Artificial Intelligence: A Tool for System Dynamics

Janet Marjorie Gould
Massachusetts Institute of Technology

ABSTRACT

This paper presents the findings of my research in artificial intelligence applications for system dynamics. The sudden appearance of microcomputers in homes, schools, and businesses has opened an opportunity for dissemination of system dynamics to a wider audience than we could ever hope to reach with the earlier computer technologies. This opportunity should not be lost by clinging to obsolete, or soon to be obsolete, technologies. User-friendly micro-based software should be immediately available for those individuals, schools, and corporations who are interested in systems thinking. The demand for such systems far surpasses the current supply. Artificial intelligence software is now available for microcomputers. This new software development can significantly improve current and future systems for the novice and the experienced system dynamicist.

INTRODUCTION

Artificial intelligence has recently become one of the most publicized fields of computer technology. This success has many implications for system dynamicists. There are many similarities between current system dynamics and artificial intelligence research. The problems being solved by artificial intelligence for complex physical systems are very similiar to problems system dynamicists have been solving for complex social systems. This paper presents an introduction to basic artificial intelligence research and possible applications for system dynamics. A more detailed study will follow in a later paper.

The recent commercialization of artificial intelligence, especially expert systems, will eventually allow some of the simpler ideas of system dynamics to be disseminated to a wider audience. The lack of an available human expert will be compensated for by an available artificial assistant.

I will discuss some of the current research in artificial intelligence and particularly that research which will be potentially useful for system dynamics. I will begin with a brief discussion of the definition of artificial intelligence. A brief description of current artificial intelligence topics is supplied with primary emphasis given to expert sytems.

Artificial intelligence may give us the opportunity to closely examine the way in which we present system dynamics to others. It may help us to develop a more innovative approach. Explanation and discussion of these new technologies will yield important advances in our research.

WHAT IS ARTIFICIAL INTELLIGENCE?

The leaders of the field of artificial intelligence have offered different definitions of what it is and should be. A widely agreed upon definition is not easily found. Below are listed some rough definitions of intelligence provided by major researchers in the field.

> A distinctive aspect of what we call intelligence is the ability to solve a wide range of new, different kinds of problems. (Minsky 1985, p. 128)

> Artificial intelligence is part of the grand attempt to understand thinking. The programs we write are experiments...the real results will be a new kind of understanding of ourselves, an understanding that is ultimately much more valuable than any program. Being able to learn from experience and apply that knowledge in relevant situations is an important step toward actual intelligence. (Schank 1985, p. 155)

> Artificial intelligence is concerned with extending the capacity of machines to perform functions that would be considered intelligent if performed by people. (Papert 1980, p. 157)

> Artificial intelligence isn't about creating smart computers. It is about something much more interesting: intelligence. Mind. The nature of thought. -- John Seely Brown, Xerox Corporation (Waldrop 1985, p. 39)

> Artificial intelligence is the study of ideas that enable computers to be intelligent. (Winston 1984, p. 1)

One underlying debate concerns how broadly to define artificial intelligence. Roger Schank, from Yale University, seems to have one of the stronger opinions about what artificial intelligence should be. Schank believes that artificial intelligence focuses on intelligence, something that remains mysterious and elusive. Most good artificial intelligence programs aren't terribly useful, and many very useful "smart" programs aren't artificial intelligence at all. He thinks that rule-based programs (a type of expert system) "do not attempt to reason the way a human expert would and therefore are not a true application of artificial intelligence" (Schank 1985, p. 152). Other researchers have equally strong opinions, but as a rule seem to take a much broader definition of what type of research is included in the field of artificial intelligence.

For the purpose of this paper, I will take the broader view of artificial intelligence, one which would include rule-based, expert systems.

CURRENT ARTIFICIAL INTELLIGENCE RESEARCH

Figure 1 shows many of the areas of research which have been labeled artificial intelligence. As mentioned earlier, whether or not they really are artificial intelligence is debatable, but for now we will except them as basic research moving in the direction of machine reasoning.

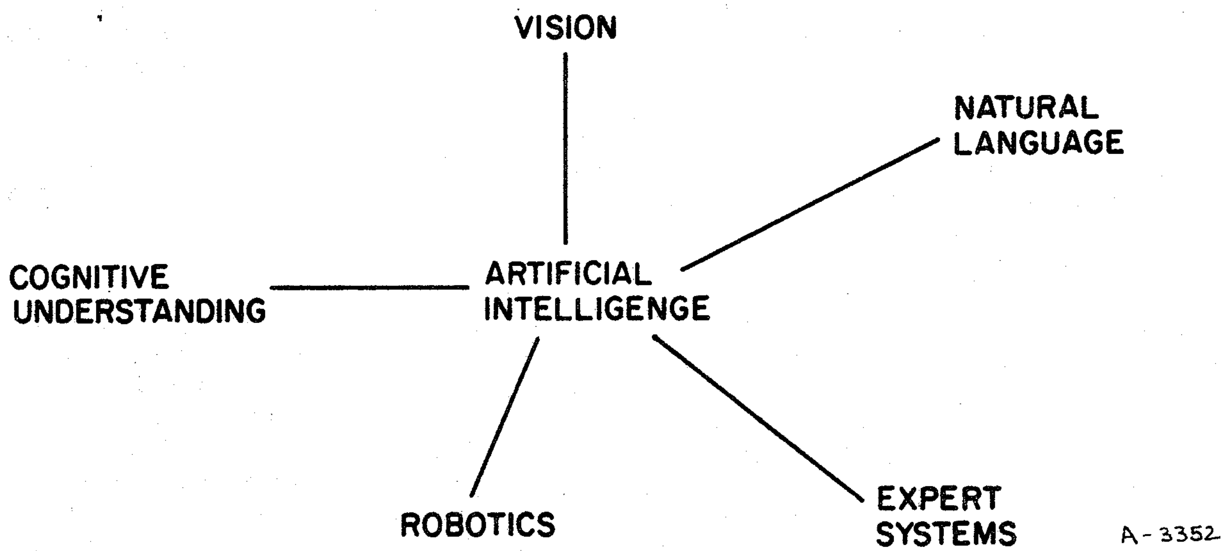

Figure 1 Artificial Intelligence Research Areas

Expert systems, natural language understanding, and robotics, are the major areas of artificial intelligence which are being commercialized. There has been a concerted effort to make artificial intelligence available to the general public. This is being accomplished by creative software development. Simplified presentations of artifical intelligence are attracting considerable public attention. These ideas should not go unnoticed. Simplified presentations of key system dynamics concepts could help us disseminate our ideas to the general public.

Robotics is being successfully used in large manufacturing plants around the world. There are robots that walk and adjust their height for low doorways, robots that are skilled assembly line workers, and robots that walk up stairs. I find robotics fascinating, but not clearly relevant to the field of system dynamics at this time.

Vision is another active area of artificial intelligence research. Vision is finding wide application in the military. How is a threatening military tank different from a native produce truck or even the surrounding terrain? These types of questions are vital for understanding object differentiation on the human as well as machine level.

The study of cognitive science or understanding is particulary interesting especially for those who are concerned with educational uses of system dynamics. Studying how a computer might learn will also allow us to gain insight into how humans learn. Human learning can also give us insight into how computers might be programmed to learn. Cognitive understanding explores our use of language and our memory from a semantic and syntactic perspective. It is quite clear that our brains allow us to easily differentiate between junk, a pile of useless things or a Chinese ship, based on the context in which they are discussed. Giving a computer the ability to differentiate in this way is an extremely interesting and complex challenge. The human memory is extremely powerful tool for complex learning and creating of new ideas.

The way in which system dynamics is taught at the university level could certainly be enhanced by a better understanding of the way students learn. And anyone who is interested in introducing system dynamics at the elementary or high school level, would certainly benefit by exploring research in cognitive science.

Natural language is becoming commercially available on small enough systems that the general public can begin to participate with applications of their own. Natural language research includes developing interfaces with the computer that will allow a user to communicate with the computer almost entirely in the English language. These interfaces can be utilized by entering information through the traditional keyboard or verbally. There are some voice recognition systems available, but currently these are severely limited by not recognizing more than one person's voice. The goal of future research would be to allow any user to communicate with the computer. Natural language communication through the keyboard is commercially available and will be one of the most popular artificial intelligence products of the near future.

Expert Systems are the latest commercially successful artificial intelligence product. Expert systems try to emulate the knowledge of an expert. Some of the well known systems are MYCIN, PROSPECTOR, and XSEL. These systems are used to identify bacterial infections, for geological surveying, and to configure computer systems, respectively. MYCIN has been useful for physicians making complex medical diagnoses, PROSPECTOR for finding specfic occurances of certains ores, and XSEL for the salesperson faced with the complexities of designating the correct components and peripherals for a Digital Equipment Corp. computer system. XSEL is based on an earlier system XCON which is used to configure computers like the VAX 11/780 or PDP-11.

A small subset of expert systems research is being done in rule-based systems. Other systems are based on nets and frames, which are more complex representations than rules. Frames are similar to records in Pascal. That is an object, such as an address, is associated with a person's name, street, city, state, and zip code. A net can connect different frames together. Addresses of different computer stores would be connected to restaurant addresses for a directory, such as the Yellow Pages, to form a net.

Rule-based systems refer to the way in which knowledge is represented in these particular types of expert systems. They are based entirely upon rules of the if-then type. For example, if it is raining, I will bring my umbrella. The decision tree for this type of statement would begin with the statement, it is raining. A response of either true (it is raining) or false (it is not raining) would be determined. True and false would be separate branches of the tree. If true, then I will bring my umbrella, and the tree would branch out from there. If false, the other branch would be chosen and continue from there.

```
                          M.1
                   Teknowledge, Inc.
          Copyright (c) 1984,  All rights reserved.
             Version 1.0,   Serial Number 0165

M.1> load wine.
M.1> go.
Is the main component of the meal meat, fish or poultry?
>> meat.
Does the meal have veal in it?
>> no.
Does the meal have a sauce on it?
>> yes.
Is the sauce for the meal spicy, sweet, cream or tomato?
>> why.
    Your answer to this question will help determine whether the
        following rule is applicable in this consultation.

    rule9:
       if   not main-component=fish and
            has-sauce=yes and
            sauce=tomato
       then best-color=red.

    The following knowledge base entries are also being used:

          rule26            (a rule)
          rule45            (a rule)
Is the sauce for the meal spicy, sweet, cream or tomato?
>> show rule26.
    rule26:
       if   best-color=red
       then recommended-color=red.
Is the sauce for the meal spicy, sweet, cream or tomato?
>> unknown.
Do you generally prefer red or white wines?
```

Reprinted with permission of Teknowledge, Inc.

A-3348

Figure 2 Wine Program - the use of "why" and "unknown"

Figures 2, 3, and 4 are examples of the printed ouput of a rule-based system called M.1 that contains knowledge about the correct wine to be served with a meal. I chose this example so it would be easy to follow. M.1 is available from Teknowledge, Inc. It includes some examples of expert systems such as the wine program, a banking program, an engineering program, etc. M.1 is designed with two separate components, the inference engine and the knowledge base. The inference engine manipulates the knowledge supplied by the user and provides a simplified parser to be able to understand English responses. The knowledge is created by the user with a series of if-then rules. M.1 will accept about 300 rules.

M.1's inference engine is backward chaining. It begins with a conclusion and searches backward trying to satisfy specific conditions. For example, in an animal identifying program, suppose the conclusion is that the animal is an elephant. What conditions must be satisfied to make this true? If the animal has a trunk, and a tail, and is a mammal, then it is an elephant. Tracing backward from the conclusion that the animal is an elephant, we must examine the conditions that must be satisfied about the trunk, tail, and mammal. For example, if the animal nourishes its young with its own milk and has hair, then it is a mammal. We then need to trace back further to the conditions about hair and nourishment. This process continues until all conditions are satisfied to conclude that the animal is an elephant.

```
>> white
>> .
Is the flavor of the meal delicate, average or strong?
>> why.
   Your answer to this question will help determine whether the
        following rule is applicable in this consultation.

   rule2:
      if   tastiness=delicate
      then best-body=light cf 80.

   The following knowledge base entries are also being used:

         rule14               (a rule)
         rule45               (a rule)
Is the flavor of the meal delicate, average or strong?
>> average.
Do you generally prefer light, medium or full bodied wines?
>> medium.
Do you generally prefer dry, medium or sweet wines?
>> medium.
   wine = zinfandel (90%) because rule54
   wine = cabernet-sauvignon (90%) because rule54
   wine = pinot-noir (64%) because rule55
   wine = gamay (64%) because rule45
   wine = burgundy (30%) because rule56
   wine = valpolicella (30%) because rule53
```


A-3349

Figure 3 Wine Program - rules and results

M.1 is essentially ignorant until the user provides the rules. Figure 2 shows an example of an if-then rule, "rule9". The wine program essentially is a dialogue with the system. After the "go." statement the program begins a series of questions which will help it traverse the tree of rules to come to a final conclusion about which wine is appropriate. The user is allowed to question the system at any time by typing "why.". In response, the system will display the rule it is currently using to try to reach a conclusion. The user may ask the system to show what is contained in a particular rule, such as "rule26" in Figure 3. If the answer to one of the program's questions is not known, the user may type "unknown." and the system will ask another question.

```
M.1> show.
     wine = zinfandel (90%) because rule54
     wine = cabernet-sauvignon (90%) because rule54
     wine = pinot-noir (64%) because rule55
     wine = gamay (64%) because rule45
     wine = burgundy (30%) because rule56
     wine = valpolicella (30%) because rule53
     recommended-color = red (90%) because rule26
     best-color = red (90%) because rule6
     main-component = meat (100%) because you said so
     has-veal = no (100%) because you said so
     has-sauce = yes (100%) because you said so
     sauce was sought, but no value was concluded.
     preferred-color = white (100%) because you said so
     recommended-body = medium (64%) because rule18 and rule15
     recommended-body =
     wine = zinfandel (90%) because rule54
     wine = cabernet-sauvignon (90%) because rule54
     wine = pinot-noir (64%) because rule55
     wine = gamay (64%) because rule45
     wine = burgundy (30%) because rule56
     wine = valpolicella (30%) because rule53
     recommended-color = red (90%) because rule26
     best-color = red (90%) because rule6
     main-component = meat (100%) because you said so
     has-veal = no (100%) because you said so
     has-sauce = yes (100%) because you said so
     sauce was sought, but no value was concluded.
     preferred-color = white (100%) because you said so
     recommended-body = medium (64%) because rule18 and rule15
     recommended-body = full (30%) because rule16
     recommended-body = light (30%) because rule14
     best-body = medium (60%) because rule3
     best-body = full (30%) because rule3
     best-body = light (30%) because rule3
     tastiness = average (100%) because you said so
     preferred-body = medium (100%) because you said so
     recommended-sweetness = medium (100%) because rule41
     best-sweetness was sought, but no value was concluded.
     preferred-sweetness = medium (100%) because you said so
M.1>
```


Figure 4 Wine Program - explanation A-3350

In Figure 3, "rule2" is displayed. At the end of the rule is printed, "cf 80". This indicates that the certainty factor is 80. The certainty factor, supplied by the person who provided the knowledge, indicates that if the response, "delicate", is received when the user is questioned about tastiness, then it is about 80% likely that the best body of the wine is light. The last question in Figure 3 has the response, "medium". The last statement of figure 4 shows that the preferred sweetness of the wine is definitely medium (100%), "because you said so". That is, I supplied the information with my response of "medium". I did not provide a certainty factor, so the system assumes a factor of 100. The last six lines of Figure 3 show which wines the program would recommend. Figure 4 shows the user what conclusions were reached along the way and which rules they were based on. Those rules could then be examined for further understanding.

What does all this information imply for system dynamics? The goal is to harness the power of the computer to manipulate heuristics as well as specific algorithms and numeric data. The ability to query a system about its line of reasoning and to be able to use the answer "unknown" in response to a question are extremely powerful ideas. We all realize that in conventional programming methods this power is not available. During a standard interactive program a response of "unknown" would (unless specifically programmed as an acceptable response) cause an error in the program. And certainly attempting to ask "why" or how the program was processing information would be futile. These capabilities present us with the first attempts to model human reasoning. Humans can explain their paths of reasoning. We are capable of reaching conclusions although certain information is unknown.

ARTIFICIAL INTELLIGENCE AND SYSTEM DYNAMICS

In order to make artificial intelligence available to the general public, an effort has been made to make certain ideas and software packages available which one can understand without a Ph.D. in the field. The field of system dynamics has faced some difficulty in being accepted by other academics and certainly by the more general audience because of perceived complexity. As in other sciences and related disciplines, we should devote some time to simplifying the key ideas of systems dynamics. "The central task of a natural science is to make the wonderful commonplace: to show that complexity, correctly viewed, is only a mask for simplicity; to find pattern hidden in apparent chaos" (Simon 1969, p. 1). The social sciences should be challenged to simplify certain complexities. This is what is being accomplished by Papert through his development of LOGO microworlds to educate young students about math, physics, etc. System dynamics should begin to take this approach not only for students, but for corporate executives and anyone else who displays an interest in our research. We should be ready to take advantage of new artificial intelligence applications as they emerge over the next few years. As an academic discipline and a public policymaking perspective, system dynamics could thrive on these developments. "Within the next two decades, new capabliities such as computer-driven video disks and video tapes--along with clever use of computer instructions games and self-paced instruction packages should push the field to new levels of accessibility and appeal" (Gould 1983, p. 19).

My current research is exploring combinations of natural language and expert systems for system dynamics. The software currently available for system dynamics models has the same drawback as most other software, it does not allow standard English input. By designing interfaces that incorporate a "system dynamics parser" with expert knowledge and the ability to generate dynamic simulations, we would then have a system usable by people with different levels of expertise.

One of the major difficulties in simplifying the system dynamics approach is the in-depth understanding one needs to become a competent model builder and interpreter of model behavior. These abilities are not gained after one course at the graduate level. It can take many years of model building before one is a good modeler. Some people never become competent model builders even after years of practice. The skills of a Jay Forrester are rarely mastered. If Forrester's knowledge were available in a database, we could eventually develop a computer system which would generate its own outstanding models. In artificial intelligence, knowledge engineers try to extract knowledge from experts to be entered into an expert system. The MYCIN program, mentioned earlier, has the knowledge of more than one physician. This extensive knowledge can then be used to guide those who do not know "all the right answers". Of course, one still needs the right questions. Although MYCIN contains information about infectious diseases, the user would not be able to interact with the system without knowing the correct terminology.

Knowledge acquisition tools are being developed, but much more slowly than expert systems themselves. It has become clear that no knowledge base is ever complete. A dynamic process is needed for changing the set of rules. As new information is needed, the knowledge base must be changed. Unfortunately, the knowledge engineer is often unavailable, so the user is expected to improve the system. Knowledge acquisition systems are needed. It would seem reasonable to expect that future expert systems will be supplied with knowledge acquisition tools. TEIRESIAS was developed to allow a user to update and expand the knowledge base of MYCIN. A system is being developed to improve the current version of XCON. As these systems are developed, it will certainly be useful for system dynamacists to be aware of them and be ready to create their own systems.

Papers by Davis and Kuipers of MIT on structure and behavior of physical systems are interesting for our work in social systems. Kuipers paper is "concerned with the qualitative simulations of physical systems whose descriptions are stated in terms of continuously varying parameters. The examples presented demonstrate a representation for qualitative reasoning about causality in physical mechanisms. The system as described in this paper has been completely implemented in MACLISP. The structural description is essentially a qualitative form of a differential equation, specifying a set of parameters which characterize the state of the mechanism and a set of constraints holding among the parameters. Qualitative simulation produces a behavioral description which specifies the ordinal relationships and directions of change of the parameter values at each point in time" (Kuipers 1984, pp. 189).

Davis is interested in troubleshooting electronic circuits. Davis' research "goal is to develop a theory of reasoning that exploits knowledge of structure and behavior. The initial focus is troubleshooting digital electronic hardware, where we have implemented a system based on a number of new ideas and tools. We have developed languages that distinguish carefully between structure and behavior, and that provide multiple descriptions of structure, organizing it both functionally and physically. We have argued that the concept of paths of causal interaction is a primary component of the knowledge needed to do reasoning from structure and behavior. We describe a new technique called constraint suspension, capable of determining which components can be responsible for an observed set of symptoms. We show that the categories of failure, previously derived informally, can be given a systematic foundation. We show that the categories can be generated by examining the assumptions underlying our representation" (Davis 1984b, p. 3). Davis has also developed a simple circuit drawing system that permits interactive entry of pictures which represent the circuit structure and are then translated in to the proper Lisp code. Davis is creating a type of expert system for physical systems which could eventually be very useful for social systems. Of course, at this level physical systems are easier to manipulate than social systems because they are more easily quantified. Davis' paper has lead me to theorize about the type of systems we can expect to be using in the future.

Figure 5 is a representation of the type of expert system which could eventually be available for system dynamics. This system would initially be designed for educational purposes, using very simple system dynamics generic structures or models. These simple models are currently a major research focus designed to demonstrate that the same basic structures are often found in many different environments. An interesting example of this which I recently encountered concerns similar structures underlying the desire for new members in a church and a health maintenance organization. At first, one would probably not recognize the similarities between a church and a health maintenance organization, but a very simple generic structure can be developed that underlies both. The desire for more members puts pressure on the ministers and doctors to interact with more people. This brings in more members. The more members there are the less time the ministers or doctors have to spend with them. This brings about dissatisfaction among members. Members will quit unless more ministers or doctors are hired.

My basic theory is that a user can communicate with a system dynamics model through an expert system. The rule-based interface would be knowledgeable about the differences between the various generic structures to which it has access. One could communicate with the interface either in English or graphically (much like Davis' system for circuits). A user would begin with a particular problem that, when understood by an expert in system dynamics, would be associated with a particular type of generic structure. The interface would take information from the user and decide which generic structure to use. The interface would then pass control on to a more complex expert system, the simulator, which contains information about specific generic structures. In the simplest configuration the fault detector would have knowledge about preferred types of behavior that are associated with a specific generic model. When that behavior is not displayed based on the user's parameter definitions or changes in structure, a fault or error detector would attempt to determine from the structure, where the fault lies. Control could be passed between the simulator, fault detector, and finally to the user for assistance.

A more sophisticated version of Figure 5 could be used to develop simple models. The user would provide information to the fault detector about the type of behavior that the user expects based on his model structure. After the model is created (in English or graphics) and simulated, the fault detector would act as a helper to the user in identifying problem areas in the model.

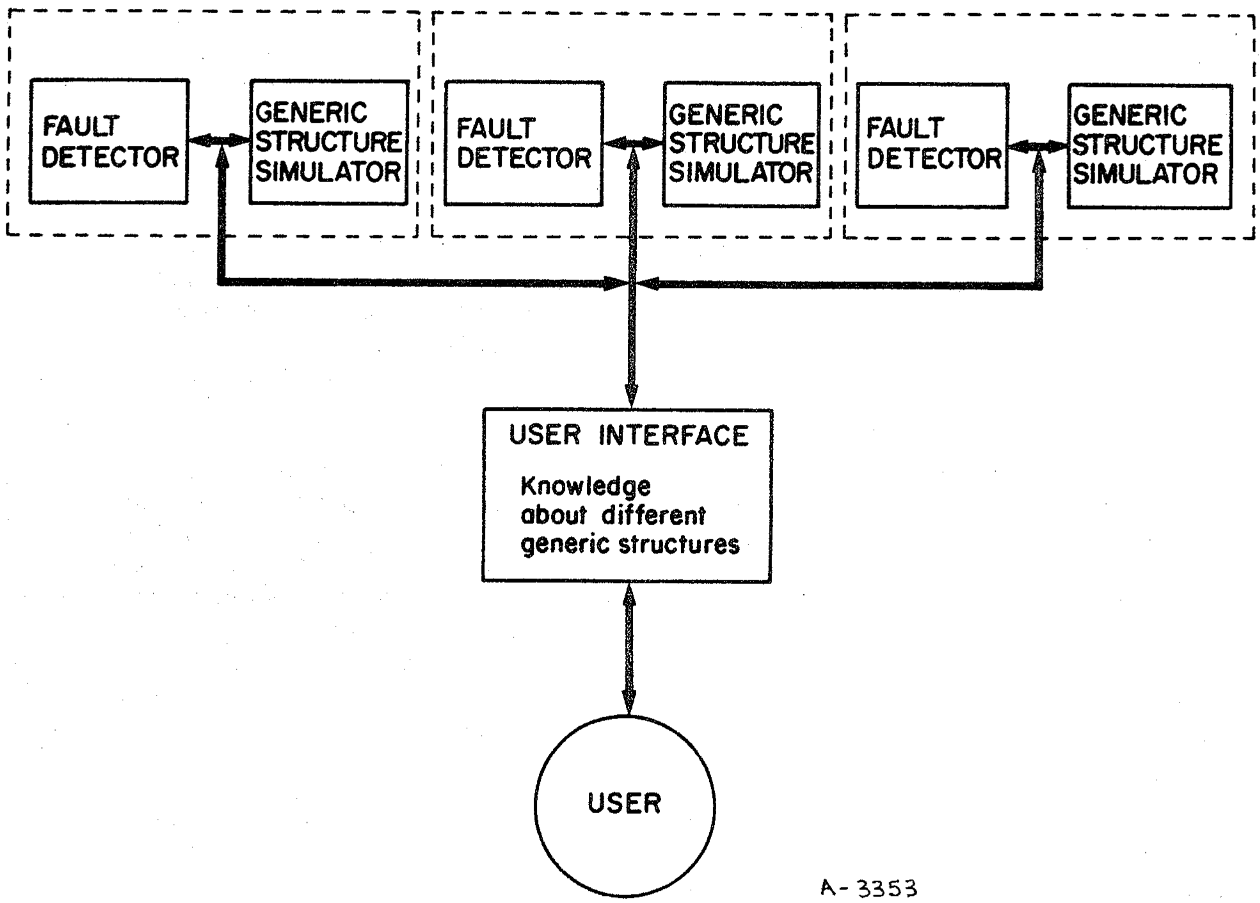

Figure 5 System Dynamics Expert System

A simple example may help to illustrate the idea shown in Figure 5. Suppose a student was interested in "playing with" a model. Suppose the student, while interacting with the interface, created a one-level model. The student's model would then become an object to simulate (in the generic structure simulator). Suppose the student made an error and thought that the model should exhibit oscillations. The model would simulate and display growth. The fault detector is then faced with a discrepancy between the student's anticipation and the actual behavior. The detector would also contain specific knowledge about the behavior of system dynamics structures. The fault detector would analyze the structure of the model in the simulator and realize that the specific structure should not oscillate. That information would then be passed back through the interface to the student, if desired.

The interaction between the student, the interface, the simulator, and fault detector provides an intelligent educational environment. Anticipating behavior based on structure is a complex task associated with physical and social systems. The complex nature of systems often creates counterintuitive behavior. To understand how behavior is generated from specific structures, one must have substantial experience creating and manipulating structures. The intelligent computer environment I have suggested creates what is necessary to encourage the learning process.

The more complex a model becomes the more knowledge the fault detector would need in order to assist the user. Eventually knowledge about phase and gain relationships, stable and unstable equilibria, eigenanalysis, etc. would be needed. For now, I would envision the early prototype of the system to be designed for a student rather than for the professional system dynamacist.

CONCLUSION

Much more research is needed before artificial intelligence can be immediately useful for system dynamics. The areas which should be explored are cognitive understanding, natural language, and expert systems. The work of Davis and Kuipers is particularly noteworthy, and should be more closely examined. Understanding the system dynamics learning and model-building processes is vital. Interacting with the computer in English would be helpful for disemminating the ideas of system dynamics to others.

I do not intend to suggest that the ideas represented in Figure 5 will be available immediately. It is generally assumed that an expert system can easily take five man-years to develop. I suggest that we build on earlier developments. The work of Kuipers and Davis can provide us with the needed base for developing our own intelligent systems. A rule-based system such as M.1 provides an opportunity for system dynamacists to experiment with and manipulate our own knowledge base. Microcomputer software such as M1, Expert Ease, Gold Hill Lisp, etc. put the power of knowledge systems in the hands of the individual. The opportunity is worth exploiting.

Figure 6 suggests system dynamics, education, and artificial intelligence combining into intelligent educational systems. We realize that "in knowledge lies the power"; but even more important is the ability to communicate effectively and thereby pass the power on to others. System dynamics gives us the ability to understand complex systems. Artificial intelligence gives us insight into our own thought structures and gives us freedom to focus on the more challenging components of learning. Education teaches us to communicate to others about our expert knowledge. Understanding, insight, and communication are the keys to our power as individuals and as a society. My goal is to give individuals the power to learn through their own discoveries.

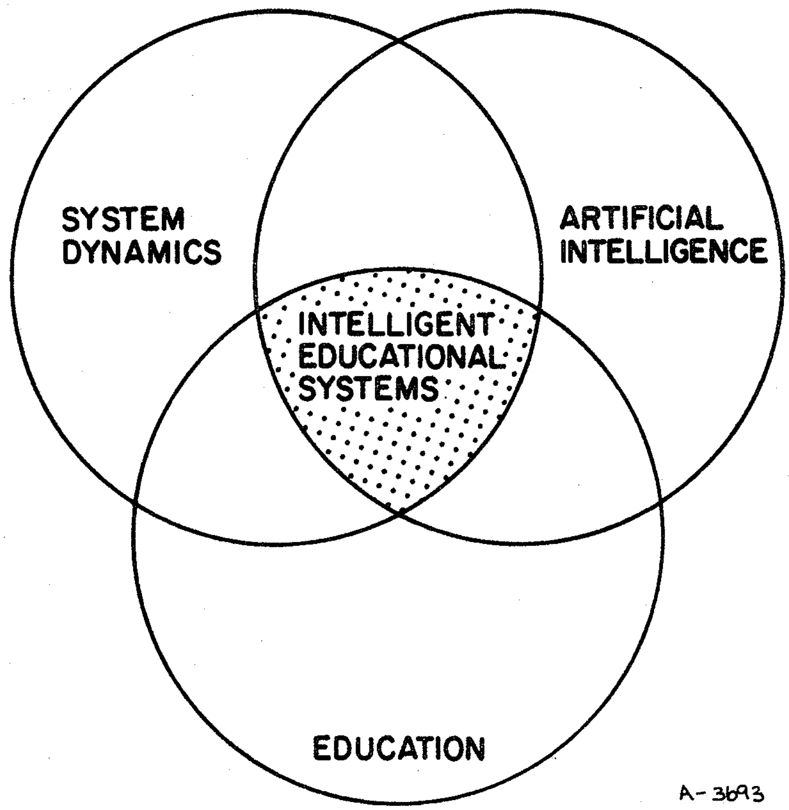

Figure 6 Intelligent Educational Systems

REFERENCES

Black, R. "Artificial Intelligence for Micros", Hardcopy, March 1985, Volume 14, Number 3, pp. 43-49.

Davis, R. "Amplifying Expertise with Expert Systems," The AI Business, M.I.T. Press, 1984, pp. 17-40.

Davis, R. "Diagnostic Reasoning Based on Structure and Behavior", M.I.T. A.I. Memo 739, June 1984.

Fogel, L. J., A. J. Owens and M. J. Walsh. Artificial Intelligence Through Simulated Evolution, John Wiley and Sons, Inc., New York, 1966.

Forbus, K. D. A Study of Qualitative and Geometric Knowledge in Reasoning about Motion, M.I.T., AI-TR-615, February 1981.

Foster, E. "Artificial Intelligence: Beyond the Buzzwords", Personal Computing, April 1985, pp. 62-69.

Gould, J. M. and D. P. Kreutzer. "New Computer Graphics Technologies for Representing the Structure and Behavior of System Dynamics Models", System Dynamics Group Working Paper D-3431, December 1983.

Jones, K. "Microsoft, Others Plan Expert Systems," Mini-micro Systems, July 1984, pp. 89-91.

Kuipers, B. "Commonsense Reasoning about Causality: Deriving Behavior from Structure," Artificial Intelligence, 24 (1984), pp. 169-203.

Michaelsen, R. H., D. Michie, and A. Boulanger. "The Technology of Expert Systems", BYTE, April 1985, Volume 10, Number 4, pp. 303-314.

Miller, R. K. "Artificial Intelligence: A New Tool for Manufacturing", Manufacturing Engineering, April 1985, pp. 56-62.

Minsky, M. "Communication with Alien Intelligence", BYTE, April 1985, Volume 10, Number 4, pp. 127-138.

Papert, S. Mindstorms, Basic Books, New York, 1980.

Probert, D. E. "Bridging the 'Interpretation Gap' in Business Policy Analysis," Presented at the Fourth International Symposium on Forecasting, July 1984.

Probert, D. E. "Towards Expert Systems for Telecommunications Policy Analysis, Computer Communications, Butterworth and Co., Volume 6 Number 2, April 1983, pp. 58-64.

Schank, R. and L. Hunter "The Quest to Understand Thinking", BYTE, April 1985, Volume 10, Number 4, pp. 143-158.

Shiel, B. "Power Tools for Programmers," Datamation, February 1983, pp. 131-143.

Simon, H. A. The Sciences of the Artificial, The M.I.T. Press, Cambridge, Mass., 1969.

Smith, E. T. "When Will a Computer Be Able to Communicate Like a 6-Year-Old?", Business Week, April 1, 1985, pp. 84-85.

Thompson, B. A. and W. A. Thompson. "Inside an Expert System", BYTE, April 1985, Volume 10, Number 4, pp. 315-333.

Waldrop, M. M. "The Machinations of Thought", Science 85, March 1985, Volume 6, Number 2, pp. 38-45.

Webster, R. "Expert Systems on Microcomputers", Computers and Electronics, March 1985, Volume 23, Number 3, pp. 69-73.

Winston, P. H. Artificial Intelligence, Addison-Wesley Publishing, Reading, Mass., 1984.

Winston, P. H. "The Lisp Revolution", BYTE, April 1985, Volume 10, Number 4, pp. 209-222.