Presentation on C. Langton paper’s entitled “Artificial Life” by Enrique Areyan

Thursday 8, September 2011

Key Concepts:

OVERVIEW

* Artificial life is the study of man-made systems that exhibit behaviors characteristic of natural living systems.
* Traditionally, biology attempts to *analyze* (top-down approach) living organisms while artificial life attempt to *synthesize* (bottom-up approach) life-like behaviors within computers and other artificial media.
* AL can contribute with biology by exploring not only *life-as-we-know-it* but *life-as-it-could-be*

THE BIOLOGY OF POSSIBLE LIFE

**Key Idea**: life-as-we-know-it vs. life-as-it-could-be

Traditional biology is the scientific study of life (*life-as-we-know-it),* based on carbon-chain chemistry, simply because this is the only kind of life that has been available for study.

Is it possible to derive general theories from single examples?

Life, as a dynamic physical process, could “haunt” other physical material. What matters is the organization of such material.

According lo Langton, life, as a process and regardless of its material bases, must share certain universal features. What are these universal features? Once we know what these are, we will be able to recognize life by its dynamic form alone, without reference to its matter.

We can try to synthesize alternative life-forms *(life-as-it-could-be)*, different from the life encountered in earth that have very specific properties, and study them in search of universal features of life.

* Artificial Life (**Key Idea**: synthetic vs. analytic approach to study life, and emergent behavior)
	+ A field that employs a synthetic approach to the study of *life-as-it-could-be*. Life as a property of the organization of the matter, rather than a property of the matter itself.
	+ Biology deals with the material basis (using an analytical, top-down approach) whereas AL deals with the formal basis (using a synthetic, bottom-up, distributed, local determination of behavior approach).
	+ Construct large aggregates of simple, rule-governed objects, which interact with one another in a nonlinearly fashion.
	+ **Emergent behavior**
* Artificiality (**Key Idea**: synthetic approach to study life)
	+ Connotes perceptual similarity but essential difference, resemblance from without rather than within. The artificial object imitates the real by turning the same face to the outer system.
	+ AL seeks to capture the behavioral essence of the components of a system – if organized correctly they should exhibit the same behavior as the natural system
	+ Force the basic components (behavioral primitives) of a system to obey basic rules of interaction among them, organized them as in the real system and let the behavior of interest **emerge**.
	+ The computer is the right tool for this computing-intensive approach, but one that follows an approach that is distributed and locally rather than globally governed. The idea is to focus on ongoing dynamic behavior.
	+ Illustrative idea: **colony of ants**.
* The animation of machines (**Key Idea**: rejection of vitalism)
	+ Rejecting vitalism, we do not need to “bring” life to a machine, we only need to organize its components in such a way that their interactive dynamics is “alive”
* The behavior generation problem (**Key Idea**: how do we go about creating behaviors generators? – consider that nature is fundamentally distributed and parallel)
	+ AL main focus is to create behaviors generators.
	+ Identify the mechanisms by which behavior is generated and controlled in natural systems, and recreate these in artificial systems.
	+ While AI has focused on the production of intelligent solutions, AL cares about the behavior of interest as expressed by its ongoing dynamics. Thus, it does not reach “a solution”.
	+ Nature’s fundamental architecture is both distributed and parallel, consisting on millions of parts, each one with its own behavior but constantly exchanging information.
	+ Our models should also be distributed and parallel to properly capture life

REVIEW OF DIFFERENT ASPECTS OF ARTIFICIAL LIFE

**Key Idea**: machines and the behaviors that they are capable of generating

Historical Roots of Artificial Life (**Key Idea**: discuss the roots of complex behavior)

The history of machines involves a continuing process of rendering in hardware progressively more complicated sequence of actions -physical and/or mental- previously carried out solely by recourse to muscle and brain.

The key invention was the mechanical clock, which allowed the construction of artifacts exhibiting complicated internal dynamics. Time keeping became a secondary function to control, which enable a large number of figures to engaged in different activities.

Example: duck of Vaucanson, an automaton that emulated the behavior of a natural duck. One single wing contained over 400 articulated pieces.

* The development of Control Mechanisms
	+ From the technology of clockwork regulation of automata emerged the more general technology of process control.
	+ Programmable controllers appeared. One could program arbitrary sequences of actions on the part of the automaton. This was one of the primary developments towards general-purpose computers.
	+ Example: Jaquet-Droz family
* Abstraction of the logical “form” of machines (**Key idea**: separation of logical form of a machine from its material)
	+ Early part of the 20th century. Logic + arithmetic = abstract formulation of a “procedure”
	+ Realization that the essence of a mechanical process, the “thing” responsible for its dynamic behavior, is not a thing but an abstract control structure or program.
	+ Separation of “logical form” of a machine from its material.
	+ Today, the formal equivalent of a machine is an algorithm: the logic underlying the dynamics of an automaton.
* General purpose computers
	+ Programmable controllers + calculating engines + formal theory of machines = general-purpose, stored program computer.
	+ A general computer has no intrinsic behavior of its own, it must be told how to behave through a program. A program is a formal specification for a machine thus, we are telling the computer to behave as it were the machine specified by the program.
	+ **Computers’ plasticity of behavior**
* Formal limits of machine behaviors
	+ Computability in principle. There are behaviors for which no formal specification can be given or no program can be written. Example: Turing’s halting problem: there is no formal specification for a machine which, when provided with the description of other machine together with its initial state, will –by inspection alone- determine whether or not that machine will reach its halt state.
	+ Computability in practice. There are behaviors for which we can give a formal specification in principle, but we have no formal procedure for producing that formal specification in practice.
	+ To determine the behavior of some machines, we have no alternative but run them and see how they behave.
* From mechanics to logic
	+ 1950’s and 1960’s. General purpose computer -> attention turned from mechanics of life to logic of life.
* The roots of complex behavior (**Key idea**: from the first attempts to capture life –paintings and statues- to clock-work automata the failure lies in centralized control. We must decentralized control to obtain life-like behavior)
	+ To sum up: since ancient times man have tried to build imitations of life. Early attempts captured the “form” (statuary and paintings) and then later attempts were made to “animate” such forms using hidden machinery.
	+ Throughout history we can see that the material out of which the constructing were made were irrelevant: what matter was the model’s dynamic behavior.
	+ In the time of the clock-work automata, the idea was to use a central “program” which was responsible for the dynamic behavior.
	+ In the view of Langton, this centralized approach constituted the failure of these models. This idea has been with us since that time up to the work in Artificial Intelligence.
	+ The most promising approach would have to dispend of any kind of global controller and focus on local, distributed behavior.

8BIOLOGICAL AUTOMATA

**Key Idea**: the field of AL searches for abstractions of the logical form of organisms from its material configuration.

Behavior generation in nature is primarily bottom-up, exceedingly parallel and distributed; unlike our serial and centralized inventions, i.e. computer.

* Genotypes and phenotypes (**Key Idea**: look at the basis of behavior generation from a biological point of view)
	+ These are the most important characteristics of living systems from the behavior generation point of view.
	+ Genotype is the set of genetic instructions encoded in the linear sequence of nucleotide bases that constitutes the DNA.
	+ Phenotype is the physical organism itself, the structure that emerge in space and time as the result of the interpretation of the genotype in the context of a particular environment (morphogenesis)
	+ Genotypes are to be thought as unordered sets of instructions, each one of which being the specification for a “machine” (program)
	+ Each instruction will interact with one another within the context of all other instructions but when its own triggering conditions are met and will have specific, local effects in the cell.
	+ The highly nonlinear genes’ interactions, coupled with its parallel and distributed nature of computing, will give birth to the phenotype, i.e. the structure and dynamics that emerge through time.
* Generalized genotypes (GTYPE) and phenotypes (PTYPE) (**Key Idea**: abstract the way nature generates behavior to approach other, non-biological situations)
	+ Metaphors to describe natural genotypes and phenotypes in non-biological situations
	+ GTYPE = largely unordered set of low-level rules
	+ PTYPE = behaviors and/or structures that emerge out of the interactions among GTYPE in some specific environment
	+ Think of GTYPE as an abstract specification for a set of “machines” and PTYES as the result of this interaction.
	+ Nowhere is the behavior as a whole specified. The global behavior of the aggregate is a consequence of many interactions.
	+ Local, nonlinear interaction produces PTYPE.
* Unpredictability of PTYPE from GTYPE (**Key Idea**: by simple inspecting a system’s initial condition and GTYPES one cannot predict the kind of PTYPE that will emerge. The key here is a myriad of components interacting in a nonlinear fashion)
	+ The set of all possible PTYPE is huge (factorial order). Therefore there is a vast richness of possible behaviors but we cannot predict PTYPES out of specific GTYPES, given initial conditions.
	+ To make PTYPE somewhat predictable, we must restrict severely the nonlinear dependence of PTYE on GTYPE. Notice trade-off between behavioral richness and predictability.
	+ A proper analogy is that of Turing machine where you cannot determine any nontrivial property of a computer from mere inspection of its program and initial state.
	+ Think of the machine’s transition table as the GTYPE and the resulting computing as the PTYE. Thus, we cannot predict PTYPE only by looking at the GTYPE. Just start the system and watch!
	+ We cannot know (through a formal procedure) which specific alterations must be made to a GTYPE to effect a desired change in PTYPE! The problem is that PTYE is the result of many nonlinear interactions among the system’s primitives.
	+ They may be a way to change a GTYPE to alter an specific portion of PTYPE but it is not feasible to compute (exhaustive search)
	+ Nature addresses this issue through a process of trial and error grounded on natural selection, arguably the *only* efficient procedure

RECURSIVELY GENERATED OBJECTS

**Key Idea**: from recursive rules (GTYPE) that apply to local structures *emerges* complex structures or behavior (PTYPE) of the global system. This is a much simpler approach to the generation of complex behavior than its alternative, top-down, analytical approach (counter example: expert system).

This is a general approach to construct GTYPE/PTYPE systems. The GTYPE is the recursive description of the object and PTYPE is the developing structure or the recursively-generated object or behavior obtained trough a process of morphogenesis. The system starts with a single part to which the rules are applied recursively over and over again.

* Example 1: Lindenmayer systems (**Key Idea**: present an example concerned with the growth and development of structural PTYPE) consist of a set of rules for rewriting strings of symbols. “X -> Y” means that one replaces every occurrence of symbol X in the structure with symbol Y. The rules can be applied recursively to newly rewritten structures *ad infinitum*.
	+ Simple Linear Growth: show example.
	+ Branching Growth: show example.
	+ Signal propagation (I will not mention it)
* Example 2: Cellular Automata (I will not mention it)
* Example 3: Computer Animation (**Key Idea:** present an example concerned with the development of a behavioral PTYPE)
	+ Based on Craig Reynolds implementation of a simulation of flocking behavior
	+ Large collections of autonomous but interacting objects (Boids) inhabiting a common simulated environment. The interaction is only at the local level thus; the global behavior of the aggregate of Boids is strictly an emergent phenomenon.
	+ Three simple rules (GTYPE) each Boid must obey:
		- Maintain a minimum distance from other objects in the environment, including other Boids.
		- Match velocities with Boids nearby.
		- Move toward the perceived center of mass of the Boids in its neighborhood.
	+ Out of this rules (GTYPES) emerges the behavior (PTYPE) of the flock of Boids. Notice that in these rules there is no reference to the global structure or its growth and development.
	+ The result is a very fluent and natural behavior, even in the presence of obstacles.
* Genuine Life in Artificial Systems (**Key Idea**: distinguish the ontological status of the various parts within the system)
	+ Clearly Boids are not birds. Moreover, Boids have no physical structure, but rather exist as information structures or “programs”
	+ However, at the behavioral level, both flocking Boids and flocking birds are two instances of the flocking phenomenon.
	+ **Key Concept**: the behavior of a flock does not depend on the composition of the entities inside the flock. The important thing is how each entity interacts with one another in the flock.
	+ Therefore, Langton states that flocking in Boids is true flocking, and may be used as empirical data to study flocking in general.
	+ The trait of being “artificial” in the field of AL refers to the component of the parts, not the behavior of the global system. Langton concludes that the behavior of these kinds of systems is just as genuine as their natural counterparts.
	+ Can this reasoning be extended to living systems? i.e., will an organized set of artificial primitives carrying out the same functional roles as biomolecules in natural living systems support a process that will be “alive”?

EVOLUTION

**Key Idea:** how do we go about finding GTYPES that will generate lifelike PTYPES?

So far we have use a trial and error process, modifying GTYES until the appropriate PTYPE is obtained. The problem is that we are guiding the search to what we think an appropriate PTYPE (life-like behavior) should be. But the search space is far bigger than our preconceptions allow us to see. Therefore, we should automate the process so that we do not overly constrain the search for appropriate GTPYES.

How do we do that? Nature has the proper mechanism, i.e., evolution by the process of natural selection among variants.

* Natural selection among populations of machines & Genetic Algorithms (**Key Idea**: use natural selection as an algorithm to search the GTYPE space)
	+ 1- Form populations of PTYPE by interpreting a set of GTYPES within a specific environment. Let the different PTYPES interact with one another and with the environment.
	+ 2- Evaluate the relative, application-specific performance (fitness function) of PTYPES and select (with certain probability) the GTYPES of the best performing PTYPES. The best the PTYPE, the higher the chances of selecting its associated GTYPE.
	+ 3- Out of these GTYPES select a pair and reproduce them (apply genetic operators, e.g. crossover operator) in such a way that the copies are similar but not identical to the originals.
	+ 4- Replace the least successful GTYPES with the offspring recently created. Repeat steps 1-4 *ad infinitum*.
	+ As imposed by the formal limitations on predictability, we have to let each GTYPE interact with the environment and others in order to explicitly evaluated its performance as expressed by its associated PTYPE.
	+ The GA explores a very large space of possible PTYES in an intelligent manner. In general, the GA selects the GTYPES building blocks most often associated with the most successful PTYPES thus, biasing the sample towards a probable better population of GTYPES.
* Emergent Fitness Functions (**Key Idea**: model both population, environment and fitness function as a bottom-up process)
	+ Problems with the binary fashion in which population is usually evaluated, either a GTYPE is “fit” or “unfit”. This creates a clear-cut boundary between environment and population, unusual to natural systems.
	+ While actors are specified bottom-up, the environment is often specified in a top-down fashion. This is a problem.
	+ This rigid, unlifelike environment produces rigid, unlifelike behavior. Therefore, the environment itself should be specified at the lowest possible level, in a bottom-up fashion.
	+ In the same manner, the fitness function should be an emergent property of the system.

THE ROLE OF COMPUTERS IN STUDYING LIFE AND OTHER COMPLEX SYSTEMS

Both AI and AL are concerned with generating complex behavior and employ the computer to study complex, natural phenomena. However, they do this very different:

* AI uses the technology of computation as a model of intelligence. In other words, it attempts to “explain” life as a kind of computer program.
* AL attempts to develop a new computational paradigm based on natural processes that support living organism. AL uses the computer as a tool to explore the dynamics of interacting information structures or programs.
	+ Computer should be thought of as an important laboratory that provides an alternative medium within which we can try to synthesize life (life as the structure or dynamic behavior that emerges from the interaction of local constituents)
	+ If we think of life just as a myriad of information structures interacting with one another, then the computer is the tool to try to synthesize it. The computer is the primary tool for the manipulation of information.
	+ Computer as a workstation for performing scientific experiments within artificial universes. Let the Computer take care of the mundane -but huge amount of- calculations.
	+ Complex behavior does not need to have complex roots. If the same is true about what we call life, then we can try the much simpler task of synthesizing complex behavior in the computer rather than creating it from a top-down approach.

NONLINEARITY AND LOCAL DETERMINATION OF BEHAVIOR

* Linear vs. nonlinear systems
	+ Linear system => the behavior of the whole is just the sum of the behavior of its parts.
		- Obey the superposition principle, i.e., break the system into simpler parts, analyze them independently and then compose them again. You would know everything you need to know about the system.
		- Analysis should be effective to understand this systems.
	+ Nonlinear systems => the behavior of the whole is much more than the sum of its parts.
		- Does not obey the superposition principle. The behavior of interest arise out or interactions between the parts not the part themselves. The behavior disappears if parts are studied independently.
		- Analysis won’t work; try the inverse of analysis, i.e. synthesis. Rather than start with the behavior of interest and try to analyze it down into its parts, start with its parts and put them together to attempt to synthesize the behavior of interest.
		- The fundamental parts of these kinds of systems can be behaviors themselves (virtual parts) which depend on nonlinear interactions between physical parts to exist.
		- AL is after these parts as the fundamental atoms and molecules of behavior.

CONCLUSION: THE EVOLUTION OF WATCHMAKERS

The process of evolution by natural selection (Blind Watchmaker) has created “seeing watches” capable of understanding what makes them “tick”, i.e. humans, which in turn may be able in the future to construct “watches” of their own design.

(Show picture of discovery of the structure of DNA and interpretation of the genetic code and a feedback loop stretching from molecules to men and back again or the process of evolution creating genotypes that result on phenotypes capable of manipulating their genotypes either copying them, altering them, or creating new ones)

Despite the technological thrill that this supposes, we need to address these questions:

* How can we justify our manipulations?
* How can we take it upon ourselves to create life, even within the artificial domain of computers, and then snuff it out again by halting the program or pulling the plug?
* What right to existence does a physical process acquire when it is a “living process”, regardless of the medium in which it takes place?
* Why should these rights accrue only to process with a particular material constitution and not another?
* AL is a technical as well as a social, moral, philosophical and religious challenge.

**References**

Langton, C. [1989]. “Artificial Life” In Artificial Life. C. Langton (Ed.). Addison-Wesley. pp. 1-47.