AGENT-BASED GENETIC AND EMERGENT COMPUTATIONAL MODELS OF COMPLEX SYSTEMS

Dooley K. and Corman S.
Arizona State University, USA

Keywords: complexity, simulation, self-organization, cellular automata, artificial life

Contents

Summary

Computational models based on agents and genetic learning have a tremendous potential for contributing to our understanding of how complex systems behave. Computational models, especially when embedded in a rigorous simulation experiment methodology, can give both theoretical and practical insight. In order for such models to continue to contribute, agents must be modeled as heterogeneous rather than homogeneous, and we must make advances in modeling the real complexity of schema, or mental models. Modelers should remember that the goal is not to mimic the system in detail, but rather capture its salient behavioral features. This is often difficult to accomplish because, unlike traditional statistical analysis, we lack formal methods for rejecting small, incremental changes in the models.

1. Introduction

Models are key to human understanding of the phenomena around them. Models help humans simplify the complexity of the world that they see, and facilitate prediction and thus experimentation and validation. Historically modeling has been a "top-down" activity: one would identify the key "state variables" that defined the system in question, and then one would formulate relationships (equations) that related these state variables to one another, and how they changed over time. For example, in an economics problem one might specify that GDP and unemployment rates are the key variables defining the economic system, and then formulate (analytically or empirically) a model that would relate the two variables to one another and how they change over time.

In a complex system, the ability to *a priori* identify state variables and their interrelationships is limited. It has been found that such systems are better modeled in a "bottoms-up" fashion: one starts by identifying the most basic building blocks of the system (agents), and then specifies how these agents interact with one another and the external environment. In this manner aggregate behavior is not prespecified, but rather emerges from within the system itself. Such a modeling approach is appropriate when the primary form of control in the system in question is self-organization--which is true in the case of many living systems. The computer has played an integral role in such modeling efforts, acting not only as an enabler but also a basic frame by which complex systems are specified.

In this topic-level article we shall describe the basic computational elements of a complex system, give examples of complex systems models formulated at different levels of detail, and discuss some of the extensions, limitations, and philosophical issues involved in such modeling.

2. General Computational Features of a Complex Adaptive System

The study of complex adaptive systems arose from the confluence of several disciplines: artificial intelligence, evolutionary biology, nonlinear dynamics, and living systems theory, *inter alia*. There is no doubt that the computer played a key role in both helping researchers understand complex systems, as well as acting as a frame for specifying models of how complex systems worked. In this sense, the computer acted not only as a research instrument, but also as an active participant in the development of complex systems theory.

John Holland's work ("ECHO") was the first to make explicit the minimal computational requirements for modeling complex systems. As stated in his book *Hidden Order*, part of the purpose of his formulation was to enable a computer language to be developed which would emulate the behavior of complex systems. As in any formulation of complex systems, Holland states the starting point of all complex systems are agents, and agents follow a stimulus response model such that rules predetermine that if stimulus "s" occurs then response "r" follows. These behavioral rules are subject to evolution and change via various genetic mechanisms. For instance, an agent in a financial market might behave according to the rule " if the stock price goes down more than 10 percent, sell." Of course that agent may learn that such a rule is over-reactive, and instead formulate a new rule that states, " if the stock price goes down two percent, sell; if the stock price goes down more than two percent, hold." In this sense the agent is adaptive. From a computational standpoint, rules can be embedded in lookup tables, expert systems-style if-then clauses, or numerical algorithms. Learning, or rule adaptation, is typically implemented via either genetic algorithms or schemes that change degree of belief parameters associated with rules.

Agents interact with one another to form a complex system that subsequently has emergent properties. For example, in an economy, each firm acts as an agent, and multiple firms make up the complex system known as the economy. One emergent aggregate property of the economy is the gross domestic product. In the complex system known as the nervous system, individual neurons interact in such a manner that behavior emerges.

Holland states that there are seven basic elements that constitute a complex system, and each of them in turn has computational implications. These seven basic elements are broken into two types, properties and mechanisms. Properties include aggregation, nonlinearity, flows, and diversity. Mechanisms include tagging, internal models, and building blocks. Let us consider the purpose of each of these elements and their computational implementation.

Aggregation is the property by which a group of agents can be collectively modeled as a meta-agent. Such a meta-agent has all the properties of a typical agent, and a collection of meta-agents forms a yet higher level complex system. For example, fish and other living matter in a lake can all be considered agents in the complex system labeled "lake". The lake itself can be considered an agent in the larger ecosystem. The lower level agents dictate the behavior of this meta-agent, known as "lake". Hence, complex systems are hierarchical. When modeling a complex system, these hierarchical levels can be taken into account explicitly by simply sequencing the activities in the simulation appropriately. For instance, in the case of the lake, one would first simulate the fish and plankton to determine the emergent properties of the lake, and then in turn simulate the lake and its surroundings to determine the emergent properties of the ecosystem.

Tagging is the mechanism that facilitates such aggregation. In simulating a business organization as a complex system individual people could be considered agents. The aggregate behavior of individuals would determine the emergent properties of the organization. However, individuals within an organization obviously have different roles. It may be more advantageous to aggregate individuals by their functional affiliation, thereby determining the aggregate behavior of a given work area. In this case, agents would be given a "tag" that associated them with a particular functional department: marketing, purchasing, sales, management, and so on. Tags not only facilitate aggregation, but also allow agents to differentiate behavior towards one another. For example an individual from marketing may behave according to one set of rules when encountering another person from marketing, but may behave according to a very different set of rules when encountering a person from management. In the computer, tagging an employee would be a matter of assigning a class(es) of attributes to the agent.

The third element is nonlinearity. In general a system is considered linear if its response is proportional to its stimulus; thus, nonlinearity implies a disproportionate response to the stimulus. When agent behavior is determined by a rule that is nonlinear, the subsequent interaction of agents will tend to be much more complex than a simple summing or averaging phenomena. For example, suppose we consider a collection of ants. One variable that might be tracked is how much food each ant has at any given time. The aggregate behavior, the total collection of food for the entire ant colony, is in fact an emergent property. In this case however the rule is linear: if an ant has two parcels of food, and adds three more, it now has five. Conversely, if one ant has five and another has six, the aggregate is 11. Now consider a nonlinear phenomenon. The ant lays down pheromones according to the following rule: if food is present and no pheromone exists, lay down much pheromone; if food is present and some pheromone exists, lay down much pheromone; if food is present and much pheromone exists, lay down little pheromone. Collectively, when aggregated together, the amount of pheromone present will not simply be related to how much food there is; it will be nonlinearly related to both how much food there is and how many ants have traversed the trail. Computationally, nonlinearity is implemented via rules that embed nonlinearity within them; such formulations may become as complex as a neural network.

The fourth property of a complex system is flow. In general, there are two things that flow between the elements of a complex system; physical elements, such as resources, and information. Two properties of flow are important to complex systems. One is the multiplier effect. For example, if investors all decide to put their financial resources into a single organization, then that organization can induce positive gains from having excessive capital. Its ability to gain from such investments may be quite a bit larger than simply the aggregate linear benefit gained for each individual investor. Such multiplier effects are the result of nonlinearity and flow. The second property is recycling. For example, when considering the effects of farming on the local ecosystem, one must determine whether the method of farming is in fact leaching the soil of nutrients or returning nutrients to the soil. Computationally, flows represent the simultaneous change of attributes of agents within the complex system. For example, in the financial system, investors have an attribute called "cash", and organizations have an attribute called "capital". When investors decide to back an organization, their cash is depleted while the organization's capital is replenished.

The fifth property of a complex system is diversity. Diversity is related to the fact that agents are idiosyncratic; each agent is potentially unique not only in the resources that they hold, but also in terms of the behavioral rules that define how they see the world and how they react. In a complex system diversity is the key towards survival. Without diversity, a complex system converges to a single mode of behavior, which can be perilous. As Holland states, "the diversity observed in a complex system is the product progressive adaptations... each new adaptation opens the possibility for further interactions and niches". The diversity of agents in the complex system leads to co-evolution. For example, parasites -- that is, individual organisms that live off the waste products of other organisms -- are in fact the most abundant form of life on earth. As an organism evolves so will its parasites; and as the parasite evolves, it supplies opportunities for the host to evolve to an improved state. Diversity is implemented within the computer via a diversity of rules that agents are both initiated to as well as allowed to evolve to, the diversity of tags and associated hierarchical aggregations that are included in the model, and the diversity of flows and agent attributes that constitute the system.

The sixth property of a complex system is internal models. These internal models are in fact behavioral rules, or mental models, or schema that define behavior. Internal models interpret the external world (both the complex system itself and its external environment) and also define the actions to be taken based on those observations. When dealing with agents that represent living objects, one must take into account imperfect observation; in human systems, these shortcomings are known as bounded rationality.

One of the criticisms of many complex systems models is that they fail to get anywhere near modeling the complexity of the actual internal models in action, and that they ignore the idiosyncrasy of individual agents by modeling the agents' internal models as homogenous throughout the system.

The final property of a complex system is building blocks. Building blocks are linked to internal models, in that most internal models are made up of existing building blocks. Building blocks are the reusable pieces, or modules, that economize the construction and reconstruction of internal models. For example, when individuals attempt to recognize a face that they observe, their minds do not refer to aggregate models of "face"; rather, building blocks relating to sub-elements of recognition (eyes, nose, hair) are pieced together into an aggregate "face". While it is expensive to develop new building blocks, it is relatively inexpensive to piece together existing blocks into new forms, i.e. new internal models. For example, individuals learn the alphabet not as an end in itself, but rather to use the letters as building blocks to form words. Even words themselves alone are meaningless; words act as building blocks towards the construction of purposeful sentences that can be interpreted with meaning. Computationally, it is typical that one must define *a priori* the building blocks that constitute the complex system being modeled. That is because one must know the structure of the building block, in computational terms, in order to develop the algorithms and logic that will simulate behavior and learning. Thus, selection of building blocks is an absolutely critical step in the formulation of the complex systems model.

One characteristic that is also an essential element of any complex system is a fitness function, which Holland discusses but for some reason does not consider elemental. Each agent has associated with it a fitness function that measures the amount of fitness, or health that the agent has at any given moment. The fitness function is typically composed of some linear and/or nonlinear combination of agent attributes. For example, the fitness of individual on a financial landscape may be a function of how much cash they have on hand, their reserves, and their immediate potential for additional income. In general, the fitness function of an agent is composed of both local and global attributes, i.e. having to do with both the fitness of the individual as well as the fitness of either the aggregate system that the agent is a member of, or the larger complex system itself. Fitness functions may be unique, or they may be shared across aggregates of agents or across the entirety of the system. To the extent that fitness criteria are shared, agents tend to act in a collective fashion. A fitness function makes the agent teleological, or goal driven. This does not necessarily mean that each agent behaves in a selfish manner. In fact, the fitness function may be such that the individual may take action that is negative for themselves but positive for the collective, because the fitness function may be weighted towards global concerns.

There are two ways in which fitness functions play a role in evolving internal models. First, adaptation can occur by assigning credit to particular rules that tend to increase fitness. Then the selection of a given rule will probabilistically depend on the amount of credit assigned to it. Or, rules may combine to form new rules via genetic mechanisms, mutate, live, and die according to the fitness of outcomes associated with them. Fitness functions are as key to behavior of the complex system as are building blocks: Change the fitness function, and behavior of the system will also change. Therefore, from a modeling perspective, choice of an appropriate fitness function is essential. Yet in reality determination of fitness functions may be in fact more difficult than the selection of building blocks.

For example, we have some understanding of how people piece together basic sensual stimuli in a cognitive fashion to determine appropriate interpretation and behavior; accordingly, what is the fitness function of a human being? Some have argued that the sole purpose of any living organism is to replicate its DNA. While this might be true, it perhaps does little for us in terms of practicality when trying to model how a consumer will react in a complex economic environment.

There are many ways in which the both immediate and long-term health of the individual may be impacted by their decisions. Even if one can identify the particular criteria that should be considered part of a fitness function, it is often difficult to specify the exact nature in which these criteria to be aggregated into a single fitness number. When criteria act somewhat independently, then a linear combination can be contemplated and one must determine the associated weights given to each attribute; when criteria act in interdependent way, than a multiplicative combination is in order. Real living systems probably do something much more complex than even this by using some sort of mechanism akin to a neural network.

Taken as a whole, these criteria do appear to fully specify a complex system in terms of its "standard", well-accepted (scientific vernacular) definition. As Stacey notes, complex systems as modeled above, follow both a reductionist as well as cognitive tradition. They differ from other modeling approaches, however, in that there is no attempt to model aggregate behavior at the level of itself; rather, aggregate behavior is the result of emergence--surprise from within.

In the next several sections we shall discuss computational models of complex systems. These models start off very simply, and become increasingly complex to the point where Holland's model is specified. Cellular automata have no fitness function *per se*; they have agents that share common internal models that do not evolve (typically), and have rigid rules of interaction (as operationalized by a spatial grid). Artificial life models relax rules of interaction, and may allow for unique rules for agents. As we move towards more specific models of human behavior we find explicit assumptions being made about agent fitness, interactions, attributes, and flow. We conclude with discussion of some philosophical issues pertaining to modeling and interpretation of such simulations.

3. What is Emergence in a Computational System?

As has been highlighted, one of the key features of any complex adaptive system is its emergent behavior. Behavior in a CAS is induced not by a single entity but rather by the simultaneous and parallel actions of agents within the system itself.

Thus, we refer to the system as self-organizing which undergoes “a process . . . whereby new emergent structures, patterns, and properties arise without being externally imposed on the system. Not controlled by a central, hierarchical command-and-control center, self-organization is usually distributed throughout the system".

In other words, the behavior of a CAS is emergent. Emergence is "the arising of new, unexpected structures, patterns, properties, or processes in a self-organizing system. These emergent phenomena can be understood as existing on a higher-level than the low-level components from which emergence took place. Emergent phenomena seem to have a life of their own with their own rules, laws and possibilities unlike the lower-level components."

Consider, for example, the phenomenon of bird flocking. At first glance, one may be tempted to believe that the complex order observed in the flocking pattern is the result of either a predetermined plan, or the result of unilateral control employed by the lead bird. In fact, flocking patterns emerge as part of the system's self-organizing behavior. Individual birds behave according to simple rules that are enacted based on local information. Any individual bird determines their speed and direction by flying towards the center of the flock, mimicking the velocity of birds around them and staying a safe distance away from neighboring birds. We also observe this same type of self-organizing behavior in human systems, where simple behavior based on local information can, in the aggregate, lead to complex global behavior.

Holland, in his book *Emergence*, attempts to develop an understanding of how emergence actually comes about. He states "recombination of elementary building blocks play a key role… (and that)…the component mechanisms act without central control, and the possibilities for emergence increase rapidly as the flexibility of the interaction increases… Emergence usually involves patterns of interaction that persist despite a continual turnover in the constituents of the patterns. A simple example is the standing wave in front of a rock in a white-water river. The water molecules making up the wave change instant by instant, but the wave persists so long as the rock is there and the water flows. Ant colonies, cities, and the human body (which turns over all of its constituent atoms in less than two years) offer more complex examples. These emergent macropatterns that depend on shifting micropatterns make emergence fascinating, and difficult to study."

-
-
-

Bibliography

Axelrod R. (1997). Advancing the art of simulation in the social sciences. Santa Fe Institute, paper no. 97-05-048. [Outlines the basic strategy and steps involved in computer simulation.]

Axtell R. (1999). The emergence of firms in a population of agents. Santa Fe Institute, paper no. 99-03-019. [A monumental study that introduces a new way to look at the theory of the firm.]

Bhargava S. C., Kumar A., and Mukherjee A. (1993). A stochastic cellular model of innovation diffusion. *Technological Forecasting and Social Change* **44**: 87-97. [Paper describes a logistic growth process of innovation adoption.]

Bonini C. P. (1963). *Simulation of Information and Decision Systems in the Firm*, Englewood Cliffs, NJ: Prentice-Hall. [A very complex model of organizational decision making, winner of the 1961-62 Ford Foundation dissertation award.]

Casti J. (1999). BizSim. *Complexity* **4**(4): 11-14. [Author summarizes some of the recent agent-based models used in business.]

Contractor N. S., and Grant S. J. (1996). The emergence of shared interpretations in organizations. In J. Watt & A. Van Lear (Eds.), *Dynamic Patterns in Communication Processes* (pp. 215-230). Thousand Oaks, CA: Sage. [Presents a network contagion model of organizational innovation adoption.]

Corman S. R., Stage C., and Scott C. R. (1997, May). *A computational organizational model of focused activity in a grocery store chain.* Paper presented at the annual meeting of the International Communication Association, Montreal, Quebec, Canada. [Develops a computational, agent-based model of organizational activity.]

Dooley K. (1997). A complex adaptive systems model of organizational change. *Nonlinear Dynamics, Psychology, and the Life Sciences* 1(1): 69-97. [Paper demonstrates the linkages between complexity and management theory.]

Drazen R. and Rao H. (1996). The adoption of competing technologies as an organizing process. *Technology Studies* 3(1): 1-29. [Presents a bandwagon model of organizational innovation adoption.]

Epstein J. and Axtell R.(1996). *Growing Artificial Societies*, Cambridge: MIT Press. [Detailed explanation of Sugarscape, an entire society built from the bottom-up.]

Holland J.H. (1995). *Hidden Order*, Reading, MA: Addison-Wesley. [Discusses the basic principles of complex adaptive systems.]

Holland J.H. (1998). *Emergence*, Reading, MA.: Addison-Wesley. [Discusses the roots of emergent behavior.]

Kelly K. (1994). *Out of Control*, Reading, Mass.: Addison-Wesley. [Extensive examples of complex systems at work.]

Langton C. (ed.) (1989). *Artificial Life*, Reading, MA.: Addison-Wesley. [Collection of papers investigating the new phenomena of artificial life.]

Resnick M. (1997). *Turtles, Termites, and Traffic Jams*, Cambridge: MIT Press. [Discusses basic steps in building and simulating agent-based models.]

Walker C. and. Dooley K. (1999). The stability of self-organized rule following work teams. *Computational and Mathematical Organization Theory.* [Shows example of theory building via a computational model.]

Wolfram S. (1994). *Cellular Automata and Complexity*, Reading, MA: Addison-Wesley. [Everything you'd want to know about CA.]

Biographical Sketches

Kevin J. Dooley, Ph.D. is a Professor of Management at the School of Business, Arizona State University (ASU) in Tempe, Arizona. Following undergraduate and graduate work in industrial engineering, he received his PhD in Mechanical Engineering from the University of Illinois. He holds concurrent appointments at ASU, at the School of Health Administration and Policy, and is a member of the core faculty in the Supply Chain Management PhD Program. His research interests are in quality, chaos and complexity, new product development, knowledge management and organization, and manufacturing systems. He is a co-author of the book, *Organizational Change and Innovation Processes: Theory and Methods for Research*, published by Oxford in 2000. He holds numerous awards, and is the current (2002) President of the Society for Chaos Theory in Psychology and Life Sciences.

Steven R. Corman, PhD, is Associate Professor in the Hugh Downs School of Human Communication at Arizona State University. He serves on the editorial board of *Human Communication Research*. His publications on communication networks, interaction processes, and computer models of communication have appeared in a number of professional journals. He is an editor of *Perspectives on Organizational Communication Finding Common Ground*, published by Guilford publications in 2000.