that involves the ultimate goals of the scientific enterprise as well as the issue of "centralized" versus "distributed" mindsets. The point is the following. There is a fundamental difference between being able to describe or predict phenomena on the one hand and "explaining" them on the other hand. One can use Navier-Stokes equations to describe or predict fluid flows in many cases, but fluids are not calculating Navier-Stokes equations! The descriptive and predictive power of the Navier-Stokes approach is useful, but the phenomena of fluid flow is actually generated by quite different mechanisms. Physics has largely been considered successful when it has been able to produce a predictive, abstract description of a physical phenomenon, one that generally ignores the low level mechanisms by which the phenomenon is actually generated. This is a benefit of adopting the centralized mindset, and it is very useful to be able to do this whenever possible.

However, for most natural phenomena, there is probably no simpler description than the generative mechanisms themselves. In these circumstances adopting the distributed mindset is necessary. Note that just because we cannot provide an abstract, high-level predictive description of a phenomenon does not mean that we have failed to provide a scientific description, a low-level description may be "as scientific" as we can get concerning the phenomenon. Even when a simpler, high level description is possible, it is important to keep in mind the difference between the description of a process and an understanding of the mechanisms by which it is generated.

What does all of this have to do with the study of life? The most surprising lesson we have learned from simulating complex physical systems on computers is that complex behavior need not have complex roots. Indeed, tremendously interesting and beguilingly complex behavior can emerge from collections of relatively simple components. This leads directly to the exciting possibility that much of the complex behavior exhibited by nature — especially the complex behavior that we call life — also has simple generators. Since it is very hard to work backwards from a complex behavior to its generator, but very simple to create generators and synthesize complex behavior, a promising approach to the study of complex natural systems is to undertake the general study of the kinds of behavior that can emerge from aggregates of simple components (Figure 4).

4

Another surprising lesson is that a distributed system can behave as if it had a central controller when in fact it does not. Computers, especially parallel computers, allow us to adopt Resnick's "distributed mindset" in our studies of Nature. Before computers, it was extremely difficult to come to an understanding of how complex behavior could emerge in the absence of a central controller, which is why the "centralized mindset" dominated the study of Nature for so long. With computers, however, the capacity to build distributed models provides us with the necessary tools for experimenting with the dynamics of distributed systems, and it is becoming apparent that the most interestingly complex behaviors in Nature owe their complexity in part to their distributed, noncentralized architecture.

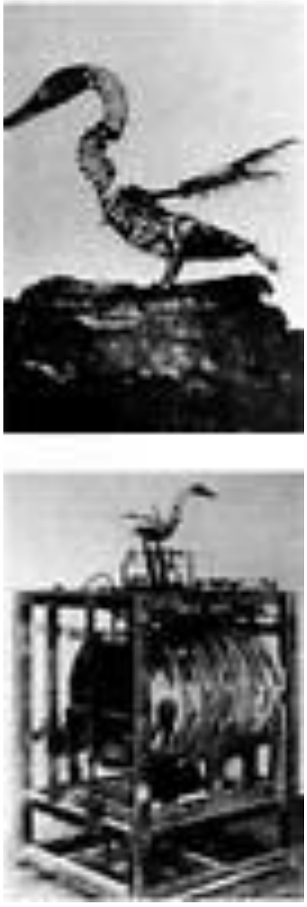


Fig. 1

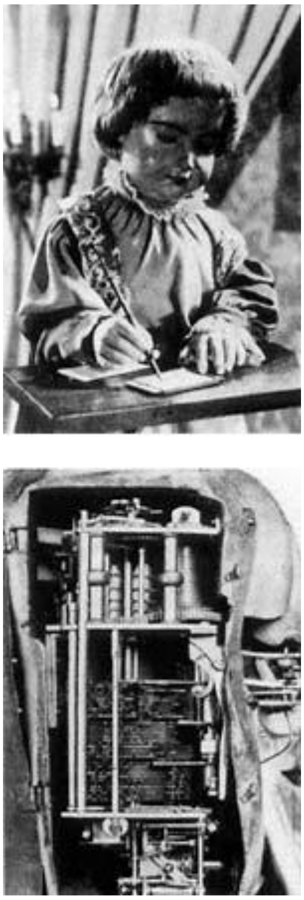
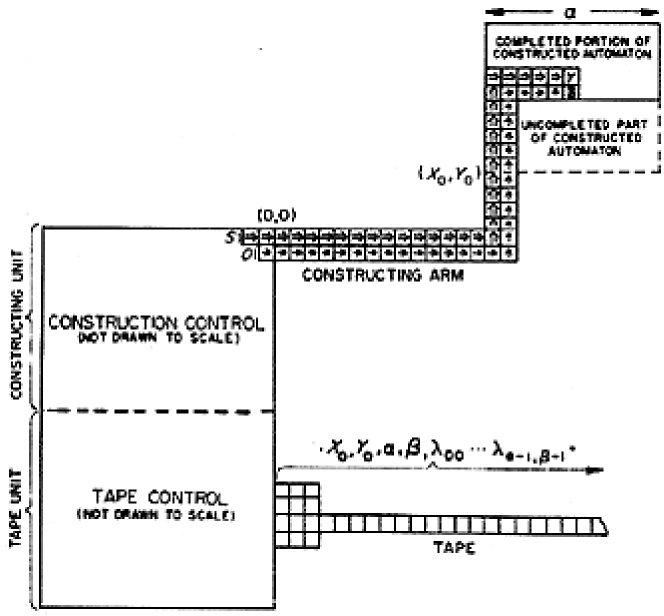
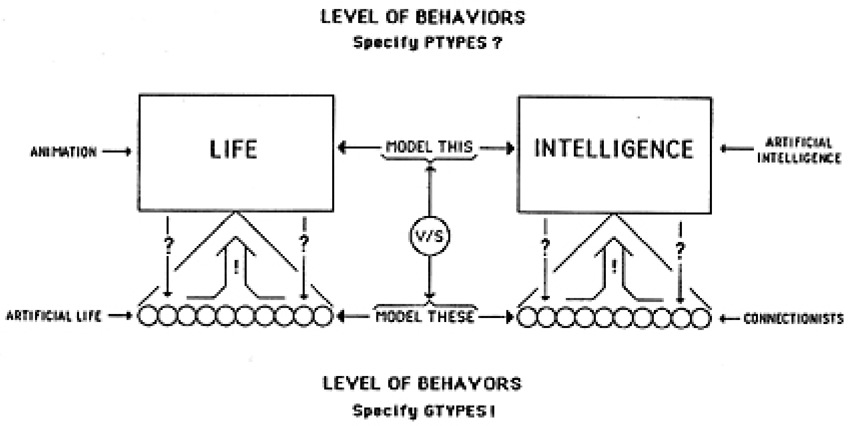


Fig.2


α
COMPLETED PORTION OF CONSTRUCTED AUTOMATON
β
UNCOMPLETED PART OF CONSTRUCTED AUTOMATON
γ
(X₀, Y₀)
(0,0)
CONSTRUCTING ARM
CONSTRUCTING UNIT
CONSTRUCTION CONTROL (NOT DRAWN TO SCALE)
TAPE UNIT
TAPE CONTROL (NOT DRAWN TO SCALE)
·X₀·Y₀·α·β·λ₀₀ ··· λ_{α-1,β-1}·
TAPE

3

fig. 3



4

fig. 4

http://personalwebs.coloradocollege.edu/~mwhitehead/courses/2013_2014/HCI/Papers/langton_al.pdf