

ON THE PLAUSIBILITY AND SCOPE  
OF EXPERT SYSTEMS IN MANAGEMENT

Vasant Dhar

July 1985

Center for Research on Information Systems  
Information Systems Area  
Graduate School of Business Administration  
New York University

Working Paper Series

CRIS #98

GBA #85-61

This paper appears in the *Journal of Management Information Systems*, Vol. 3, No. 5, 1987

**Abstract**

Over the last decade there have been several efforts at building knowledge based "expert systems" , mostly in the scientific and medical arenas. Despite the fact that almost all such systems are in their experimental stages, designers are optimistic about their eventual success. In the last few years, there have been many references to the possibility of expert systems in the management literature. However, what is lacking is a clear theoretical perspective on how various management problems differ in nature from problems in other domains, and the implications of these differences for knowledge based decision support systems for management. In this paper, I examine some of these differences, what they suggest in terms of the functionality that a computer based system must have in order to support organizational decision making, and the scope of such a system as a decision aid. The discussion is grounded in the context of a computer based system called PLANET that exhibits some of the desired functionality.

## 1. Introduction

Over the last several years, computer-based modeling systems have made it relatively easy for end users to develop powerful decision support systems in many application areas. Yet, there is a growing recognition that unless such systems are augmented with representational frameworks and inference mechanisms that take explicit cognizance of the intellectual component of managerial decision making, their utility as decision aids is limited. In parallel efforts in the field of Artificial Intelligence (AI), researchers have been concerned with similar issues, although in problem areas that would probably be regarded as more "structured" than those encountered in management. Some of the programs that have resulted from this research, commonly referred to as "expert systems", have received considerable attention because of their ability to engage in judgmental reasoning similar to that of domain experts, and exhibit comparable levels of performance.

It seems natural to ask whether similar systems might be built to support decision making in the management arena where many of the more challenging problems tend to be fairly open-ended, non-repetitive, and not amenable to analytical solutions. Answering this question requires addressing four, more fundamental questions:

1. what is the nature of expertise in domains where knowledge based support systems<sup>1</sup> have heretofore been developed,
2. what is the nature of complex managerial problems that distinguishes them from the above class of problems,
3. given these differences, what problematic aspects of management problems might knowledge based systems be used to support,
4. what system functionality and architecture are needed in order to alleviate such problems.

An answer to the first of these is based on a summary of existing literature in cognitive science and expert systems. In order to keep the discussion on the last three questions in focus, I restrict the discussion to managerial problems that require *modeling* problem situations for decision-making purposes. Specifically, I shall ground the discussion in the context of a resource planning problem for which I have attempted to

---

<sup>1</sup>I use the term "knowledge based system" to refer to a program where domain-specific knowledge plays a major role in inference. This is in contrast to programs that use "weak methods" (Newell and Simon, 1972), that is, syntactic, domain-independent methods to guide inference.

develop a "planner's assistant" (called PLANET) to help planning managers with the formulation and maintenance of planning models to support decision making. The investigation was initiated by planning managers in a large computer manufacturing company (CMC) who expressed concern over the inadequacies of existing computer based support tools and the need for a knowledge based tool to support the planning function. This effort has brought into focus some of the problematic aspects of managerial problems such as planning, sharpened the distinction between such problems and those encountered in other domains, and the implications of the differences for knowledge based support system architectures.

## 2. The Relation Between Expertise and Problem Type

The type of knowledge required to solve a problem is influenced by the degree to which the task has been formalized [37]. As a domain becomes better understood, formal theories or normative models are articulated. These provide a basis for understanding and solving problems within that domain. In the absence of this formalization, problem solving and understanding are more likely to depend on informal, intuitive, possibly unarticulated models.

In this section, I consider the nature of problem solving in domains that lie at three different points of this "structuredness" spectrum: highly formalized domains where clearly identifiable bodies of knowledge exist, less structured domains where expertise is more implicit but nevertheless identifiable, and unstructured problems where the knowledge brought to bear in solving problems, is evolutionary and often "distributed" across several individuals. The last of these is characteristic of managerial planning, where information that is used to construct models for decision-making, is continually changing.

### 2.1. Expertise in Structured Problem Domains

There have been many psychological studies of human problem solving mostly in problem domains that would generally be considered "well structured". Broadly speaking, the problems studied have either involved "common sense" reasoning pertaining to everyday physical phenomena [16, 24, 8, 20, 12],<sup>2</sup> or specialized knowledge from highly formalized domains such as physics or algebra [22, 38, 30, 6, 5].

---

<sup>2</sup>Sometimes referred to as "naive physics".

Although humans are generally competent with naive physics problems, competence in solving real physics or other scientific problems is less common. Larkin [22] explains this phenomenon as follows:

"...the process of mentally simulating events so as to predict their outcome, a facility possessed by most people for common contexts, is extended and refined in a skilled scientist to become a sharp and crucial intuition that can be used in solving difficult, complex or extraordinary problems. Novices, lacking this extended intuition, find such problems difficult" (Larkin, 1983, p.75).

Several studies of problem solving in these areas have contrasted expert and novice behavior in order to understand the nature of this extended intuition. A common finding has been that the quality and speed of solution is influenced by the *nature of the representation* adopted. Experts appear to possess the functional equivalent of a large set of perceptual patterns and an "indexing scheme" that enables them to perceive the important features of a problem. If the problem is not exceptionally difficult, they often work "forward" without trial and error (i.e. without the need for backtracking) from general principles toward results that "include" the solution. Chi. et.al [6] explain this in terms of the ability of the expert to rapidly categorize the problem into an appropriate "principle-oriented" schema. Once correctly classified, axiomatic knowledge can be used to solve the problem in a primarily top-down manner.

Many studies of human problem solving behavior have involved the design of simulation programs. Several of these programs have been used for theory development and validation in domains such as statics [30], dynamics [26], and electronics [4]. Evidence gained from observations of human problem solving is typically used to judge the validity of these computational models. An understanding and measurement of the "quality" of expertise is facilitated considerably because of the existence of a stable, clearly identifiable body of knowledge in the form of theoretical principles or normative models. Not surprisingly, expertise in these areas appears to be highly correlated with individuals' abilities to recognize and apply the appropriate physical principles involved.

The major usefulness of computer based systems as *support* tools in these domains appears to be as intelligent tutoring systems that can take cognizance of students' naive concepts about scientific domains, and facilitate the transfer of a principled body of knowledge to novices. Several experimental systems along these lines have been built for symbolic integration (Kimball, 1983), electronic troubleshooting [4], axiomatically based mathematics [39], probability theory [1], and a consultative system for MACSYMA

[17].

## 2.2. Expertise in Expert Systems

Expert systems research has been influenced by a growing recognition that high performance programs are not likely to emerge through the clever use of a few powerful domain-independent techniques, but through a systematic formalization and use of large amounts of domain-specific knowledge. The implications of this shift toward a "knowledge based" approach are well summarized by Goldstein and Papert [18]:

"The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a fashion that permits their effective use and interaction. The current view is that the problem solver (whether man or machine) must know explicitly how to use its knowledge -- with general techniques supplemented by domain-specific pragmatic know-how. Thus we see AI as having shifted from a *power based* strategy for achieving intelligence to a *knowledge based* approach" [18].

Most AI research in expert systems has involved development of large knowledge based systems in problem areas where consultative decision support is a practical necessity for solving difficult problems. Major efforts have been in medicine [31, 36, 41],<sup>3</sup> geological exploration [15, 10,] mass spectroscopy interpretation [23], and computer layout [25]. In contrast to physics-like domains, these areas are less well understood. Because of this, it is much harder to measure expertise against a formal, axiomatized body of knowledge. Rather, expertise tends to be *implicit*, manifested by consistently high performance with difficult problems. These problems typically involve uncertain, ambiguous, and fragmentary data. An expert must therefore judge the reliability of facts in order to clarify the problem, and acquire additional evidence in such a way so as to discriminate among competing conceptualizations of a situation. In affect, "noisy" data coupled with an inherently large search space requires the use of intelligent heuristics, typically refined through experience, in order to impose pragmatic constraints on complex, open-ended problems.

A major reason for the impressive performance levels of expert systems has been the extensive efforts by

---

<sup>3</sup>CASNET [41] specializes in glaucoma assessment and therapy, MYCIN [36] in antimicrobial therapy, whereas CADUCEUS [31] deals with the whole of internal medicine.

system designers at formalizing this mostly experiential, often subjective knowledge extracted systematically through experts.<sup>4</sup> In fact, an important benefit of this knowledge extraction exercise is the systematization of previously unrecorded or unexpressed knowledge. Some researchers consider the primary contribution of constructing expert systems in such domains as being one of theory formation, thereby moving such problems into the category of "structured" problems.

### 2.3. Expertise in Managerial Problems

In the types of problems discussed above, the expertise involved is typically *individual*. For managerial problems however, it is useful to distinguish among problems where individual expertise or normative models are involved, and *organizational* level problems involving inputs from multiple individuals.

Many attempts at developing models of expertise for administrative problems have focused on the individual. Such models, some of which are embodied in computer based systems, have been designed in domains such as loan assessment and trust management [7], portfolio management [9], financial diagnosis [3], capital budgeting [2], and welfare eligibility [40].

In contrast to individual problem solving, organizational level problems introduce several types of complexity into the modeling process. These complexities are well chronicled in articles describing the early attempts at building large corporate simulation models. In these efforts, detailed mathematical models of organizations were constructed complex problems where closed form solutions were infeasible. [19,29,34,35]. A major motivation for developing such systems was they made it possible to evaluate the impacts of alternative policies, opportunities, and external events (all operationalized as parameters of the simulation model) at the level of the firm. The major knowledge inputs into such models consisted of assumptions about the organization and its external conditions, obtained from multiple sources in the organization. These were then translated into detailed mathematical models for decision making. The essential features underlying this type of modeling activity are summarized as follows:

1. **Model Formulation as Assumption Synthesis:** formulating models is an inherently underconstrained exercise involving generation of alternatives for various parts of the task

---

<sup>4</sup>By the same token, a continuing problem with such systems has been that their carefully crafted knowledge bases tend to be extremely fragile -- system behavior often changes in unforeseen and undesirable ways when knowledge is added.

environment and the making of choices from among them. Since these choices are often tentative, they can be viewed as *assumptions* or *premises* on which expectations and projections are based. Any quantitative model must be understood to be conditioned on one such set of symbolic assumptions. Model formulation as assumption synthesis is discussed more formally in section 3.2.

2. **Distributed Expertise:** formulating models for decision-making involves many individuals from different levels and functional areas of an organization. There are seldom individual experts for broad-based organizational modeling; instead, knowledge about the alternatives in various parts of the task environment is contributed by several individuals. At higher levels, policy issues shape top level decisions. These provide the context for lower level strategies and decisions which can be expressed in terms of an algebraic/mathematical model. The form and implications of distributed expertise are discussed more formally in section 3.1.
3. **The evolutionary nature of models:** decisions are not "one shot" affairs. This contrasts with problem solving in expert systems and instructional systems in structured problem domains where solutions are typically "one-shot", that is, the decision maker obtains case data, engages in a consultative dialogue (with colleagues or a system), and a solution is obtained. Rather, in an ongoing enterprise, decisions are made in a context established by previous choices. New information is evaluated in light of existing assumptions and expectations. In some cases, the new information may be assimilated cleanly into the existing conceptual framework, perhaps resolving certain ambiguities or uncertainties in the prior assessment. In many cases, however, the new information can be accommodated only if prior assumptions are appropriately modified, perhaps leading to radical restructuring of all or part of the situation model. Mechanisms for managing evolutionary models are described in section 3.3.

It should be noted that our use of the term "distributed expertise" is qualitatively different from expertise in other domains in that it is neither anchored by a stable body of knowledge as in physics, nor based on consistent virtuoso performance in some area such as medicine. Rather, it is a consequence of the necessary diffusion of responsibility across multiple departments or individuals in an organization.

The discussion so far is summarized in table 1 which draws out the essential features among the problem types in terms of five key features. In the following subsection, we discuss the implications of these differences for knowledge based decision support for management.

### 2.3.1. The Role and Scope of Knowledge Based Support

Unfortunately, a fundamental problem with large scale organizational modeling is that the richness of the modeling activity -- the problem solving involved in formulating the algebraic model itself -- is not preserved systematically. The formulation or synthesis activity is in fact the most challenging and creative part of modeling exercise that shapes the structure of the mathematical model. Yet, if the

<b>PROBLEM FEATURES</b> DOMAIN TYPES	<b>BASIS OF KNOWLEDGE</b>	<b>APPLICABILITY RANGE OF MODEL</b>	<b>STABILITY OF MODEL</b>	<b>USE OF EXPERTISE</b>	<b>PROBLEM SOLVING ORIENTATION</b>
<b>STRUCTURED PROBLEM DOMAINS</b>	- Theory of the domain	Domain-specific	Highly Stable	"Classification oriented" problem solving	Cross-sectional (one-shot) problem solving
<b>EXPERT SYSTEMS PROBLEM DOMAINS</b>	- Consistently high performance on tasks - Cross validation by other experts	Domain-specific	Stable	Mostly "classification oriented" problem solving	Mostly cross-sectional (one-shot) problem solving
<b>MANAGERIAL PROBLEM DOMAINS</b>	- Multiple "partial" experts	Organization-specific	Unstable	Mostly "synthesis or formulation oriented", -- major challenge is in designing the model itself	Mostly evolutionary problem solving in historical context

Table 1

relationships between a large algebraic model and the symbolic assumptions underlying it are not faithfully preserved, the interpretation of the model becomes difficult, and modification of such models in light of a changing reality can be time consuming, ad hoc, and error prone.

Their pragmatic problems notwithstanding, the basic objectives of such modeling efforts were reasonable and are still worth pursuing. If we recognize that it is actually the formulation/reformulation of the model based on changing assumptions that is most problematic for a manager and his support staff, it is in this activity where knowledge based support is *most needed*. Conceptually, this can be achieved by representing the ancillary symbolic knowledge *about* models, which includes knowledge about the assumptions underlying the various model components. With such knowledge, the system can become an active partner in reasoning about changes to the model instead of burdening the user with the complete responsibility of maintaining and exploring models.

From a design standpoint, what is needed are structures and mechanisms for representing and manipulating the qualitative data that forms the basis for the quantitative model. This emphasis on *design* of the quantitative model from fragmentary qualitative data, (as opposed to the *selection* from a pre-defined set of models) requires a computer-based architecture that is capable of representing knowledge that lies outside the scope of current day modeling systems. In the following section, I describe such an architecture that has been shaped by the concerns articulated above. I limit the discussion to synthesis and maintenance of quantitative models only; it is assumed that if such a model is maintained, an algebraic model corresponding to it can be formulated.

### **3. Knowledge Based Decision Support for Planning**

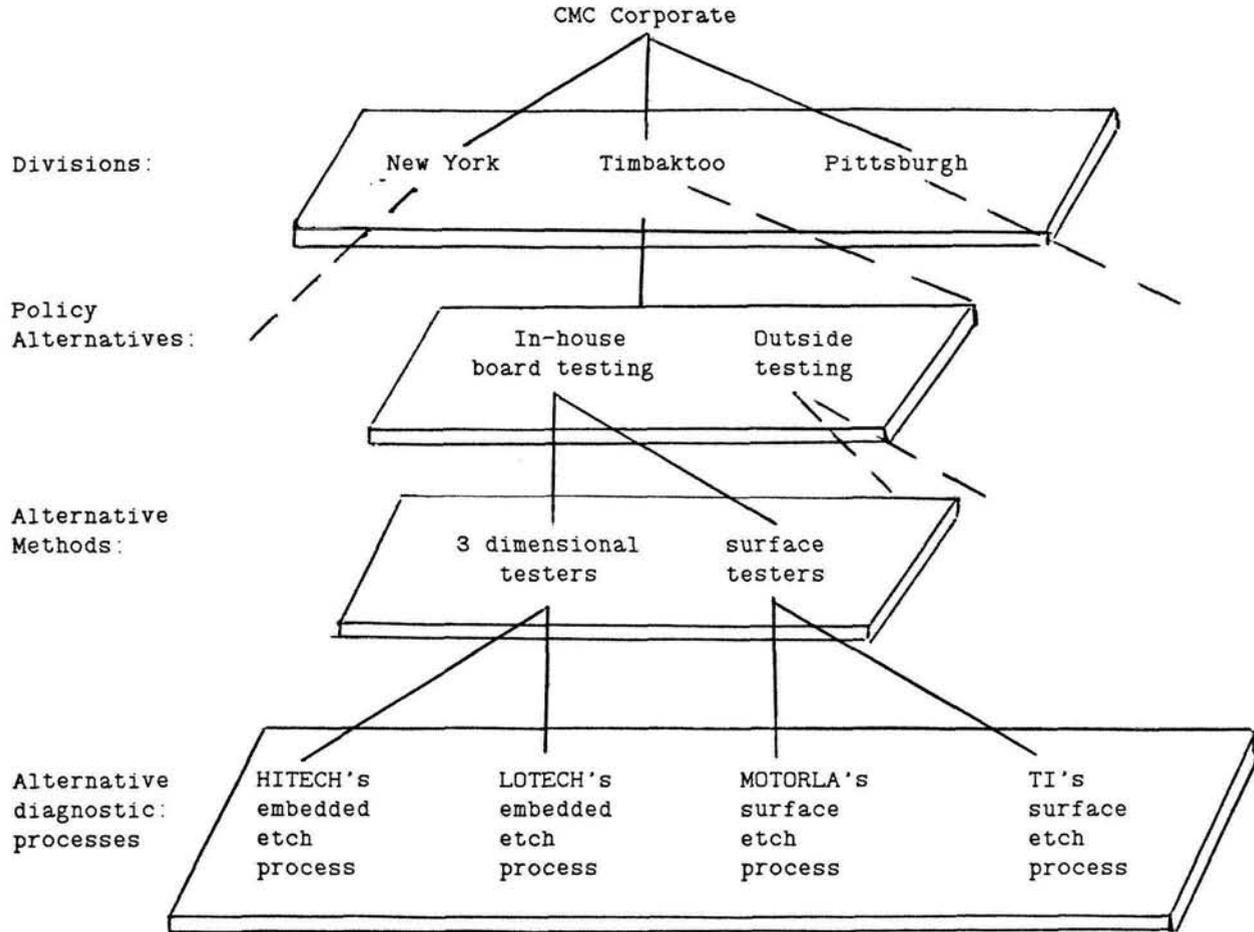
Planning is an important activity in most large organizations. Considerable time and effort of individuals from different parts of the organizations can go into building and maintaining models for planning. Several types of qualitative knowledge are involved in developing such models. However, most current day modeling systems do not adequately represent such knowledge, thereby placing a heavy burden on the decision maker to maintain the correspondence between the knowledge that can be represented within the system and that which cannot. In this section, I describe a system designed to

represent and manipulate the diversity of knowledge involved in problems such as planning. With this functionality, the system can play a more complete role in supporting decision makers.

### 3.1. Knowledge About Alternatives/Assumptions -- Distributed Expertise

Conceptually, an existing model can be viewed as being the end result of a process involving consideration of a range of alternatives (assumptions) from various parts of the task environment. These alternatives may pertain to decisions at various levels of abstraction. For example, in the CMC manufacturing environment, these assumptions pertain to computer technology to be used in the product and the processes to be employed in manufacturing it. Figure 1 shows a small set of alternatives about technology and testing processes that might be considered in such a context.

In PLANET, knowledge about these different parts of the task environment has been partitioned across a "society of agents" designed to represent standard areas of the planning activity or individual specialists in the different functional areas of the organization who have responsibility in the planning process. These specialists are represented as "objects" in HOUSE [32], a Franz Lisp object oriented programming environment that is similar in spirit to the FLAVORS package [27]. The objects correspond to the real world entities in the domain under consideration. Referring to figure 1, each of the alternatives corresponds to an object that contains knowledge about a local part of the task environment. Responsibilities of an object (which corresponds to a domain specialist) include responding to decisions being taken in other parts of the manufacturing environment and *communicating* its decisions so that other specialists may also make appropriate adjustments to their parts of the task environment. These "adjustments" are carried using "action oriented knowledge" which we describe shortly. Other, book-keeping oriented responsibilities of a specialist include keeping track of its current choice (with respect to whatever decision(s) for which it is responsible), reasons for it, and possible alternatives to the existing choice. The implementation details of this are described in Dhar [13].



ellipsis

Figure 1

A small set of alternatives considered in the course of formulating a plan in CMC's timbaktoo division. Choices at the lower end of a line indicate alternative ways of accomplishing those indicated at the top end of the line.

### 3.2. Assumption Synthesis as State Space Search

There are two sources of "action oriented" knowledge that are important in assumption synthesis. First, the problem domain itself provides constraints that reflect certain relationships among different parts of the task environment that must be realized. For example, in the computer manufacturing environment, two such domain-specific constraints (which we illustrate via an example shortly) are:

1. "A decision to employ *embedded etch* board technology rules out using test processes designed for *surface etch* technology"<sup>5</sup>

2. "Using Hitech's etch process requires using Hitech's heatsink technology too"

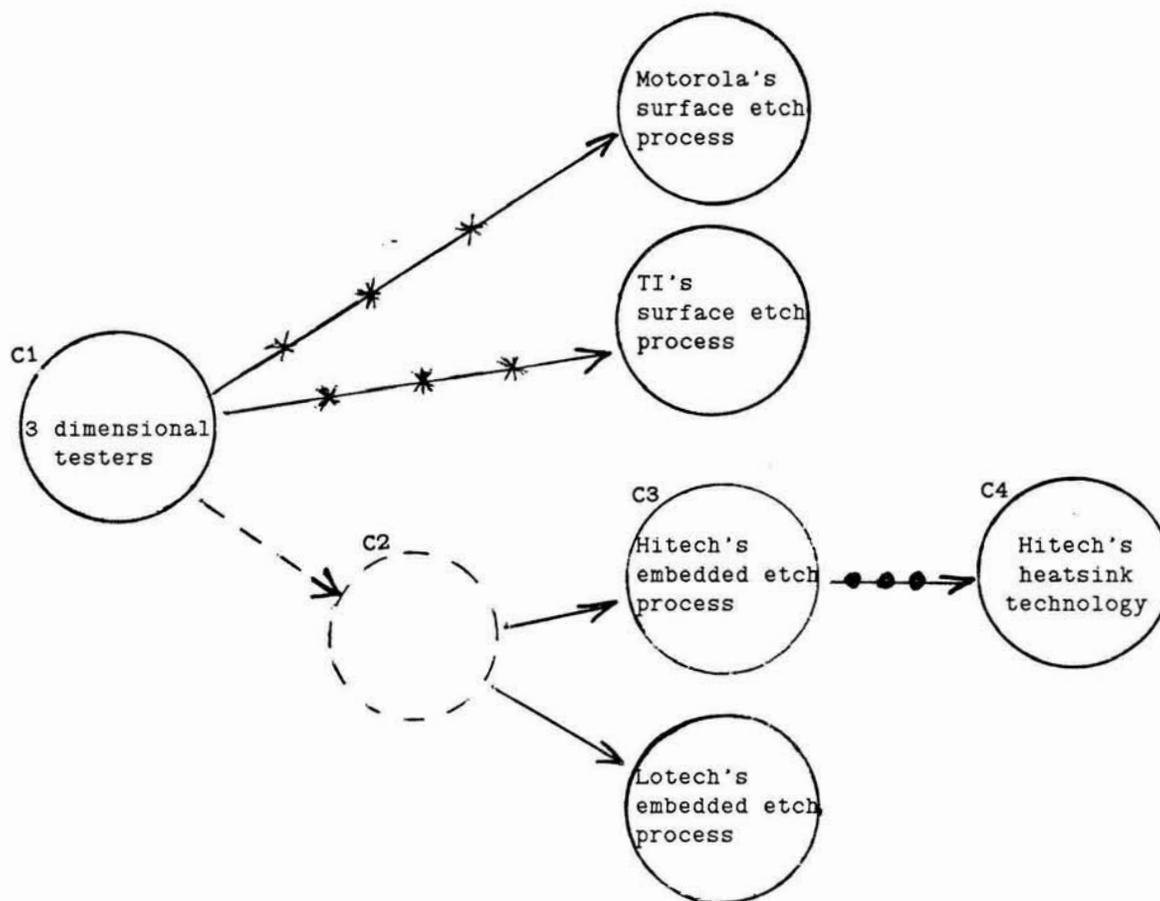
Both these constraints are indicated in the search space shown in figure 2. As long as such constraints are applicable, the problem solver is in a "constrained mode." Thus, a choice on what technology to use would rule out certain testing processes. This could in turn trigger other similar rules, setting off a chain of choices. As long as there are such choices to be made -- either due to a constraint or because there is only a single alternative with respect to some decision -- the program is in a "constrained mode."

There is also a second, quite different way by which choices are made. This is when all possible ramifications of a choice have been propagated and the problem is not yet fully solved, leaving the program in a "quiescent" state. In such situations, a "forced choice" is necessary in order to continue with the formulation process. This is a characteristic of problems that are inherently *underconstrained*, that is, the constraint relationships alone are not sufficient to make choices in all the required parts of the task environment. This requires the program to focus on some area of the task, and evaluate the set of alternatives available there. PLANET assesses the desirability of available alternatives on the basis of how they contribute toward the goals and objectives of the organization. This is operationalized as a pairwise comparison of alternatives on an "objectives vector" consisting of resources such as capital, space, and labor.<sup>6</sup> The choice is determined on the basis of the resources required by the alternatives in

---

<sup>5</sup>Embedded etch boards technology refers to boards where signals travel through the *body* of the board as opposed to its surface only. For a computer manufacturing company, the decision to use such a technology is a strategic one and has important ramifications for decisions in related parts of the task environment.

<sup>6</sup>Because the program must also sometimes compare high level alternatives for which detailed resource tradeoffs are impossible to assess before the details about these alternatives have been specified, "macro level" knowledge is used in such situations. Basically, this heuristic knowledge consists of high-level associations about how the various alternatives typically compare across the various resources.



## LEGEND:

- A -\*\*\*-> B: choice of A rules out B  
 A ----> B: B is a possible choice  
 A -•••-> B: choice of A implies B  
 ---->⊖: intermediate choices

Figure 2

A small section of a search space indicating a sequence of choices. In this space, a terminal node would represent a fully formulated plan incorporating a trajectory of choices indicated by the nodes C1, C2, C3 & C4.

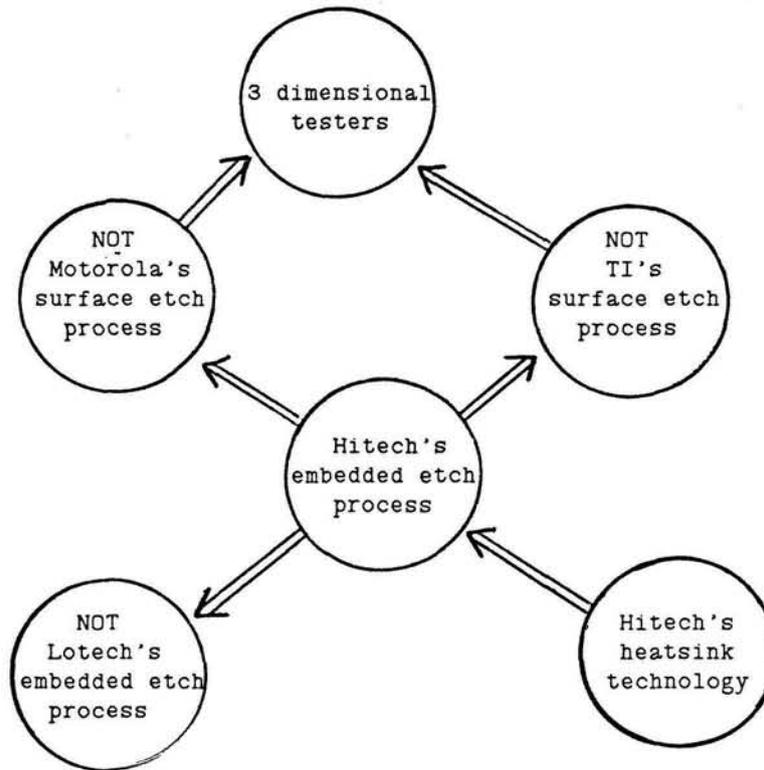
light of the organization's resource availability picture. A "choice point" involving such a comparison is shown in figure 2. The process of successively making decisions pertaining to various aspects of the task as shown in figure 2 can be viewed as a *state space search*. States toward the right represent choices being made in increasingly complete plans.

To summarize, two types of "action oriented knowledge" are brought to bear in assumption synthesis. First domain-specific constraint relationships among different parts of the task environment must be taken into account. Once the choices resulting from these constraints have been exhausted, it is necessary to make forced choices. This requires the program to focus on a critical part of the task, and make a choice based on a heuristic evaluation function that compares alternatives based on their resource requirements and the existing availability of resources. This can in turn lead to further choices based on constraint relationships. This cycle continues until selections have been made from all parts of the task environment.

### 3.3. Preserved Process Knowledge

The formulation process described above can be viewed as the result of a trajectory of choices in a state space, with the terminal nodes, if generated, representing "complete plans" from which algebraic models can be derived. This includes choices made by the program in its constrained mode, and the forced choices where alternatives are compared across the vector of objectives. Since some of these choices may have the effect of influencing others, the complete plan consists of "clusters of dependencies" in the state-space. One such cluster is shown in figure 3. Comparing figures 2 and 3, we can see that a choice is not necessarily dependent on all chronologically earlier decisions, but only on those that directly or indirectly led to it.

This "dependency information" can play an important role in the incremental modification of a large plan. Referring to figure 3, we can see that if the choice "3 dimensional testers" is retracted, choices dependent on it and all *their* dependents if any, need to be undone. Revised choices in the affected areas are made from the available alternatives. In this example, retracting the embedded etch boards decision would also bring into contention, previously eliminated alternatives pertaining to the surface etch



LEGEND:

A  $\longleftrightarrow$  B: choice of B depends on choice of A

Figure 3

A dependency network corresponding to the choices indicated in figure 2

processes. The revised choice for test processes would then be made among the previously passed over alternatives (indicated in figure 2), plus others that might have become available since a choice was last made in that part of the plan. For plans containing thousands of choices, this process of *incremental* plan evolution can serve an important attention focusing role by highlighting only the affected areas of a plan, and suggesting revised choices in these areas.

Maintaining the state-space associated with an existing plan can also be useful for carrying out qualitative "what if" analyses of choices. For example, a query of the form "what if I use the surface etch board technology" boils down to undoing the dependencies of the existing assumption (namely, the embedded etch technology), making revised choices for these parts of the task environment, and generating the resource requirements for the hypothesized scenario. This elevates the what-if analysis from the level of a quantitative model to one allowing for perturbations of the symbolic assumptions underlying such a model.

More generally, this functionality is a statement about the *role* of a *user* in such man-machine interactions. Most DSS literature uses the ambiguous term "judgement" to account for the gap between the symbolic reasoning process of a decision maker and the outputs from a model underlying a system. Unfortunately, this view of decision support does not address issues about whether it is *reasonable* to expect the user to make all the right "adjustments" in translating qualitative reasoning into a form expressible for the quantitative model. In contrast, elevating the system functionality to a level where the symbolic real-world assumptions can be manipulated relieves the user from making possibly unrealistic transitions between the two levels.

### **3.4. Summary of the Main Points**

It is worth summarizing the discussion so far in light of the four questions raised at the beginning of this paper, in particular, the last three.

A fundamental characteristic of much of managerial problem solving is that the symbolic knowledge about a problem domain is distributed and evolutionary, and must be maintained. A major problem facing planning managers is one of orchestrating the synthesis of assumptions into a coherent model, and

because of the instability of such assumptions, one of maintaining the integrity of this model over time. My approach to decision support emphasize the design or maintenance of a qualitative model on which a quantitative model can be based. In contrast, most DSS approaches have viewed support via the selection from a predesigned set of models. Similarly, the design emphasis here also contrasts with most knowledge based systems to date which have been concerned mainly with *classification* problems that involve mapping "facts" to "conclusions", *given* a stable model of the domain. In contrast, I have argued that it is the formulation and maintenance of the model of the domain itself that is a particularly problematic reality in organizations that knowledge based systems can support.

From a functionality standpoint, modeling systems or "DSS generators" form one component of such a support system. They are appropriate for representing algebraic models and performing parametric explorations within a given algebraic model structure. However, much of the problematic aspects of managerial decision making involve "getting the model right", an exercise that must make use of symbolic knowledge not expressible within modeling systems. For a system to be sensitive to the context of the decision making process, it must be able to explicitly maintain and reason in terms of this *process* knowledge, and tie the outputs of this process with a modeling system. Basically, this requires a level of intelligence over and above the knowledge expressed in an algebraic modeling system.

In order for a system to maintain the context surrounding its models, a decision support system must therefore maintain knowledge about alternatives, general domain-specific constraints, resource availability information, and dependency among prior decisions. These constitute the qualitative knowledge components required in order to synthesize and maintain evolving models. Equipped with this functionality, knowledge based systems can play an important role in facilitating an incremental evolution of models, and provide a continuity perspective that is crucial to managerial decision making, but lacking in the support provided by current day systems.

## 4. Summary

Much of the power of the PLANET architecture derives from its ability to collect, preserve, and manipulate a store of domain specific knowledge in order to reason about a problem situation. This knowledge must be provided to the system by the user.

However, an important part of a manager's job is to *create* the alternatives and recognize their interrelationship. Reitman [33] suggests that the process of generating good moves or actions, particularly in the game playing context, is similar in spirit to heuristic search. While this approach may be reasonable for domains where the entire set of alternatives, however large, can be generated a priori (i.e. the search space has a definite size), it is of little value in a managerial planning situation where actions are not defined a priori, but continually generated or "recognized". In fact, an important function of a *human* support staff is one of *creating* a set of "good" actions to be examined by a decision maker [33]. The PLANET formalism is limited from this standpoint in that it is a *reactive* support tool; the inputs that enable it to modify a plan must always come from the user. The realization of good actions also must come from the user. It is probably accurate to say that these creative aspects of decision making are likely to remain outside the scope of computer based support in the near future.

In conclusion, while computer based decision support systems will continue to have certain limitations as decision aids, there is nevertheless considerable support potential above and beyond what is available with current day systems. In this paper, I have attempted to address what I consider to be important issues that must be addressed if we are to develop knowledge based systems that exhibit some of the intelligence that is associated with managerial decision-making. Specifically, I have argued that since models used to support decision-making are based on evolving knowledge, such systems must be able to represent and maintain such knowledge. Although such systems are not "expert systems" such as those in the scientific and medical arenas, they can nevertheless use knowledge about a problem situation in supporting a decision maker with the formulation and maintenance of assumption-based models relevant to the problem situation. The architecture described in this paper has been designed to support this activity.

Acknowledgements

I am grateful to Herb Simon, Michael Ginzberg, Barry D. Floyd, and Joseph Davis for helpful comments on earlier drafts of this paper.

## REFERENCES

1. Barzilay, Amos., SPIRIT: An Intelligent Tutoring System for Probability Theory, Ph.D Thesis, University of Pittsburgh, 1984.
2. Bohanek, M., Bratko, I., and Rajkovic, V., An Expert System for Decision Making, in *Processes and Tools for Decision Support*, Henk Sol (ed.), North-Holland, 1983.
3. Bouwman, M., Human Diagnostic Reasoning by Computer: An Illustration From Financial Analysis, *Management Science*, vol. 29, no. 6, June, 1983.
4. Brown, J.S., Burton, R., & de Kleer, J., Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
5. Bundy, A., & Bird, L., Using the Method of Fibres in MECCHO to Calculate Radii of Gyration, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
6. Chi, M.T.H., Feltovich, P., & Glasser, J., Categorization and Representation of Physics Problems by Experts and Novices, *Cognitive Science*, 5, 1981.
7. Clarkeson, G.P.E., A Model of the Trust Investment Process, in *Computers and Thought*, Feigenbaum and Feldman (eds.), McGraw-Hill, 1963.
8. Clement, J., Students' Preconceptions in Introductory Mechanics, *American Journal of Physics*, 50, 1982.
9. Cohen, P., & Lieberman, M., A Report on FOLIO: An Expert Assistant for Portfolio Managers, Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983.
10. Davis, R., Austin, H., Carlbom, I., Frawley, B., Pruchnik, P., Sneiderman, R., & Gilreath, A., The Dipmeter Advisor: Interpretation of Geological Signals, IJCAI, 1981.
11. Dearborn, D.C. and Simon, H.A., Selective Perception: A Note on the Departmental Identifications of Executives, *Sociometry*, volume 21, 1958.
12. de Kleer, J., and Brown, J.S., Assumptions and Ambiguities in Mechanistic Mental Models, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
13. Dhar, Vasant., PLANET: An Intelligent Decision Support System for the Formulation and Investigation of Formal Planning Models, Ph.D Thesis, University of Pittsburgh, 1984.
14. Dhar, Vasant., and Quayle, Casey., An Approach to Dependency Directed Backtracking Using Domain Specific Knowledge, Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI), 1985.
15. Duda, R.O., Gashnig, J., & Hart, P., A Computer-Based System for Mineral Exploration in *Experts Systems in the Microelectronic Age* by Michie (ed), Edinburgh Press, 1979.
16. Forbus, K., Qualitative Reasoning About Space and Motion., in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.

17. Genereseth, M. R., The Role of Plans in Intelligent Tutoring Systems, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
18. Goldstein, I., & Papert, S., Artificial Intelligence, Language, and the Study of Knowledge, *Cognitive Science*, 1, 1977.
19. Gordon, R.J., Financial Modeling on Small Systems, *IBM Systems Journal*, volume 23, May-June, 1973.
20. Kuipers, B., Modeling Spatial Knowledge, *Cognitive Science*, 2, 1978.
21. Larkin, J., Problem Representation in Physics, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
22. Larkin, J., McDermott, J., Simon, D.P, Simon, H.A., Models of Competence in Solving Physics Problems, *Cognitive Science* 1980.
23. Lindsay, R., Buchanan, B., Feigenbaum, E.A., & Lederberg, J., *Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project*, McGraw-Hill, 1980.
24. McCloskey, M., Naive Theories of Motion, in *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.
25. McDermott, J., R1: A Rule-Based Configurer of Computer Systems, *Artificial Intelligence*, vol 19, no. 1, 1982.
26. McDermott, J., & Larkin, J., in Proceedings of the 2nd Conference of the Canadian Society for Computational Studies of Intelligence, 1978.
27. Moon, David. and Weinreb, Daniel., *Lisp Machine Manual*, MIT, 1981.
28. Naylor, T., The Politics of Corporate Model Building, *Planning Review*, no. 13, January 1975.
29. Naylor, T., and Schauland, H., A Survey of Users of Corporate Planning Models, *Management Science*, May 1976.
30. Novak, G., Computer Understanding of Physics Problems Stated in Natural Language, Tech. Report NL-30, Department of Computer Science, University of Texas at Austin, 1976.
31. Pople, Harry, E., Heuristic Methods for Imposing Structure on Ill-Structured Problems: The Structuring of Medical Diagnostics, *Artificial Intelligence in Medicine*, Peter Szolovits (ed), Westview Press, Boulder, Colorado, 1982.
32. Quayle, Casey., Object Oriented Programming in Franz Lisp, Working Paper, Decision Systems Laboratory, University of Pittsburgh, 1983.
33. Reitman, Walter., Applying Artificial Intelligence to Decision Support: Where do Good Alternatives Come From? in *Decision Support Systems*, Ginzberg, Reitman and Stohr (eds), North-Holland, 1982.
34. Rosenkranz, F., An Introduction to Corporate Modeling, Unpublished Ph.D Dissertation, University of Basel, Switzerland, 1975.

35. Shannon, R., *Systems Simulation: The Art and Science*, Prentice-Hall, 1975.
36. Shortliffe, E., *MYCIN: Computer Based Medical Consultation*, American Elsevier, 1976.
37. Simon, Herbert A., *The New Science of Management Decision*, Prentice Hall, 1960.
38. Simon, H.A., & Simon, D.P., Individual Differences in Solving Physics Problems, in *Children's Thinking: What Develops?*, Siegler (ed), Erlbaum, 1978.
39. Smith, R.L., Graves, H., Blaine, L.H., & Marinov, V.G., Computer-assisted Axiomatic Mathematics: Informal Rigor, in *Compute Education* Learne and Lewis (eds), North Holland, 1975.
40. Subrahmanian, Eswaran., WELDS: A Welfare Eligibility Determination System -- An Expert System in an Administrative Context. Expert Systems in Government Symposium, Washington D.C. 1985.
41. Weiss, S., Kulikowski, C.A., Amarel, S., & Safir, A., A Model-Based Method for Computer-Aided Medical Decision Making, *Artificial Intelligence*, vol. 11, no. 1&2, 1978.