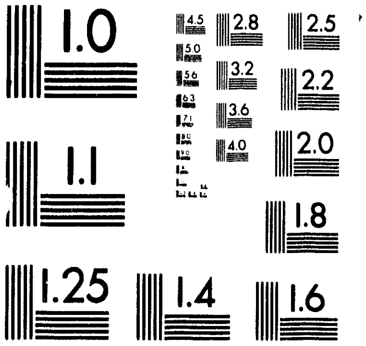
1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

1 of 1

On Modeling and Controlling Intelligent Systems

W. B. Dress
Instrumentation and Controls Division
Oak Ridge National Laboratory

an invited paper for the

Second International Conference on Information and Knowledge Management
November 1 - 5, 1993
Washington D. C.

Session: Tools for Realizing Intelligent Systems

Chairman: P. A. D. de Maine, Auburn University

V2.0 of 8/16/93

On Model:ng and Controlling Intelligent Systems*

W. B. Dress
Instrumentation and Controls Division
Oak Ridge National Laboratory

1. Abstract

The aim of this paper is to show how certain diverse and advanced techniques of information processing and system theory might be integrated into a model of an intelligent, complex entity capable of materially enhancing an advanced information management system. To this end, we first examine the notion of intelligence and ask whether a semblance thereof can arise in a system consisting of ensembles of finite-state automata. Our goal is to find a functional model of intelligence in an information-management setting that can be used as a tool. The purpose of this tool is to allow us to create systems of increasing complexity and utility, eventually reaching the goal of an intelligent information management system that provides and anticipates needed data and information.

We base our attempt on the ideas of general system theory where the four topics of system identification, modeling, optimization, and control provide the theoretical framework for constructing a complex system that will be capable of interacting with complex systems in the real world. These four key topics are discussed within the purview of cellular automata, neural networks, and evolutionary programming. This is a report of ongoing work, and not yet a success story of a synthetic intelligent system.

2. Introduction

However valuable for computational psychology, we are not attempting to create a system that merely mimics intelligence—no matter how clever, but rather one that can become the basis for a tool set to construct an intelligence, even though initially limited in scope and ability. To this end, we need to examine some attributes of intelligence that are desirable in a construct built for utilitarian purposes of information management.

In the introduction, we discuss reasons for placing identification, optimization, and modeling at the fore of this evolving methodology. To provide a focus for our investigation, we ask the obvious question: "What is an intelligent system?" Although no concise answer is forthcoming, tentative answers can indicate a direction for the ensuing research.

The remaining sections address the issues of identification, optimization, modeling, and control with examples of ongoing work at Oak Ridge National Laboratory (ORNL) that have practical impact for intelligent system construction. Much of this work is on the level of signals, data, and information; but it provides experience in carrying out the needed low-level modules, the opportunity to observe interactions between the modules, and can provide a touchstone to the reality of intelligent systems.

2.1. What is an Intelligent System?

At this point, it is not clear that intelligence can be adequately defined, although there are many candidate definitions ranging from the thoughtful and involved to the glib and terse. The subject seems to be as slippery as a definition of life from a linguistic point of view. For our purposes, it may not be necessary to have a definition as everyone will know an intelligent system when they see one (or will they?). As an adjective applied to toasters, software, and other gadgets of a useful but simplistic nature, the word is overworked to the point of having lost all meaning. Accordingly, an attempt will be made here to restore some value to the currency of "intelligent."

A property of intelligence that is by no means complete but serves as a point of departure for further thought is "an (artificially) intelligent system is one that has the capacity to surprise its maker." Some further consideration leads us to suggest the following minimum set of properties that an intelligent system should possess. This list

*Research performed at Oak Ridge National Laboratory, operated by Martin Marietta Energy Systems, Inc., for the U. S. Department of Energy under Contract No. DE-AC05-84OR21400.

is merely a starting point: necessary, but certainly not sufficient:

- ability to discriminate between self and environment,
- ability to improvise,
- ability to surprise,
- goal-possessing (establish, plan, and reach goals), and
- self-awareness (if only at the stimulus-response level).

An intelligent system cannot be created *in vacuo*. Such creation is at best a painstaking process replete with advances and retreats, successes and failures. Theory must advance beyond current practice to break new ground, and practice must catch up to and surpass (in capacity, at least) the theory to inspire the next intellectual advance. It has been suggested that hardware will (somehow) always "catch up" and so not need be considered. This narrow view overlooks the psychological need for people to see some hint of the actualization of their ideas. Current hardware must be capable of executing the best available code at a speed that does not discourage further development. On the other hand, potential future advances in intelligent systems must be foremost in the minds of those engaged in implementing their ideas on advanced hardware.

An intelligent information or knowledge system is, at the very least, a complicated system of interacting hardware and software modules. The more "intelligent" such a system is the closer this complication will approach complexity. Here, we distinguish between complicated systems (ones that have many interacting parts, requiring perhaps pages of diagrams for representation) and complex systems that incorporate some notion of self, however primitive, and goals that are inherent to the system. A complicated system is in some sense equivalent to the sum of its parts (example: a modern telephone exchange), while a complex system [Hubermann, 1988] is more than just the sum of its parts (example: any living organism). How can we identify complexity, if we can't define it?

Chaotic behavior is an objective indicator of a complex system, although that it is not sufficient may be seen from the many simple mathematical and physical systems that exhibit deterministic chaos. Indeed, there is a compelling intellectual movement [Langton, 1992] based on the linguistic label "complexity." Physicists have applied this label metaphorically to certain nonlinear behaviors of simple physical systems. Biologists have also used this term in their descriptions of organisms and associated behaviors. A case can be made that a unified mathematical description underlies both uses of the word—this provides the basis for a unifying synthesis in science [Anderson, 1988]. However, there is a danger in ignoring the very real differences between simple physical systems and complex biological ones by loading the term "complexity" with more that it can handle.

With these caveats in mind, we borrow a central idea from the Complexity School: ensembles of finite-state machines can self-organize to produce complex behavior [Wolfram, 1988]. The goal is the emergence of some properties of intelligence. We do this without proof or other substantiation than a hope born of necessity that our present tools will be sufficient.

2.2. The Modeling Relationship

The minimum set of properties listed above is best realized if certain identifiable and well-understood techniques are used in the construction of the intelligent system. Before we can discuss such details, however, we require a methodology for expressing those details. Foremost among these methods is the concept of a modeling relationship between a natural system and a formal system [Rosen, 1985]. We take a particular instance of the modeling relation by including a second formal model, the computer simulation. Figure 1 shows schematically the relationships between the systems being modeled and indicates the role of the scientific method in the encoding-decoding processes, where formal hypotheses about a natural system are encoded from observations and measurements. The ensuing predictions within the formal system are generated by the rules of inference (or the computer program) and verified or falsified subsequent to decoding.

This formalization of the modeling process serves as a framework for the simulation as well as a conceptual aid for distinguishing the model (whether mathematical or computer code) as separate from the process being modeled. At a later stage, we hope that the simulation itself becomes the intelligent system; but, given our current understanding of intelligence, we need to keep the process of simulation separate from the idea and goal o' intelligence.

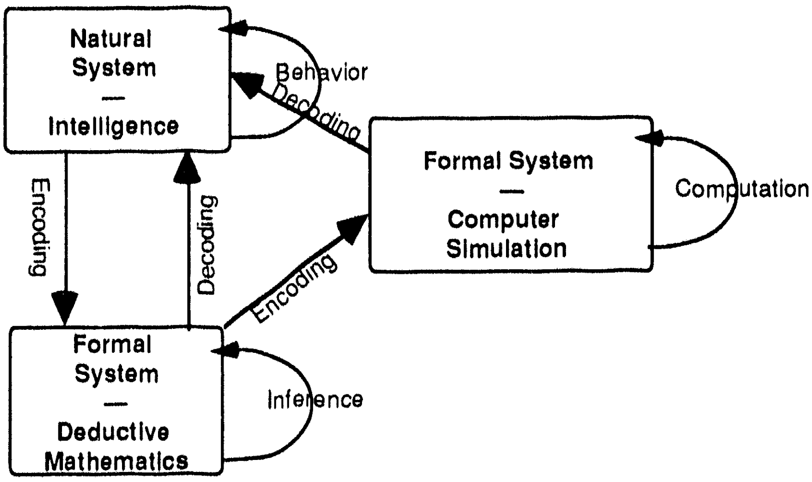
Natural System — Intelligence
Behavior
Decoding
Formal System — Computer Simulation
Computation
Encoding
Decoding
Encoding
Formal System — Deductive Mathematics
Inference

Figure 1. Modeling relationships between the natural, formal, and simulation systems. The encoding process establishes the behavior of the natural system to be captured in formal models, the decoding process allows verification of formal predictions.

The simulation as a model of the intelligent system comprises a large number of functions and processes expressed as modules. The purpose of each module is transduction of a data stream that subsequent modules can accept as meaningful. The entire simulation requires coordination and this coordination can be brought about by applying standard control technology, aided by estimation, identification, and optimization. We now turn to a discussion of some of these enabling techniques.

2.2.1. Episodic Processing

Episodes in sensory input—not merely stationary data streams—lead us to consider the use of wavelet transforms for ensuring channel independence and dimensionality reduction. Examples of such episodes may be user query of a database system, access attempts to a secure system, or receipt of a requested data packet by an information management system. Episodes occur in the continuous realm as well as in the discrete—voice access and intrusion detection are examples. Timely processing of episodes enables our system to respond intelligently to events.

One obvious constraint demanded by episodic processing is the requirement for real-time techniques so that the responses may be interrupt driven. The dimensionality reduction provided by wavelet signal processing can help ensure real-time capability. Some of the successes we have had using wavelet transforms for processing acoustic episodes (voice and aircraft) are mentioned below. Both discrete episodes (events) and symbolically coded episodes, where wavelets are not appropriate, lead to consideration of clustering and pattern recognition approaches.

2.2.2. Clustering and Classification

Clustering is the first step in concept formation and, as such, is an essential capability of any proto-intelligent system. The clusters, based on some measure of similarity, form the basis of class discrimination and structure. Formally, the similarity measure arises in defining a norm on a suitable vector space. The familiar Euclidean norm is only one of many, and may be inappropriate for most of the event clustering required by intelligent perception. Furthermore, the formal idea of a Cauchy sequence, leading to the notion of a Banach space [Debnath and Mikusinski, 1990] and thence to that of a Hilbert space (so useful in signal processing theory) is perhaps too restrictive for the kinds of sets that comprise the clusters for our incipient intelligence.

A particularly effective method of inducing clusters on a data set is provided by the finite-state machine (discussed below). Here, the selection of the set of internal states and the mapping of the input set to the state set is crucial. Most of the clustering algorithms developed over the last five decades can be recast as such automata and can thus provide the first stage of perceptual processing.

2.2.3. Pattern Recognition

Once the proto-concepts are established as clusters, the process of pattern recognition provides the next stage in perceptual processing.

That pattern recognition is a mature discipline with a large body of techniques ranging from the formal ideas of a primal sketch developed by Marr [1982] to the latest applications of neural networks, greatly eases our task. However, the many methods to choose from may give rise to confusion or incompatibilities in the system we wish to construct. There are two methodologies for overcoming this problem: look to biology for inspiration (e.g., insect vision) or allow the many methods to compete in an evolutionary approach to computer programming.

2.3. Automata Theory and Finite State Machines

As long as we are restricted to the mechanisms of modern digital computers for realizing our systems, we need to examine the concepts provided by automata theory. Finite automata or finite-state machines provide a powerful theoretical framework for discussing those intelligent information systems that we can potentially construct with present-day hardware.

Formally, a finite-state machine (FSM) [Kalman, Falb, and Arbib, 1969] is a quintuplet {I, O, S, ß, ø} where I is the set of input symbols, O the set of output symbols, and S the set of internal states. We restrict the set S to be finite, giving the machine its name. ß is a mapping from I x S —>S determining the next state of the machine, and ø is a mapping from I x S —>O determining the output of the machine. It may seem that the sets I and O, are not restricted as the inputs may originate in the external world and the outputs may effect actions or events in the external world. However, it is easy to see that the mappings ß and ø are severely restricted by the set S to operate on equivalence classes of possibly continuous input and output symbol spaces. Consider the ordered pair {i, s} that belongs to the domain of ß, and s to its finite range. A simple assumption for ß (namely that it is surjective) implies that I is really the set of equivalence classes of possibly continuous inputs partitioned by the mapping ß and the structure of the finite set S . A similar argument shows that the mapping ø likewise induces an equivalence relation on the set O. Thus the finiteness of the internal state set implies, from the point of view of the FSM itself, that the input and output spaces are likewise finite. A different FSM would induce a different partition of the externally continuous input and output states and consequently exhibit a different behavior: the structure of the machine determines its perception of the world.

This theorem has far-reaching consequences for our models and the complexity of the systems that can be modeled by that ubiquitous FSM, the digital computer. It also injects a note of caution: the partitions of the input space may be too coarse to allow adequate modeling of intelligence. The theorem is the foundation upon which the discriminatory capability of our intelligent system is to be built. If two crucially distinct states of the environment are placed in the same equivalence class by the relation {S, ß}, the FSM and our intelligent system will be totally unaware of the distinction. Selection of the correct internal states becomes crucial for the FSMs we need to employ: an incorrect set of input states entails a functionally inadequate FSM.

3. Modeling The Intelligent System

Some recent models of intelligent systems rely on nearly exact correspondences with what is currently known about the mammalian and human brain structure and function, e.g., [Albus, 1991]. Our view here is not so grandiose: we would be satisfied with an architecture that behaves intelligently no matter what its metaphorical correspondence with the human brain. As a partial justification, consider the process of a putative futuristic evolution of a machine intelligence. Given the crude raw materials that exist today (sensors, CPUs, RAM, ROM, tranducers, etc.), there is no compelling reason that an intelligence evolving from such an assortment would have either an architecture or function resembling that of the human nervous system. Quite the contrary, this intelligent machine would perceive its world through sensors specific to its teleologically defined tasks, and its structure would appear quite alien to a human neurobiologist.

Otherwise stated, we do not wish to impose form before function, but rather let form follow secondarily from the desired function. This means that behavior is paramount and the means of obtaining that behavior is of concern only much later in the design process. Thus, when efficiency, performance, size, weight, mobility, cost, and so on, become important, the particular issues of implementation, architectural details, and execution hardware will be addressed. For now, it is enough that complexity can arise from ensembles of relatively simple modules (the FSMs).

3.1. Cellular Automata

Cellular automata (CA) are special cases of the FSM defined above, and can be made to simulate a large number of physical and informatic processes [Dress, 1993]. As noted in that reference, the CA is a computational system that operates on locally available information, but can respond to global circumstances by changing its rule set or its neighborhood function. The latter describes which machines communicate with which. As an example, a neural network, as implemented in silicon or simulated on a serial computer, is a special case of a CA, having a particular neighborhood function and (as traditionally implemented) a very simple state set, a trivial input mapping (usually raw data sets), and a simple output mapping (e.g., the sigmoid function).

3.2. Neural Networks

Neural networks comprise a restricted class of FSM ensembles with an extension of the properties of the CA as mentioned above. A directed neighborhood function and a single state, namely the sum over all the inputs to a cell, define the most commonly used network architecture. As shown by Poggio and Girosi [1990], a neural network can be thought of as a means to synthesize an approximation to a multi-dimensional mapping. This synthesis is usually obtained from a limited set of sample pairs drawn from the range and domain of the function being approximated. In this light, neural network technology merely provides an alternate set of methods and its own specialized vocabulary to the field of statistical and functional approximation.

The unknown mapping is parameterized in a special way as a set of linear combinations of the "input vector" components that are sampled from the domain of the unknown function. The corresponding sampled value of the range of the function serves as the domain of a cost function (usually sum of the squares of the differences between the range samples and the current approximation determined by the parameters). The quality of the approximation to the (unknown) mapping depends on the "training set" used and the particular optimization techniques applied (most commonly a form of gradient descent in the parameter or "weight" space).

At their most useful, neural networks can serve as approximate representations of mappings too cumbersome to represent exactly, thereby allowing efficient pattern recognition or effective control of some of the complex interactions among modules of our intelligent simulation. At their worst, they often over fit the approximated function thereby producing useless generalizations of new domain vectors (patterns), or merely converge during their "training" or optimization stage to sub-optimal solutions—they approximate the wrong function.

3.3. Genetic Programming

In several instances throughout this discussion on intelligent systems, we defer the choice of a particular technique (e.g., the choice of optimization method) because not enough is known about a method's behavior under conditions that cannot be adequately prespecified. An outgrowth of genetic algorithms, known as genetic programming, can help us make those choices as well as assemble the entire intelligent system into a functional and efficient whole. Genetic programming (GP) [Koza, 1992] is a direct descendent of the genetic algorithm, but without the bit-string representation. Instead, programs or procedures are represented as parse trees, and the genetic operators are defined as tree operations.

Unfortunately, GP suffers from the same drawbacks as the genetic algorithm: slow convergence to a generally sub-optimal solution. However, it is one of the best and certainly the most thoroughly explicated of methods for obtaining reasonably effective computer programs from a high-level behavioral specification. It certainly has a role to play in constructing intelligent information systems and may undergo a needed evolutionary process itself if enough researchers start to use it on a wide set of problems.

4. Optimizing The Intelligent System

Since intelligence cannot exist without perception (and, of course, an environment to perceive), a proper starting point for intelligent-systems design is to identify the environment and those things in the environment that are important to the designer and potential user of the system. Taking a clue from our own sensory perception, we note two crucial facts: (1) the environment is continuously bombarding our senses with relentless data streams, and (2) our perceptual system is somehow able to filter this > 200 MHz data bandwidth (visual, auditory, tactile, proprioceptive) to obtain useful information that the brain can process. (We do not yet consider the possibility of internal data streams that an intelligence is able to generate for its own purposes.)

We should make provision for a system that can continuously receive data streams from its environment, transduce these data where appropriate, and filter the ensuing bit streams for the purpose of converting "data" into "information." These data streams typically contain queries from a local user or requests from remote sources; the responses are either stored as local and distributed data or obtained from the physical environment via appropriate sensors. The conversion process involves a dimensionality reduction by means of appropriate digital transforms and perhaps pattern recognition and classification. This process leads to the identification step that is essential in constructing any model.

The process of optimization is not only essential for producing an efficient system, but is needed to find a correct one as well. The more complex a system is, the more crucial the process of optimization to system performance. A system of sufficient complexity is nonfunctional over most of the "genetic" landscape available to its descriptive parameters. Optimization of some kind is required, not only to locate those sparse regions that are viable, but to optimize performance on those few viable regions.

Optimization depends on correct identification and estimation of system parameters. These disciplines are rooted in the theory and techniques of signal processing.

4.1. Signal Processing

At ORNL, we have had good success in processing incoming signals with the wavelet transform (to obtain information about the occurrence of an event as well as its relevant scale), and the Fourier transform (when only the frequency information is required), [Kercel, 1992a, b]. These two transforms, when used in a complementary fashion, provide a set of descriptors for one- and two-dimensional signals that preserve the relevant information in the analyzed event streams while reducing the amount of data subsequently handled by later stages of pattern recognition and linguistic mapping. This reduction is often by several orders of magnitude and significantly reduces the computational burden for subsequent processing stages.

The above-mentioned work was in the acoustic domain where aircraft types were recognized from their acoustic signatures. An extension of these methods of system identification, using a variation of auto-regressive modeling, has enabled us to identify certain physical aspects of the signal source as well. These methods are applicable to voice recognition and verification, as well as other episodic signal streams. We are presently attempting to produce a set of non-adaptive models that yield certain feature vectors based on the notion of a receptive field [Hubel, 1988].

4.2. Wavelets for Information Transduction

The process of extracting useful information from data streams can be viewed as a process of transducing the data into information. Information here may be defined as any data that is useful to the task or process at hand. This scope can be narrowed to an information-theoretic aspect when the concept of novelty is introduced. The purpose of the signal processing is to perform the first stages of this information transduction. A particularly effective method is provided by the wavelet transform—it is both efficient and has a firm foundation in the theory of Hilbert spaces where the machinery of orthornormal basis sets is fully applicable.

The underlying concept of the wavelet transform is almost 50 years old, but the field itself has been in existence for about a decade. Gabor, in 1946, introduced what is now called the windowed Fourier transform. This technique is quite useful but suffers from some serious mathematical defects that can both destroy its utility and fool the user with spurious effects. Wavelets overcome these drawbacks and greatly extend the utility of Gabor's original ideas. Recent work, primarily by the French school, has put the topic on a firm mathematical footing. For an excellent introduction to the subject, see the recent review article by Rioul and Vetterli [1992]. Applications are appearing in data compression, signal and time-series analysis, quantum physics, as well as in certain areas of applied mathematics. A central reason for considering wavelets as a preprocessing stage for intelligent systems was given by Mallat [1989]: a multifrequency channel decomposition (as produced by wavelet analysis) seems to occur in the human visual cortex. Thus, we can incorporate an approximation to a known feature of mammalian sensory processing inexpensively by using wavelets.

4.3. Simulated Evolution & Genetic Search

Over the past four decades, several methods of optimization have been developed by various researchers. The "classical" methods may be defined as those relying on gradient descent, conjugate gradient methods, and the simplex approach. These methods are quite efficient for well-behaved functions, but quickly become ineffective or non optimal when noisy systems or unpleasant functions are the target. The need for optimization arises in the general intelligent system much as in its neural-net component discussed above. In particular, the simulation becomes more efficient and more closely approximates the natural system being modeled when properly optimized. Correct functionality of the system can rest on an optimization problem: the correct set of states for the FSMs are needed for the system to function at the necessary level of discrimination.

We discuss three methods of optimization that we have had occasion to use on particular problems at ORNL. We found that each has its own niche and none is generally applicable. Since the success of any of these methods is problem dependent and there is no sure way to choose the best method for a particular problem (unless the problem is well-understood), the only method left to the designer of the intelligent system is to incorporate each of these optimization techniques into the "genetic pool" of the evolving system.

4.3.1. Simulated annealing

Simulated annealing is a recent method shown to be effective for optimizing systems described by large parameter sets. The notion is to parametrize a search over a parameter space with an abstract temperature [Kirkpatrick, *et al*, 1983] in a manner analogous to annealing a metallic or glass sample. First used in signal processing for identification purposes by Geman and Geman [1984], it soon became a topic of some importance. The cooling schedule proposed by these workers, however essential to their convergence theorems, is too slow to be practical in many cases. Harold Szu [Szu and Hartley, 1987] recognized this drawback and proposed an algorithm based on the Cauchy distribution instead of the Gaussian distribution. Both algorithms base acceptance of an "up-hill move" on a Boltzmann factor. The result of Szu's algorithm is convergence in reciprocal linear time rather than in reciprocal logarithmic time, with the added benefit of preserving the capability of taking occasional long jumps in parameter space so the entire space is effectively sampled at all times.

We have obtained satisfactory results using Cauchy simulated annealing on certain polynomial delay systems (ARMA and ARMAX models), with convergence being at least an order of magnitude faster than Boltzmann annealing. The standard gradient descent methods used for many neural network applications may be replaced by Cauchy annealing also, providing both faster convergence and avoidance of local minima.

4.3.2. Genetic algorithms

When there is a good "natural" mapping between the parameter space describing a natural system and a set of bit strings, the genetic algorithm (GA) as developed by Holland [1992] may prove most effective. The GA is defined by its genetic operators on bit strings (crossover, inversion, and mutation) and its method of credit allocation. The genetic operators take large steps in parameter space by exchanging sections of each individual's descriptive bit string (the "chromosome"), inverting selections of a given bit string, or mutating a bit string by altering one or more bits. The credit-allocation scheme is based on relative fitness over the entire population of solutions to the problem so that a given individual has a probability of entering into the next generation (possibly altered by one of the genetic operators) according to its relative fitness.

A particular advantage of the GA is its preference for discrete systems. There are more effective methods for functions described by continuous parameter sets.

4.3.3. Simulated evolution

The methods developed over the last three decades by L. Fogel and D. Fogel [David Fogel, 1991] under the descriptive term "simulated evolution" (SE) are also quite effective at optimizing a large class of functions—including the ARMA model mentioned above. For a certain large class of problems, simulated evolution outperforms the standard GA in computational ability and has less of a probability of getting stuck on a local minimum of the cost function. A defining characteristic of SE is its reliance on small changes in parameter space. While excellent at optimizing continuous functions, its success for discrete problems is somewhat lacking. The other central characteristic of SE is as an alternate to the credit allocation scheme used for the GA. SE holds a competition between pairs of randomly selected individuals, and the population is ranked according to the total scores that each member attains in the competition. No selection is made according to the fitness of an individual other than by this competition based on relative fitness scores. The top-scoring half of the population is allowed to generate offspring (one each) by the method of small random variations in each individual's parameter set, while the bottom half is discarded. This selection method is a valid driving force for evolutionary progress and performs effectively in both the discrete and continuous cases.

5. Controlling The Intelligent System

A simulation can be thought of as a model residing in a computer. If the model is sufficiently complex, it must be supplied with either internal or external control to be useful. A neural network can provide needed input to the control section by monitoring the state of the entire complex system. Additional control is to be provided at the top level by human interaction with the system. An effective way of obtaining such control is through the use of fuzzy set techniques.

The culmination of all the techniques we have discussed so far is embodied in the concept of an anticipatory system. A high-level feature of any intelligent system, from insect to small mammal and beyond, is that of anticipation. This anticipation is intimately connected with the environment (for the organism) or the context of the stimulus (for the information system).

5.1. Neural Networks

A novel set of neural network techniques for complex system monitoring and fault detection have been developed in a collaboration between ORNL and the University of Tennessee, [Uhrig, 1991], [Tsoukalas, *et al*, 1991], [Gao and Uhrig, 1992]. The key features of Uhrig's work lie in its application to complex systems, in this case, nuclear power plants. He has addressed plant monitoring, performance analysis, and failure modes. Each of these techniques has a part to play in a complex intelligent system.

5.2. Fuzzy Logic Systems

Communication with the intelligent information system is as essential as controlling it. The linguistic mappings and calculations on linguistic variables with linguistic values provided by fuzzy set theory give an exceptionally efficient means of communication with a potentially intelligent construct. It may be that the machinery provided by fuzzy sets is insufficient for the task. However, as above, we will choose to embrace this method because it exists, is well known, and is simple to employ. Should it prove inadequate, it can be amended or replaced.

A small vocabulary can easily be defined for purposes of query and control. The next stage of defining initial linguistic values and their corresponding fuzzy sets is likewise a straightforward task. At this point, a FSM or a CA would be operationally responsible for each of the fuzzy variables. The architectural design of such a system is a task best left to one of the simulated evolutionary techniques discussed above.

5.3. Time Series Modeling & Prediction

If we need to extend our list of properties possessed by intelligent systems, the idea of predictive ability has high priority. An appealing view of complexity [Fogel, *op. cit.*, p 21] holds that it was invented by the evolutionary process "because it is necessary to discriminate between, recognize, and predict environmental 'strings' of increasingly longer sequence, ultimately to the level of environmental noise." Our intelligent information system must able to predict not only such relatively easy things as system utilization and relate each user to possible information request categories, but actually aid the user much as an intelligent companion or servant would. An expert system with this type of interactive companionship, *The Scholar's Companion*, is under development by Park and Wood [1993].

As an example of a predictive subsystem, consider the nervous system of the horse shoe crab. Triggered by waning twilight, its brain initiates a series of chemical reactions that enhance the animal's photoreceptors by many orders of magnitude, preparing the creature for the essential business of species propagation by moonlight [Barlow, 1990]. Not only is this a predictive process, but it is anticipatory in that the animal's capacity for behavior is altered according to an internal model of the environment, namely the dramatic change in illumination levels that correspond to the diurnal cycle are anticipated many hours before they actually occur.

5.4. Anticipatory Systems

The culmination of these techniques of model building with finite-state machines as elementary building blocks and the predictive methods developed for time series is the anticipatory system [Rosen, 1985]. In its essentials, an anticipatory system is a model of a natural system having a means of communication with its environment. At its core, it possesses a model of itself, its own behavior, and a model of its environment. It also has the ability to adjust the states of its internal models given information from the environment, interact its model of self with its model of the environment faster than real time, and thereby obtain a predictive anticipation of the future state of the entire system. Such anticipation is a salient feature of any intelligence capable of coping in a complex environment.

6. Summary

In this paper, I have discussed a number of seemingly disconnected ideas, methods, and techniques with a view of forming a synthesis. The object is to show that such disciplines as signal processing, wavelet transforms, pattern recognition, and anticipatory systems, are all pieces of the same puzzle: a perceptual appratus for the intelligent system. The glue holding these pieces together is to be found in the ideas of genetic programming and the various optimization methods mentioned.

An essential aspect of intelligence that was neglected in the above discussion concerns the role of concepts and the process of concept formation. The perceptual model put forth here has the necessary "hooks" for concept formation based on clustering and pattern recognition. No information management system can be an adequate intelligent companion until it possess a set of functional concepts recognized by both the system and its human users. The treatment of concepts, even in the broad view taken here, requires at least as much attention as given above to the consideration of perception.

Although no intelligent construct exists other than those created by natural evolution, there are many ongoing attempts ranging from sophisticated efforts to realize formal models of the human brain as functional software modules, to naive assemblies of neural networks. The argument presented here is that a technique by itself cannot produce intelligence—it is merely one component among many that must be assembled into a complex system before intelligence can be attained. Taking a hint from nature, intelligence must be assembled from a myriad of methods and techniques using as close an approximation to the process of evolution that we can devise. As engineers and scientists, we feel we can provide some measure of structure as long as we are not prohibiting necessary interactions or processes: here we must tread carefully, and perhaps leave the structure to the evolutionary process itself. Our job will then be to assemble the raw material of the simulated evolutionary process and provide a clear measure of the goal to be attained.

7. References

J. S. Albus, "Outline for a Theory of Intelligence," *IEEE Trans. Sys. Man, and Cyber.*, Vol. 21, Number 3, 473-510, 1991.

P. W. Anderson, "Spin Glass Hamiltonians: A Bridge Between Biology, Statistical Mechanics and Computer Science," David Pines, editor, *Emerging Syntheses in Science*, Addison Wesley, Redwood City, Ca., 1988.

Robert B. Barlow, Jr., "What the Brain Tells the Eye," *Scientific American*, pp. 90-95, April 1990.

L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*, Academic Press, section 1.4 and 1.5, Boston, 1990.

W. B. Dress, "On Doing Science By Computer," part of a Technical Report to be published by the U. S. Air Force, Wright-Patterson AFB, Dayton, Ohio.

David B. Fogel, *System Identification Through Simulated Evolution*, Ginn Press, Needham Heights, Mass., 1991.

Stuart Geman and Donald Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Patt. Ana. Mach. Intell.*, Vol. PAMI-6, No. 6, 1984.

Zhichao Guo and Robert E. Uhrig, "Use of Artificial Neural Networks to Analyze Nuclear Power Plant Performance," *Nuclear Technology*, 1992.

John H. Holland, *Adaptation in Natural and Artificial Systems*, second edition, MIT Press, Cambridge, 1992.

David H. Hubel, *Eye, Brain, and Vision*, W. H. Freeman, New York, pp. 39- 57 1988.

B. A. Hubermann, "Computing With Attractors," David Pines, editor, *Emerging Syntheses in Science*, Addison Wesley, Redwood City, Ca., 1988.

R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in Mathematical System Theory*, McGraw-Hill, New York, 1969.

S. W. Kercel and W. B. Dress, Special Projects Internal Report K/ITP-482, Oak Ridge National Laboratory, January 1992a.

S. W. Kercel, W. B. Dress, M. R. Moore, and R. A. Hess, Special Projects Internal Report K/ITP-494, Oak Ridge National Laboratory, March 1992b.

S. Kirkpatrick, C. C. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, pp 671-680, 1983.

John R. Koza, *Genetic Programming*, MIT Press, Cambridge, Mass., 1992.

C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, Addison Wesley, Redwood City, Ca., 1992

Stephane G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. on Acoustics, Speech, and Sig. Proc.*, Vol 37, No. 12, pp 2091 - 2110, December 1989.

David Marr, *Vision*, W. H. Freeman, San Francisco, 1982.

Jack Park and Dan Wood, "The Scholar's Companion," part of a Technical Report to be published by the U. S. Air Force, Wright-Patterson AFB, Dayton, Ohio.

Tomaso Poggio and Federico Girosi, "Networks for Approximation and Learning," *Proc. IEEE*, Vol. 78, No. 9, 1990.

Olivier Rioul and Martin Vetterli, "Wavelets and Signal Processing," *IEEE SP Magazine*, October 1991, pp. 14 - 38.

Robert Rosen, *Anticipatory Systems*, Pergamon Press, New York, 1985.

Harold Szu and Ralph Hartley, "Fast Simulated Annealing," *Physics Letters A*, Vol. 122, No. 3,4, pp. 157-162, 1987.

Lefteri H. Tsoukalas, Andreas Ikonomopoulos, and Robert E. Uhrig, "Hybrid Expert System—Neural Network Methodology for Transient Identification," *Proc. Am. Power Conf.*, 1991.

Robert E. Uhrig, "Potential Application of Neural Networks to the Operation of Nuclear Power Plants," *Nuclear Safety*, Vol. 32, No. 1, 1991.

Stephen Wolfram, "Complex Systems Theory", David Pines, editor, *Emerging Syntheses in Science*, Addison Wesley, Redwood City, Ca., 1988.

DATE

FILMED

2/2/94

END