APPEARED IN: PROCEEDINGS OF THE
FOURTEENTH ANNUAL CONFERENCE OF
OF
The Society For Information Management
(formerly SMIS)
September 20-23, 1982

THE NEED FOR ACTIVE SYSTEMS IN THE FACTORY OF THE FUTURE INTELLIGENT SYSTEMS LABORATORY

Mark Fox
Robotics Institute, Carnegie-Mellon University

Organization Theory is the study of the structure and behavior of organizations. Basic to this study is the concept of bounded rationality, first proposed by Herbert Simon. Bounded rationality states that people are bounded in their ability to operate, control, and understand complex tasks and organizations. As tasks grow, the ability to manage is reduced. In particular, bounded rationality limits the amount of information that we can absorb at any given time. It limits our span of control; our ability to tell other people what to do; to understand what they should be doing; the ability to analyze and plan what we should be doing. Anything that exceeds these bounds is too complex, that is, we are unable to effectively deal with those tasks and hence have to look for alternative methods of dealing with complexity.

Hence the concept of an organization arises, that is, the organization of two or more people to perform a task. We sub-divide the task, we increase our span of control, hierarchies arise and continue to grow into more decentralized organizations until we split up into market environments which communicate through prices alone. That is the minimum amount of information necessary to manage tasks.

Does the concept of bounded rationality have any relevance to our factories? Consider job shop, the problem of scheduling a job-shop factory. The job shop I am talking about turns out to be a Westinghouse factory. They produce turbine blades. This is a very complex product, requiring high manufacturing accuracy which few countries possess. This job shop possesses thousands of part numbers; they have been producing blades for decades and the number of part numbers they add is on the order of hundreds if not thousands per year. The production of a part requires ten to twenty operations. Milling and grinding are examples of operations. A part may be produced in more than one way. Each way may incorporate differing technology to produce the same item. In reality, factories do not always have the most up-to-date machines. Expensive machines are not scrapped when newer ones are purchased. So along with the new machines there are older ones which may perform similar operations. Hence there exists many ways in which an item may be produced. In addition, the routing selection differs by order. One order may have to be shipped quickly. Another order may not have to be shipped for two years. Orders may be for stock, or for planned maintenance six months from now. Some orders may require high quality or minimal quality. The criteria used to select a process routing may vary on an order by order basis. And each order will be contending for the same resources.

In our study of the scheduling problem, we found that in order to successfully schedule the job shop, the scheduler must communicate with almost every other department in the factory. This is not true of just the metal cutting industries. We found this in printed circuit board production and other types of manufacturing environments. In scheduling a factory, a major problem is the identification and satisfaction of constraints. Examples of constraints include alternate routings, due dates, working process time, resource availability, utilization of resources, resource substitutability, cost, and preference within the organization: "Do I prefer using this machine or another machine? Do I prefer this operation? Do we have any production goals from advanced planning? Is a critical facility supposed to be used more than another critical facility? Do I have to keep the shop stable all the time?" When composing a schedule, the scheduler has to consider all the alternatives, the complexity of the environment, and all other constraints.

If scheduling is so difficult, how well is it being performed today? In the metal cutting industry, there is eighty percent queue time. And twenty percent of the time that it is on the machine value added operations are applied. Hitachi, in their analysis of their turbine blade plant, found that a hundred white-collar personnel touched an order in some fashion or another. Why are a hundred people at the white-collar level touching it? Hitachi decided it was more important for them to reduce the number to thirty than to increase the cutting speed of the particular machine. The reason why jobs spend so much time in queues has little to do with machine speed, coordination is the problem. Another indicator of the complexity of scheduling is overhead charges. Three-hundred percent is a conservative figure. In analyzing a number of different industries, I found a five to forty-to-one ration of white-collar to blue-collar workers. It would appear that the complexity of scheduling exceeds our abilities.

Let us look at another case where bounded rationality is hurting us, that of managing product design. In particular, computer design. The design of a computer takes from three to five years. Thousands of activities must be coordinated to complete the product. More than one product may be designed in parallel, each contending for the same resources. Just like in a job shop, there is resource contention and there are constraints on the product being designed. Complicating the task are change orders which redefine the product and its design. Hence project management is complex. Currently, CPM and PERT systems are used to manage projects. They provide activity scheduling and resource assignment functions, but do not provide day-to-day management of activities and their interactions. Much of that is provided by project support personnel. How well do we manage projects? Excessive overhead charges, cost overruns, and late completions are indicators of our inability.

What is the problem? Tasks are too complex, organizations are too large, and there are too many things to keep track of. Organizations are just not as decomposable as we would like them to be. There are too many interactions among the parts. In other words, bounded rationality has limited our ability to deal with complexity. Is there any hope? Are there any solutions out there? Some people believe that the factory of the future will solve the problem. In the factory of the future we won't have to worry about complex control and efficient utilization of resources. All that will be done by machines. There will be very few people in the factory of the future. In this mythical factory you will sit down at a terminal and describe the functional characteristics of your product. The computer will design your product, program the factory machines and have the product pop out five minutes later. It sounds nice but has very little to do with what is going on in factories today. How far are we from the factory of the future? A long way. Nevertheless, there are important changes occurring. Computerization of the factory is increasing. Computer-aided manufacturing, computer-aided design, computer-aided engineering are all parts of the change. Regretably, the level of computer-aided manufacturing in industry is quite low. Maybe four percent of the industries that have tools have computer controlled or NC tools. There is much to be done before the future arrives. What does the factory of the future really look like? We will have factories composed of lots of machining cells, with intelligent sensors monitoring production, and intelligent diagnostic aids. We will have networks throughout the factory. Every tool in the factory will communicate with every other tool. Every network will have an applications level protocol so that they will be able to talk about their applications as opposed to talking in X.25.

What is really happening is not the solution to our problem. Our problem is getting worse instead of getting better. Computer-aided design and manufacturing, real time control, monitoring and diagnosis are increasing our capabilities. This increase in flexibility is reducing manufacturing times and the time we have to decide what task to perform. That is, if something goes wrong on the factory floor, we no longer have an hour or two to decide what to do about it, but have to make a decision in minutes or seconds. Though we increase our flexibility, we are decreasing our response time, and we are getting so much information back from these machines we do not know what to do with it. So I contend that matters may get worse before they get better. The question then is whether we can deal with the complexity of the factory of the future? I can answer that now. No! Why? Because we cannot deal with the complexity of the factory of today. And we have a lot of hardware and software out there. Why are these problems not being solved? There is a number of reasons. The majority of software available today is passive, that is, it reacts only to user commands. You do a capacity plan every week, or every day because you tell it to do a capacity plan, not because the factory changes. Scheduling systems are simply capacity planning systems. Due to the lack of good scheduling theories software is also autistic. It is unable to communicate effectively with other modules, other computers, and other applications. There is a difference between having a computer network, and having your application software communicate with each other. Much of our software is also idiosyncratic, that is, we do a lot of adaptation of our software; they are good for solving the problem at hand, but not adaptable or extendable to other problems in the environment.

If we are going to solve this problem of complexity, or the invisible hand of bound rationality, we are going to have to deal with the problems of real time monitoring, analysis, control, management and adaptation. That is, can we provide computer systems that will effectively operate and control the factory environment? So the question is "are there any solutions?" Before answering the question, there are some criteria I would like to place on the solution. One criterion is *robustness*. Does the solution represent all the knowledge necessary to operate a factory, and can it handle problems previously unanticipated? The second criterion is *accessibility*. Are these systems going to be accessible? That is, are you, the manager, going to walk up and use these systems with very little training, or are you always going to have an application programmer or systems analyst running these programs for you? The third criterion is *accountability*. If we question an employee's answer, why should our systems be any less accountable? A fourth criterion is *adaptability*. They must be adaptable because organizations change. The often quoted ninety percent maintenance versus ten percent creation cost for software does not imply that we built the software incorrectly. What it does imply is that we are continually adapting to changing situations. We are not fixing errors, we are adding more capability. Last and most important is the system must be *active*, that is, it must actively monitor and control the factory. It cannot sit and wait for you to tell it to go do something, it must do it on its own.

Can we solve the complexity problem? The Japanese think so. They are pursuing the creation of fifth generation computers. The goal of fifth generation computer technology is in increasing the intelligence of software in order to solve problems within organizations. And they do it at a number of levels: they do it at the interface level, the problem-solving level, the knowledge representation level, and the communication level. And the key technology to fifth generation computers is artificial intelligence. What is artificial intelligence (AI)? Artificial intelligence is concerned with the creation of machine-based intelligence. It is not an attempt to mimic human intelligence. As a matter of fact, AI people do not care necessarily how humans solve problems. They may use that as indicators of how to build intelligent machines, but it is not necessary. The goal is to build intelligent machines. AI people do it not only by creating theories but by building systems. AI began around 1954-55 at Carnegie-Mellon University and MIT. Newell and Simon at Carnegie-Mellon created a system that solved the first twenty-five problems of *Principia Mathematica* by Whitehead and Russell. This was the first evidence of machine intelligence.

One part of artificial intelligence is concerned with problem solving, that is, how can we build systems that solve problems? It is not just solving known problems, but solving unanticipated problems. We distinguish between ill-structured and well-structured problems. Well structured problems are those whose solutions are well defined. Calculating somebody's salary at the end of a month is a well-structured problem, the algorithm is well defined. Calculating the amount of time it takes to produce a product given the operations and the times is also well defined. Other problems, such as generating a process plan or a computer design are ill-defined, ill-structured. Every ill-structured problem requires a new solution. Hence, problem solving is concerned with creating solutions to previously unforeseen problems. The primary AI problem-solving technique is heuristic search.

Another concern of Artificial Intelligence is knowledge representation. How do we represent knowledge? If we are to create intelligent systems—how is the knowledge necessary to support intelligent action represented. Let us take a look at all the things that a person has to know on the factory floor. "Joe is an NC programmer." Well that is relatively easy. "The oven is in the lay-up room;" that is relatively easy; "the inserter is broken," that is a bit harder because it is a time dependent piece of information. It is true today, maybe false tomorrow. When is it true—what caused it? "The manipulator is placed above the circuit with the fingers in open position." All of a sudden we are getting into more process descriptions. How we do something? How do you represent that? "Bake time is five hours," well that is reasonably easy. "All NC drills must have their drills changed every fifty hours." Well that is an assertion. It is true of all NC drills. How do you represent that? "The average age of programmers is thirty-five." We are not talking about an individual programmer, we are talking about the set of all programmers. We are not saying that everybody's age is thirty-five, but we are saying that the average age of all programmers is thirty-five. So you have to distinguish between the instance, that is, each person and the set of all people. This is an interesting one: "The telephone is black"—what does that mean? Does it mean that all telephones are black? Does it mean that a particular telephone that is sitting on my desk is black? Does it mean that telephones by default, if I do not know, are black? You and I know what all this means, but does the computer know? The typical solution is to get the people who create the database together in the same room and create a database design document. The meaning of data is not defined by the database. By looking at the data base I cannot be sure whether all telephones are black or this individual telephone is black. Instead I have to look at how it is being used by the programs that access it.

Other research issues in Artificial Intelligence include machine learning and natural language understanding.

In order to understand the usefulness of AI research in the factory, lets look at some of its results. Game playing is one area of possible interest. To date, there have been a number of successful game playing programs which are based on AI techniques. For example, BKG, is the world champion backgammon program developed by Hans Berliner of Carnegie-Mellon. Bell is an expert rated chess program by Thompson and Condon at Bell Labs. The contribution made by game playing programs to problem-solving is in the use of search as a method to find solutions to new situations. This is basic to all problem-solving.

Another application area is medical diagnosis. There have been a number of medical diagnosis systems created. One of which is Cadeuseus by Myers and People of the University of Pittsburgh. Its specialty is internal medicine. The second system is MYCIN by Shortliffe at Stanford. Its specialty is blood infections. A third system is the Digitalis Advisor at MIT. These are all artificial intelligence based systems that perform medical diagnosis at an expert level. The MYCIN system performed as well as the majority of experts in ninety percent of the cases that it looked at. These experts were selected from across the country. MYCIN not only diagnoses the disease but recommends treatment and explains, in English, why it chose a particular disease or treatment.

The relevance of game playing and medical diagnosis applications to the factory of the future is demonstrated by these next examples. R1 is a system by John McDermott at C-MU which configures computers. Digital Equipment Corporation had a problem—they sell what they call "a la carte" computers. You can order anything you want on your computer and they will deliver it. The only problem was that it is difficult to configure. They employed a number of people whose full-time job is to take an order and figure out the individual pieces and how they fit together. It takes anywhere from a half-hour to three days for a person to configure an order and be eighty percent correct, that is, eighty percent of the orders were configured correctly. R1 takes three to five minutes and is ninety percent correct. This results in a substantial savings. A second system of interest is ISIS, developed by myself at C-MU. It performs job shop scheduling. ISIS takes a constraint-directed approach to scheduling. It represents and uses all the constraints that humans use and not simply due dates and machine capacity. ISIS constructs schedules that fit the current state of the factory because it has a knowledge representation-based model of the factory. ISIS was designed to be factory independent.

A third example of AI in factories is a system called Callisto. Callisto is concerned with project management, but at an operations control level. The purpose of Callisto is to automatically track, monitor, schedule and analyze the activities of a project. Hence, one has to worry about detailed modeling of activities: what resources are required by an activity, what resources are produced, consumed? Who must do it? When must it be done? What are the alternatives? Can activities be monitored so that problems may be detected? For example, I just designed a part and it requires a year's lead time to acquire the raw materials. But we want to go into prototype build six months from now. Who notices that what is produced will not satisfy the constraints of the other activity (i.e., it is needed in six months)? All these things have to be represented in the model and have to be analyzed by some reasoning system. Callisto has been designed to handle these problems.

So those are three examples in the organizational area. They are not all at the factory level; but they are at managerial and professional level as well. Many of these systems rely upon techniques such as knowledge representation, problem solving, distributive architectures, sensors, robotics and other ingredients such as data bases. So where is this heading? I began this presentation with the concept of bounded rationality and its relation to complexity. I said we have problems today in the factory and in organizations in general, and that these problems will be exacerbated in the factory of the future. This is due to increased flexibility and reduction of lead times. If we are going to solve these problems, we are going to have to build more intelligent, more active solutions. Active in the sense that it has to provide more real time control. And AI may provide part of the solution.

The final question remains to be asked. Of what relevance is this to today's MIS manager? I think today MIS departments are spending a lot of time worrying about data bases as opposed to the content of those data bases. Worrying about network connections as opposed to what is really being said at the applications level when we communicate along those networks. Concerned about PBX's as opposed to why we are communicating voice and data. Concerned about terminals, color graphics, and technology in general, when we should be concerned with the functions of the organization. In some sense we are hiding, we are covering our eyes from what is really going to hit us in the future. Our worries about PBX's, networks, and terminals are being solved by vendors. In five to ten years the vendor systems are going to be complete, and the interconnect strategies are going to be complete. The data base systems are going to be complete, and then what you are worrying about today will not be the problems you will worry about tomorrow. The problems you are going to worry about tomorrow are of the nature—how am I going to provide problem solving support for the line problems in my organization? That is, it is not strictly staff functions you are concerned with, but how are we going to operate our corporation better in its line areas? And so, what I really believe is that MIS is not going to exist in the future, at least our current conception of MIS. MIS is dying and it is dying because we are concerned with things that may not be issues five years from now, ten years from now. We should be concerned about the goals of the organization. What are we manufacturing? What are we doing as a service organization? Can we provide the right tools to solve those problems more economically and efficiently? This means that we have to start looking out and saying, what are today's MIS tools? What is out there that allows me to solve problems as opposed to just link people? In other words, we're going to have to move from our current position, of providing passive decision support systems to more active decision making systems. If we're going to do that successfully, we're going to have to develop our problem solving tools and move from a more "technology supplier" organization to a consulting organization. The areas of AI, decision support, data bases, user interfaces, natural language are all going to be part of your tool box.

* * *

QUESTION: What is DEC's commitment to R1?

ANSWER: DEC is committed to using R1. That does not necessarily imply that R1 is used throughout DEC. DEC is like any other corporation, you can enforce only so much from the top down. What the plants do is decided locally.

QUESTION: How difficult is it to maintain a system like R1?

ANSWER: Well, there is a basic problem of unwillingness to maintain anything that is more difficult than writing on a piece of paper. The question is: can systems be built which are easier to communicate with than it is to keep the information to yourself? This is the accessibility criterion I mentioned earlier. Is it as easy for me to access a computer as it is to tell my friend sitting next to me, who is also doing the same task as I am? In the R1 case it is not easier to communicate with R1 than it is to your buddy next to you. Therefore, you need management commitment and the commitment of the engineers to make sure the changes are made. Perhaps future research in AI will increase with accessibility of systems and the maintenance of knowledge. Today, there is a lot of work that goes into maintaining these systems. What we are saying is that some of these techniques provide you with capabilities that you did not have before, but it does not solve all problems.

QUESTION: Could we talk a bit about Expert Systems, what they are, how they work, etc.?

ANSWER: I probably annoy some people in AI when I say this, but Expert Systems is an example of Artificial Intelligence people patting themselves on the back. Why do I say that? We all build Expert Systems, they do something at an expert level in some domain. So why do some Artificial Intelligence people call their work Expert Systems? Well, one reason is that we are building systems that do provide functionality at an expert level. The other interpretation, which I think is more valid, is that an Expert System is a technology where expertise is acquired from the user and used in a manner similar to how humans use it. R1 is an example of an Expert System because it is able to take the knowledge humans use to configure computer systems, represent them as rules in a computer, and manipulate them in what is called the production system manner. ISIS is not an Expert System. Its methods of reasoning are dissimilar to the human scheduler. The most common form of an Expert System is what is called a rule-based system. Knowledge is represented as "if-then" sentences. If X and Y are true *then* I will do Z. The number of rules to configure the DEC system in R1 is currently around two-thousand. The architecture of a rule-based system is composed of working memory, which represents the current state of problem solving and rule memory. The system executes a recognize-act cycle in which it determines which rules conditions are true, then selects a single rule and executes its "then" part. The cycle is then repeated. Working memory represents the current state of the problem solving. For example, in R1, if I need to add another box of memory and the memory bus is full, then I have to go and add to the order another cabinet (I do not know what the other cabinet is, but I need another cabinet) and I have to add an extender for the bus and a power supply. This rule states that when there is no more room in the computer, it must be extended with a cabinet, power supply, and bus extender. There are other rules that then say, well, if I do not have to add anything else to this bus then I do not need that much power. I just need enough power to supply this particular memory box. It does not matter where I place the box. I can place it three feet away due to some constraints on that particular piece of memory or the interconnection. One can represent all your knowledge about problem solving using these if/then rules.

QUESTION: Do you have to have priority requirements or successor requirements to know where to place components?

ANSWER: That is right. If there is some priority ordering, that is, there are some orderings on the bus that are legal or are not legal, there have to be rules that incorporate that knowledge.

QUESTION: It makes a difference on how you use the data, the data has to be incorporated in a certain way?

ANSWER: What Expert Systems do is provide you with a theory of: (1) how to represent knowledge as if/then rules; and (2) how to perform problem solving with those rules. It is a problem solving architecture. You provide the knowledge. That is the key difference.

QUESTION: How do rule-based systems relate to decision trees?

ANSWER: A decision tree is a static structure. Rules conditions define its applicability. Each execution of a rule-based system may result in a different pattern of rule executions.

QUESTION: Can industry use these ideas?

ANSWER: Most corporations are going to be burned over the next five years. You are going to pay a heavy price in terms of failures in the first five years before you really succeed. There are a number of reasons: one is that AI is not an engineering discipline; there does not exist a textbook on how to build Expert Systems. You can learn how to build Expert Systems only by doing, you learn by apprenticeship. So a lot of corporations are going to spend their first number of years just trying to build Expert Systems and end up failing because they do not know enough about it. Now, the obvious solution is to hire AI people. There are not that many out there. The number of AI people who are graduating in any year at the Ph.D. level is probably on the order of fifteen to twenty. Master's level is probably about the same because there are probably fewer Master's degrees in AI than there are Ph.D. degrees.

QUESTION: How can we build AI expertise?

ANSWER: Some companies have successfully started AI groups, but it has taken time. For example, Digital Equipment Corporation entered AI in 1978 and has now built up a sizable group, that is, they may have fifty or sixty people who they call AI related people versus six thousand people who are MIS people. But having fifty or sixty people in AI is a lot more than 99.9999% of the other corporations and they did that by working with a university; funding research in areas of mutual interest and working closely with the university to transfer technology. Other approaches include consulting groups, hiring AI people, but there are not that many people out there to hire.

The other half of the coin is that, "Gee, now I have all these people doing AI, where is my production level software that has to run day-in and day-out in the organization to do that task?" There is a big void out there. There is no AI production level software. There is a lot of software you can do research with and they are fun for building toys.

QUESTION: How large is the AI machine market?

ANSWER: I really do not know. It is hard for me to judge because nobody has said, "Everybody who wants to buy an AI machine, please stand up." But for some reason there are corporations that are putting together AI machines.

QUESTION: How much training is required to learn AI?

ANSWER: You can take somebody and teach them production systems in a week. So figure that before they really understand it, it will take them three, four, five months because you have to experience using it in writing programs. For some people it may not be that long, for other people it may take infinite time.

QUESTION: Where can I learn about AI?

ANSWER: There are two books that have been written, one by a person named Patrick Winston at MIT call *Artificial Intelligence* written in 1978 I believe. There is another book by Nilsson called *Principles of Artificial Intelligence* that came out in 1980 and is by Tioga Press. They give different views of what Artificial Intelligence is and you cannot read one without reading the other to really get the multiple perspectives.

QUESTION: In the factory of the future where is the greatest payoff in developing Artificial Intelligence systems?

ANSWER: Westinghouse did an ROI on our scheduling system and it was quite large due to current utilization of resources. In general, each factory has to be analyzed separately.

QUESTION: What do you see in the future in fostering and helping information systems?

ANSWER: Well, I guess the biggest problem I have with office information systems is that I do not understand what they are. Nobody has described to me what office automation is to any reasonable depth that allows me to understand what is the purpose of office automation. I come from an environment where we have something like three or four PDP-10s and 20s and about ten VAX 7-80s and about ten VAX 7-50s, and I do not know how many PDP-11s. We have one machine that has fifty LSI-11s and it is multiprocessor and so there are levels of power that one worries about. All machines are networked and almost everyone has a terminal in his/her office and home. When I think about office automation I try to relate it to what I do. I can generate documents, I can run my spell program over it. There are even grammar programs to see whether the grammar is correct. I can print everything on a laser printer immediately from my terminal because the Xerox laser printer is hooked to Ethernet. I can even communicate to computers across the country and in Europe over the Arpanet. That is one type of office automation. It means it is a total integration of your printing facilities, your editing facilities, the access to data bases and communication with other systems within your company and across the country. This type of office automation creates as many problems as it solves and the problem that it creates is the information glut. Just because we can communicate information more does not mean that we can do anything with the information that we communicate. For example, we have this bulletin board that is system wide, every machine has the same bulletin board. Every time you put a message on the bulletin board it gets communicated to all the other systems. Every time I log-in there may be fifteen, twenty, messages on that bulletin board that I have to weed through. I usually hit the quit button even though there may be an important message there. I can get mail from anywhere across the world almost from people who want to communicate with me. If I go away for a day, I may come back and find myself with fifty messages. How do you wade through that? Just because there is a data base out there that I know has the information that I want, does not necessarily mean that I can do anything with it. So, with this information glut there is a real issue of bounded rationality. How can we utilize this information within the bounds of our rationality? We do not really know how to do that. All these office information systems and office automation do not provide a solution to this problem. They give us the capability of communicating information, but they do not give us the capability for analyzing the information. Where AI is going to play a role is in the representation and analysis of that information so what you need to know at any given time can be communicated to you.

* * *

Thank you for your attention.

Mark S. Fox
Head, Intelligent Systems Laboratory
Robotics Institute
Carnegie-Mellon University

Mark Fox heads the Intelligent Systems Laboratory of The Robotics Institute at Carnegie-Mellon University, and is an Assistant Professor of Management Science. His research interests span both computer and management science, including artificial intelligence, man-machine communication, databases, software system organization, organization theory, and operations management. He is the author of over 20 papers and technical reports.

He designed a system (The BROWSE System) that allows computer-naive users to browse through on-line databases, and has participated in both Hearsay II and III speech understanding projects.