Mixing Cognitive Strategies:

A Strategy for Improved Expert System Performance

Duane Sharman

John Kendall

University of Calgary

1. A Perspective on Expert Systems

This paper discusses the development of expert systems distinguished from other types of software by their development methodology. In this context, knowledge acquisition is viewed as an alternative to "structured" development methodologies. The development of an expert system for a cellular telecommunications application is discussed to show how expert systems can be partitioned.

How do structured methods determine how a system should be partitioned? The structured development life-cycle divides the development of a system into four phases: analysis, design, implementation, and testing [Yourdon ??]. The analysis phase defines what the system has to do in detail, an activity that has as the important side-effect of defining the problem structure. Design is thus an effort of matching the structure of the solution to the problem structure as defined by the analysis.

Application of this methodology to software development often fails, primarily as a consequence of incomplete analysis, leaving structural characteristics of the problem and its solution to the designers' imagination. Structural flaws arising from poor analysis inhibit the evolution of the system towards the desired goal. "Knowledge-engineering" has been proposed as an alternative analysis and design method, of particular value when problems are ill-structured (e.g., the solution procedure is different on every problem) but are routinely solved by human experts. The knowledge-engineering life-cycle derives the structure from human experts whose expertise presumably reflects an implicit understanding of the problem's structure. Details of the knowledge-base can then be added incrementally: the system evolves.

To develop an expert system, three people are required:

- A human expert
- A representative user
- A knowledge engineer

Each of these people assumes a distinct role in the development process [Hayes-Roth et al 1983]. The human expert brings his compiled experience to the development process. The knowledge engineer should be part psychologist and part computer scientist. He must adopt methods from cognitive psychology in analyzing the expert's cognitive processes and in addition must draw on traditional and structured software development techniques when the expert-system shell needs enhancement or when sub-problems can be solved algorithmically. In exposing the expert's cognitive structure, the knowledge-engineer is determining fundamental constraints on the program's function and establishing a structure for transferring expert knowledge to the program. The user's involvement is not much different in expert system development than it is in any other development process -- his concerns relate to the program's competence, performance and the human-factors of the man-machine interface. As a consequence of incremental development, however, the user must be deeply involved in the development from an earlier stage.

In developing a knowledge-based (or expert) system, the focus of development efforts is not so much on the problem to be solved, but instead on the strategies and methods used by an expert in solving the problem. Expert systems' technology is most appropriate in cases where either 1) finding any solution is difficult, or 2) criteria for an optimal solution can be stated but no algorithm is available that can provide an exact solution within reasonable resource constraints. Expert systems are developed by evolutionary development within a psychologically valid structure, offering the following practical results:

- The user can understand the results and explanations offered by the program at all phases of its development.
- The human expert can understand how his knowledge, which was previously not consciously considered, relates to the rules and declarations assembled into the program's knowledge-base.
- The knowledge-engineer can correlate program anomalies (ie. bugs) with knowledge that is directly accessible within the program.

Researchers in cognitive psychology and computer science have established some general principles about the use and development of knowledge by humans. The consensus is that expert levels of performance are achieved by carefully trading-off generality and power. The problem-solving strategies with the widest scope tend to be weaker (i.e., they provide little control over the activities of the problem-solver). Conversely, better results can be attained within a narrowly bounded domain. Expert performance is possibly only when the balance is found. The expert-system described in this paper helps diagnose operational faults in cellular telecommunications networks on the basis of data given in problem reports from customers and maintenance personnel. Although the specific results reported are valid only within a narrowly-defined domain, the methodology and techniques used are applicable in other domains.

2. Organization of the Paper

This paper covers many issues relating to the development of an expert system for diagnosing operational faults in a cellular mobile telephone system. The focus is on the methodological issues that establish knowledge-engineering as an alternative to the structured analysis and design life-cycle.

The major sections that follow are:

- Analysis of the benefits to be derived through the application of expert systems in cellular telecommunications systems.
- Discussion of related AI work.
- Discussion of structural characteristics of knowledge-based systems.
- Analysis of a verbal problem-solving protocol from an expert.
- Description of a prototype knowledge-based program.
- Review of methods and results.

3. Automation of Maintenance Activities

Before addressing the various theoretical and practical issues confronting expert systems' developers we will consider the problems faced by maintenance personnel in a cellular telephone operation, and review the factors motivating the use of expert system technology.

First consider the functions performed by the switching system in a cellular telephone system. In contrast to early mobile telephone systems cellular mobile telephones are highly automated and provide a full range of services compatible with standard telephone exchanges. With this automation has come greater efficiency, allowing reuse of a limited bandwidth allotment within a geographic area. These benefits are achieved at the expense of greater system complexity, an increase that is most visible to the maintenance personnel.

Cellular telecommunications systems are a synthesis of existing technologies. Thus although it is difficult for an operating company to recruit enough cellular telecommunications specialists, technical personnel are available with experience that is relevant to specific subsystems within a cellular system. Three major subsystems are:

- Radio base-station equipment, which provides a medium for voice and data communications between the cellular system and the mobile terminals.
- Telephone voice switching systems, to tie the cellular radio base-station equipment into the international telephone network.
- Data communications networks for remote control of base-station and switching equipment.

The situation confronting maintenance personnel in a cellular operation is thus one of over-specialization: most RF technicians, for example, will know little about voice switching systems, and telephone technicians will generally not understand much about base-station equipment. To respond promptly to customer problem reports often requires an understanding of all three subsystems. Expert systems technology can help interpret customer reports, drawing on automatically collected data such as alarm logs and billing records.

How important is prompt response to customer problem reports? Consider the competitive environment faced by a cellular operating company. Throughout North America legislation provides for two competing services in each geographic area. One license is given to the telephone company operating in the area; the second license is available to a competing company. Standardization is enforced, so that consumers are not locked-in to one company. Operating companies can differentiate their service from their competitions by offering either better voice quality, price, reliability, or more features. Prompt response to customer problem reports contributes to better voice quality and reliability: both are highly visible characteristics.

In summary, an expert system for interpreting customer problem reports in a cellular telephone system would improve the competitiveness of an operating company by facilitating prompt response to customer complaints, and by improving the voice transmission quality and service reliability. Such a system can help technical personnel understand processes in the system that are outside their field of expertise. It must make extensive use of data that is available on-line, through the automated facilities of the switching system.

4. Related Work

A program of this sort must be able to explain the mechanism responsible for the observed symptoms and substantiate its conclusions by refering to behavioral descriptions for the system's components. Descriptions of a system's behavior allow for reasoning based on the semantics of the domain in constrast to the strictly heuristic reasoning that is common in expert systems. In recent literature, this semantic knowledge has been referred to as "deep knowledge", in reference to the deeper understanding of system behavior that experts use to support and extend their heuristic knowledge.

Although the use of the words "deep" and "shallow" is new, some of the earliest AI programs exploited forms of deep knowledge. Gelernter's geometry theorem-prover used a "diagram" of a geometric construction (represented as a list of coordinates). Semantic checking performed with this diagram helped prune the syntactic search for a proof. For a rational reconstruction of this system and a PROLOG theorem-prover that incorporates similar ideas, see [Bundy 1983]. The geometry theorem-prover illustrates one point that this paper deals with, namely that effective use of a domain-model requires reasoning in more than one problem-space.

The problem we have considered in our research is the interpretation of customer complaints in a cellular mobile telephone system. The output desired from such a program is a causal explanation of the events within the system, consistent with a set of temporally related observations. Structured-selection inference engines (eg. EMYCIN [van Melle]) are not applicable for several reasons. These systems are good at fitting data into one of a set of models, but their techniques for knowledge representation and inference are not effective for reasoning about relationships. The three-level representation used by CASNET (discussed in [Barr and Feigenbaum 1982]) (disease/physiological states/symptoms) provides a good example for initial guidance in diagnostic problems.

The types of problems we have encountered require representation of temporal relationships. Representation techniques for managing temporal relationships have been developed for non-linear planning systems. Early linear planners (eg. GPS [Newell and Simon 1972] and STRIPS [Nilsson 1980]) only dealt with sequential states, whereas subsequent efforts (e.g., NOAH [Sacerdoti 1977], DEVISER [Vere 1985]) have allowed partially-ordered sets of states. In discussing the heuristics used by GPS [Newell and Simon 1972], the utility of "...the use of an auxiliary problem in a different problem space..." is mentioned. The application of "envisioning" in this paper relies on the use of an auxiliary problem.

The possible applications for temporal logic [McDermott 1982] [Moszkowski 1983] have been investigated. So far these systems have not achieved the balance between expressiveness and computational complexity that would enable them to be employed in conjunction with other heuristic and model-based inference techniques. The formalism presented in [Allen 84] shows more promise in this regard, and has influenced our work.

In a previous paper [Sharman 1985] some of the issues involved in diagnosing operational faults were addressed, and a PROLOG program that employed a hierarchical generate-and-test strategy was presented. The performance of this program was acceptable only on a narrow range of problems, however; when presented with a wide range of problems the program's time and space usage deteriorated rapidly. The generate-and-test strategy, though simple to program in PROLOG, was insufficient. Acceptable performance over the range of test-cases used in initial testing was achieved through careful tuning, rather than the application of sound knowledge-engineering principles.

Another line of development relevant to this paper is the work on qualitative reasoning by Kuipers, Forbus, and de Kleer [Bobrow 1985]. The emphasis in this work has been the inference of a qualitative description in domains characterized by continuous variables. Such reasoning is needed to apply symbolic reasoning techniques in domains that require extensive reasoning about positions and magnitudes.

5. Problem Structure and Solution Strategies

The first step in knowledge-engineering an expert system is to determine the structure of an expert's problem-solving process. Knowledge-engineering derives structural constraints by analyzing the cognitive processes of a human expert: the expert has arrived at his structure through experience and can demonstrate effective solution methods. Structural aspects of his knowledge will be reflected in the strategies he uses to address problems.

Uncovering these strategies is the key to the knowledge-engineering activity. A good match between a program's problem-solving strategies and the human expert's cognitive strategies ensures that subsequent knowledge and software activities engineering take place in an appropriate environment. A good match allows explicit representation and inference about constraints of the problem domain, and establishes a tight correspondence between syntactic and semantic constraints. Many researchers have already emphasized the significance of these points [Lenat 1982] [Winston 1984].

The most popular expert system shells offer a limited selection of problem-solving strategies (eg. EMYCIN [van Melle] and PROSPECTOR [McCammon 1983] [Duda et al]. In [Newell and Simon 1972], problem-solving strategies are defined to be "...processes for deciding what to do next." (pg 190). Strategies are non-deterministic methods; real problem-solving behavior methods usually involve the coordinated application of different strategies. A sampling of the strategies recognized in human problem-solving are:

[1] Generate-and-test

[2] Means-Ends Analysis

[3] Planning

[4] Taxonomic Specialization

[5] Memory-Based Recognition and Matching

5.1. Various Strategies

Each of these five strategies will now be discussed in detail.

5.1.1. Generate-and-Test

Generate-and-test (GAT) is a cognitive strategy that has formed the basis for many prototype programs in artificial intelligence. GAT is an appropriate strategy for solution of problems that are presented in the "set-predicate" formulation. A problem presented in this fashion provides a set, U (or a method of generating the members of U), and a test to check if elements of U are in the goal-subset, G.

The definition of GAT given in [Newell and Simon 1972] is:

```
*********************************************************

    generate U:
        test if element in G,
            if true stop generation and
            report solution is element,
    stop and report no solution

*********************************************************
```

To illustrate, consider an anagram-solving problem where a person is given a string of letters and told to rearrange them to form a word. A generate-and-test strategy for solving such problems consists of two components: a generator that rearranges the given letters, and a tester that checks the validity of the resulting strings. The generator is usually not completely random, but instead uses some knowledge of domain constraints (eg. frequency of letters pairs) to guide its choices. GAT is often used in tasks that have little inherent structure, or when an understanding of the problem-space structure has not been developed. It can provide a good test for representational issues, but without more sophisticated control over the reasoning process its performance is often poor.

5.1.2. Means-Ends Analysis

Means-Ends Analysis (MEA), discussed extensively in [Newell and Simon 1972], has formed the basis for many programs. To apply MEA, a good understanding of the problem solution space structure is needed. The MEA strategy requires a representation of the problem that allows the "difference" between the current state and the desired (goal) state to be determined. Actions are chosen from a list of operators that are sorted according to the differences they reduce. Once an operator is selected it is checked against the current problem state to see if its constraints (preconditions) are satisfied. If not, a new sub-goal is setup to achieve the preconditions for the operator. Goals are thus generated recursively and once the operator's preconditions are met the operator is applied and a new difference is determined.

MEA uses three different types of goals: transformation (ie. finding paths through the solution space), difference reduction, and operator application goals.

5.1.3. Planning

Planning is applicable to the preparation of sequences of actions to attain a goal. The plan is constructed by applying operators that manipulate the plan and monitoring the states achieved in the simulation for which the plan is being prepared. Planning involves coordinated use of two representations in solving a problem, something that we will later see is crucial in interpreting reports in a telecommunications system.

A plan can be executed in the object space where the states characterize the components of the system. Operators in this space are actions performed by the components. In the notation commonly used [McCarthy] to express object-space operators, each operator adds and deletes assertions from a database that describes the current state of the world.

The other representation is the planning space, where states are complete plans and operators modify plans. Sacerdoti's planning program NOAH [Sacerdoti 1977] represents a plan as a precedence network, similar to the PERT charts used in project management. Knowledge-based programs called "critics" analyze partially-developed plans to identify conflicts and suggest modifications. Execution of the fully-developed plan produces a schedule of the activities needed to realize the goal for which the plan was constructed.

NOAH's precedence networks and critics capture important aspects of causal relationships in their domain. The planning subsystem uses knowledge of these relationships to determine how best to modify a plan when the results of executing it would be faulty.

5.1.4. Taxonomic Specialization

Taxonomic Specialization (TS) has played an important role in much of the research in software engineering and knowledge representation [Borgida et al 1984]. As a cognitive strategy it is concerned with the ordering of alternatives and goals in a search space. If it is combined with GAT, it helps control the order in which alternates are generated, thereby typically limiting the computational effort required. TS can be combined with MEA, for example, to control the priority given to each type of difference and operator.

5.1.5. Memory-Based Recognition and Matching

Memory-Based Recognition accounts for much truly "expert" human performance. Memory-based recognition processes are responsible for processing information from the senses and recognition of patterns and structures. Work in Natural-Language Processing has focussed on recognition strategies; see for example [Dyer 1983]. Recognition supplies answers in a one-step process, relying on an extensive history and long-term memory structures that allow efficient access to relevant data. Matching processes that are used to associate problem information and long-term memory structures supply assumptions and expectations as well as goals for information collection.

5.2. Performance of Various Strategies

The order in which these strategies have been presented suggests the order in which a person might learn to apply them; each requires more knowledge than its predecessor. A novice will often attack problems by random trial-and-error (GAT), progressing through MEA, Planning, TS, and finally developing an intuitive feel for solution methods (Memory-Based Recognition). Each strategy in the list should be expected to show better performance than earlier strategies.

de Kleer and Brown (in [Gentner and Stevens 1983]) offer the hypothesis that an expert's model is based on a deeper representation of device behavior than the novice's, and therefore presents a larger representation space. This increase in the size of the space confronting the problem solver is balanced by a larger set of explicit heuristic assumptions. These assumptions are not normally evident in problem-solving activity; the expert and novice will therefore often apparently use the same knowledge. The expert, however, is more likely to recognize the limits imposed by his assumptions than the novice.

5.3. Introspective Abilities and Explanation

Explanatory facility is often cited as the principle advantage of expert systems over programs that use purely algorithmic or statistical techniques. An expert can explain his conclusions by presenting a chain of reasoning in their defence. Expert systems usually develop their explanations from an "introspective" trace of their problem-solving engine's activities. An expert's introspective abilities also change as different strategies are used. Initially he will be poor at explaining his problem-solving activity, but as he learns to apply MEA, Planning, and TS he will develop better explanatory abilities. Memory-Based Recognition does not permit introspective explanation, since there is no conscious reasoning involved in the solution process. An expert who uses Memory-Based Recognition can still provide detailed explanations, but his explanations must be fabricated from his background knowledge, drawing on assumptions and ambiguities latent in deep knowledge.

5.4. Representation and Inference Using Deep Knowledge

In the preceeding discussion of planning the need for two problem-spaces was mentioned briefly. One problem-space in Planning is concerned with simulation of the system under analysis. This problem-space imposes a linear, totally-ordered temporal relationship on descriptions of the system state derived from a simulation of system execution. What is described by a point in this space is a snapshot of the system at a single temporal instant. The second problem-space abstracts from the temporal instants of the execution space, giving an omniscient, atemporal view of the system evolution. The former representation looks at the plan "length-wise", in contrast to the latter representation which adopts a "side-ways" viewpoint. All that is required of this "side-ways" representation is that causal (and other temporal) relationships be shown as a partial ordering of events and states.

A pragmatic goal for Deep Knowledge Expert Systems is that a single representation of system behavior be used, and that the distinction between the two problem-spaces should result from the variety of inference strategies used. This is not a logically necessary requirement, and there is no body of evidence that one can point to to support or refute the psychological validity of this goal. The necessity for using a single behavioral representation is exclusively a concern of implementation, for if an Expert System is to be maintainable, redundant representations cannot be permitted in its knowledge-base.

5.4.1. Representation of System Behavior

Results reported in [Williams et al 83], [Kuipers 84], and [Bobrow 85] demonstrate the psychological validity and computational effectiveness of autonomous objects or device models as a representational primitive. Device models provide a framework for representing a system's behavior. A system is described as an assemblage of sub-devices where the connections between devices define a topology for the system. This topology limits the interactions between devices. Behavior internal to a sub-device can be inferred through local inference, considering only interactions with other devices as allowed by the topology. In general, decomposition rules allow "nesting" of topological or structural descriptions, and only the bottom level devices must have explicit behavioral description.

The system topology is an explicit representation of causal connectivity in the system. Depending on the type of system being modelled, different connections will need to be considered. Physical models may need to consider, for example, mechanical connections as well as paths of heat conduction. In communications networks the paths of information-flow are a fundamental concern. Telephone systems have types of information transfer that need to be considered: voice and data communications. The transmission of voice is the raison-d'etre of telephone systems, but data communications (or in the jargon of the telecommunication industry, signalling messages) is needed to control the voice transmission facilities. In analyzing the operation of the electronic modules that are used within the telephone system, a high-level physical representation would probably be appropriate; here the major topological links are wires and other conduction paths. Where radio transmission of voice or data is used, yet another form of interconnection must be considered, one that describes the radio frequency propagation characteristics over a large region.

Each type of connection arises in relation to a different structural description of the system, and is associated with specific types of sub-device model.

The behavior of devices in the data-communication environment is commonly described by state machines. Device behavior is characterized by describing the reaction of a device to received messages, described in terms of changes to internal state memory and transmitted messages. Such a description is appropriate because the entities involved assume active roles, however, electronic modules and radio transmission paths are not conveniently described in these terms because of the passivity of their components. Where passive components are involved, both devices engaged in a causal interaction are significantly affected by their interaction; in these cases one must concentrate more heavily on the "process" in which they are involved, and less on the individual devices' behavior.

Other representations appropriate for expressing device behavior include symbolic constraint propagation networks, rote memory for specific sequences, add/delete lists, production rules, and Petri Nets.

5.4.2. Causal Inference

A model of a system's behavior can be developed using the above concepts, given a topological arrangement of lower-level devices, each of which is characterized by local behavioral rules. What questions could one ask about a system that is described in this manner

The most obvious query that such a model could help answer is to trace the paths of causality associated with system behavior. Continuing with the communications network examples introduced previously, one could ask what caused a telephone to start ringing, or what caused a burst of static heard by one of the parties in a telephone connection? Causal questions can also be formulated in the forward direction: for example, what effects resulted from the failure of a microwave link?

A causal account for a system's overt behavior is useful in diagnosing faulty behavior. Consider a problem reported to a telephone system operator by a subscriber. If the subscriber complains that he is unable to reach a certain number, then a forward causal trace could identify some of the parts of the worldwide telephone network that would have been used to complete the call, had it been successful. A backward trace based on his report (e.g., that he was connected to a recording of an "all-circuits-busy" message) would permit further reduction in the range of events that need to be investigated. A satisfactory analysis of the reported problem would include a causal trace for the call, expanded from the customer report to show sufficient causes and necessary effects. Although an adequate explanation for the call may be that it was routed to a recorded announcement, because of an "All-Trunks-Busy" condition at some point in the telephone network, a satisfactory analysis of the problem must account for the "All-Trunks-Busy" condition, and show that other predictable consequences have been verified. These requirements assure minimal consistency for events and processes. An additional check must be made to ensure that the sequence of actions for each devie is consistent with the behavioral model for that device.

It should be noted briefly that local device behavioral representations are not adequate for constraint satisfaction to construct a causal description. Thus a strict reductionist approach to behavioral interpretation will fail. In [de Kleer and Brown 85] constraint propagation methods are shown to be inadequate without heuristic assistance or non-local behavioral inference methods. In dealing with causality in qualitative simulations they found it necessary to use non-local inference methods or to introduce causality heuristics. These heuristics allow constraint propagation to proceed once an impasse has been reached and there is insufficient local information to determine what a device will do next. In discussing fluid flow in conduits, their "conduit heuristic" propagates an increase in flow into the conduit as an increase in the conduit pressure. Propagation of non-heuristic device and conduit constraints can then proceed to determine the effects of an increase in pressure.

Such behavior is ubiquitous; thus their heuristics are valuable in tracing out causal paths. Electrical circuits exhibit analogous behavior, and a form of this heuristic can be applied. The conduit heuristic applies whenever positive (frictional) resistance is encountered. That positive resistance or friction should necessitate the invention of heuristics should not be too surprising. Physicists have struggled with the heuristic nature of the Second Law of Thermodynamics for over a century, without finding a compelling explanation for irreversible behavior.

5.4.3. Teleological Inference

Behavioral simulation and causal inference help interpret the behavior of a system. Methods for reasoning about the function of a system's components are needed to relate the components of a system to the goals or purposes that the system serves. An understanding of function (or teleology) can be helpful in determining how additional knowledge about a device can be used. For example, knowing that a microwave link serves as a high-bandwidth multiplexed channel, a diagnostic program could invoke inference procedures similar to those used for fiber optic or digital links. A planning system could give priority to devices whose determined purpose relates to the goals to be achieved. [de Kleer 85] presents a method for determining component teleology in electronic circuits using a causal analysis. A teleological vocabulary for electronic components is matched against the causal paths involving a particular component, to decide what that component's function is. For example, a resistor may be described as a voltage-sensor or divider resistor, depending on causally how the resistor figures in circuit operation.

5.5. Using Deep Knowledge to Justify Heuristics

Expert systems such as EMYCIN, PROSPECTOR, and CASNET use a base of "shallow" heuristics (rules of thumb) to give a phenomenological treatment of symptoms and diagnoses. The knowledge engineer collects these rules by analysing human problem-solving behavior. Observation of human experts in problem-solving situations has led many knowledge engineers to conclude that a base of "deep knowledge" is tied to heuristic rules. This 'deep knowledge' provides a foundation for shallow inference method and may include the justifications for rules and models of the physical systems in the domain. A human expert can access this deep knowledge to explain his reasoning to verify the applicability of his heuristics in unusual circumstances and to reason from basic physical principles when shallow heuristics are inadequate.

An intuitive model for problem-solving using deep knowledge suggests that

reasoning strategies using deep knowledge. Some attempts to incorporate deep knowledge into expert systems have encountered difficulty integrating deep, model-based knowledge with heuristic knowledge. One possible strategy is to apply the shallow knowledge first reducing the solution space presented to the deep knowledge inference processes. This question will be raised below in reference to the problem-solving protocol.

Deep knowledge is more than simply having multiple sources of explanation or interpretation. Part of what a deep knowledge base must include should identify the assumptions inherent in the heuristic knowledge-base. In [Smith et al. 85], the heuristic knowledge of a system is related to a deep model by showing a proof of the heuristic using a system of logic that allows non-truth-preserving inferences. They use abductive inference, or reasoning backwards from believed consequences to their antecedents, in addition to default, definitional, and theoretical inference rules. The resulting proof tree is referred to as a justification structure, and is used to aid knowledge-base refinement but is not used in heuristic inference.

5.6. A Reference Model for Investigating Deep Knowledge

Knowledge acquisition requires a model of the expert's knowledge to direct the knowledge engineer's attention to important aspects of the expert's behavior. This theory supplies the basis for analysing problem-solving behavior exhibited by human experts. Two levels of theory are appropriate; a theory of human cognition is needed to resolve issues of experimental methodology and a domain-specific theory of knowledge-representation and inference is needed for data interpretation. The former theory ought to be as weak as possible; methodological claims cannot in general be experimentally confirmed or refuted without recourse to a larger body of scientific knowledge. The domain-specific theory is amenable to confirmation and contradiction within the bounds of knowledge acquisition and can therefore make stronger claims. A good interpretive theory makes knowledge acquisition more effective and efficient.

The most suitable cognitive model is the information-processing model expounded in [Newell and Simon 1972] and [Simon 1985]. Aside from claims of psychological validity, this model is attractive in that it suggests how a computer program could be written to replicate human problem-solving behavior. But probably the main reason for adopting this model is that it has formed the basis for many successful expert systems.

5.7. A Theory of Cognition

This section reviews our version of the standard information-processing model of cognition, building a framework for the interpretive theory of deep knowledge usage.

There are two memory components in an information-processing system (IPS). Short-term memory (STM or working memory) holds the knowledge that is used by active cognitive processes (inferences), and is limited in size. Information stored in working memory is not persistent. Long-term memory has much more storage capacity and is assumed to be permanent. Long-term memory (LTM) is not normally considered to include any inference processes, however we have adopted the position that storage and retrieval operations are non-trivial, and consider LTM together with the storage and retrieval processes as the system's representation system.

Information in working memory and the representation system is organized into symbol structures, consisting of symbols and relations. Symbol structures are transferred into working memory directly from sensory inputs and from the representation system by retrieval processes. Retrieval processes do not actively process their data; they simply locate symbol structures in the representation system and transfer them to working memory. Movement of symbol structures to and from the representation system is controlled by active cognitive processes.

5.8. Task Environment

A brief description of the task environment is needed at this point, to allow following sections to assume an understanding of the type of deep model that is involved.

The expert system that is being developed interprets events reported by customers in a cellular mobile telephone system. For example, a customer may report that he was disconnected during a telephone call. The function of the expert system here is to determine what the most likely sequence of events leading to the loss of the call was, and to acquire evidence from data the telephone system collects for maintenance and billing. The system is expected to describe a plausible, consistent scenario that includes the customer's report.

A cellular telephone system has three major subsystems: a radio system that provides voice and data communications, a computer-controlled telephone switch, and a data network for interconnecting physically separated radio and telephone equipment. A symbolic model of behavior is appropriate for the data network and switching systems; the components of these systems can be modelled effectively as communicating processes. The major complication with these components is that their behavior is state-dependent, thus some form of temporal inference is needed. Much of the radio system is amenable to this treatment, however, there is a wealth of quantitative spatial information related to RF propagation. Some symbolic encoding of quantitative information is needed to fully develop a deep model of the radio subsystem.

5.9. A Basic Interpretive Model

Preliminary research was conducted to develop an interpretive model. Protocol analysis of a retrospective report was used to gain insight into expert problem-0solving behavior, permitting an informal description of the domain-specific model for use in analyzing the protocols of other experts. The initial retrospective report was analyzed in a previous report [Sharman 1985].

6. Retrospective Written Report

The scenario presented to the expert was based on a fictitious cellular system. Cellular telephone systems provide conventional telephone service to mobile customers, using radio links to provide a path for voice and data transmission between a radio base station (transceiver) and a mobile telephone (terminal). When a mobile customer moves too far from the transceiver to which he is currently linked, the cellular telephone system tries to hand the call from that transceiver to one closer to him.

The problem chosen for the test case involves a possible mode of failure for a hand-off. Since data communications are required to control the transceiver from the telephone system, a failure on a communications link could cause the call to be lost unexpectedly. The expert was given design descriptions, a hypothetical customer report, and a variety of maintenance and billing summaries as might be available on-line.

The problem report he is given are as follows:

(1) Party A is on Route 1, at 6:00 A.M..

(2) Party A is moving (ie. he is driving).

(3) Party A calls Party B using his mobile telephone.

(4) Parties A and B talk for a while.

(5) The connection fades suddenly.

(6) The call is disconnected.

The expert's goal is to explain event (6): to determine the sequence of events internal to the system that caused the call to be disconnected prematurely.

A summary of the protocol reported in solving this problem is as follows:

(7) The telephone system successfully set up the call. Event (3) shows that it should have tried to set up the call. Event (4) shows that the set-up was successful.

(8) Event (6) is not a normal occurrence; the system should have tried to maintain the call.

(9) Since Party A is a mobile subscriber, successfully setting up the call (7) must have involved the cellular part of the system in assigning a radio channel to Party A.

(10) Because Party B is outside the cellular network, the Public Switched Telephone Network (PSTN) must have been involved in setting up the call (7). The PSTN must have dialled party B.

(11) From the known location of Party A (1), we know that Cell 1 was responsible for originally establishing the link to Party A.

(12) The fading noticed by Party A (5) suggests that Party A was on the outskirts of the region of good radio reception (coverage) for Cell 1.

(13) Knowing that a radio channel from Cell 1 was successfully assigned to Party A (from (9) and (11)), and that Party A was on the outskirts of Cell's coverage area (12), we can assume that the cellular telephone system tried to hand-off Party A from Cell 1 to another cell. This is reasonable, because he was moving at the time (2).

(14) To try to hand-off a call (13), the system controller would query neighbouring cells (eg. Cell 2 and Cell 3).

(15) The neighbouring Cells would, on receiving this query (14), measure the received signal strength and report it to the system controller.

(16) From these measurements (15), the system controller would select either Cell 2 or Cell 3 as the new radio link for the call.

(17) The cellular telephone system would try to hand-off Party A from Cell 1 to the selected target cell. (This requires establishing a new voice path for Party A). Communications between the system controller, the target cell, and Party A's mobile terminal would be directed to re-route the PSTN link to the new cell, and re-direct the mobile terminal to the new link.

(18) Since it is early in the morning (2) and there is not likely a heavy load on the system, the re-routing probably succeeded.

(19) The complete sequence did not succeed: failure of the events (14) or (15) would have blocked the handoff. But checking the list of alarms for that time, we do not find any reports of internal communications failures.

(20) Thus, the most plausible explanation for the loss of the call is that the mobile terminal did not receive the re-direct command.

6.1. Analysis of Cognitive Strategies

The goal in studying this report is to identify the problem-solving strategies used. The analysis proceeds by breaking the protocol into episodes, where the utterances in each episode relate problem-solving behavior that is directed at the same goal and only mundane inferences are made. The complete protocol is divided into phases, corresponding to the use of different cognitive strategies.

Five phases showing different strategies can be identified:

o Initial Description (1) - (6).

An initial description of the call sequence is collected. Nothing in this sequence suggests that it is particularly sensitive to the specifics of the problem. This phase uses memory-based recognition to recall expected actions and create information-gathering goals. Some information is collected to characterize the known actors (Party A and Party B and their equipment) and the basic events within a call (setup, holding, and disconnection).

o Identifying Spatial and Temporal Boundaries (7) - (12)

Most of this sequence is concerned with identifying the spatial and temporal extent of the call. Spatial boundaries are defined in a logical space determined by the interconnection of equipment that played a role in the call. Telephone industry jargon recognizes two (possibly identical) paths through the telephone network: a path for voice transmission and a path for 'signalling' control information between pieces of equipment that are involved in providing the voice path for a call. Point (5) suggests that the quality of the voice path was degraded; this probably provided the motivation for tracing the equipment along the voice path. It would also be reasonable to assume that this is a normal activity in interpreting a call scenario. Memory-based recognition (triggered by the knowledge that voice quality is related to the equipment involved) likely accounts for this goal.

o Hypothesis: Handoff Failure (13)

This point raises the hypothesis that the degradation of the voice path resulted from a failed handoff sequence. Handoff is an activity whose purpose is to maintain an acceptable level of voice communications quality. Teleological reasoning offers a good explanation for this hypothesis.

o Envisioning (14) - (17)

These points show the expert envisioning a correct handoff sequence. This is done to check the hypothesis that the handoff failed, and allows deep knowledge to be used to infer causal links. Phrases such as "try to ..." indicate goals about the reasons behind the system's actions. These goals are of course not part of the system's state, but are built into the system by its designers. Failed events or non-events require teleological reasoning to find a related goal. Forward simulation is seen here as a form of planning, resulting in the construction of a sequence of actions leading to a hypothesized state.

o Consideration of Candidate Solutions (18) - (20)

This sequence shows the application of MEA to generation and evaluation of fault possibilities. Different possibilities are tested sequentially, and the goal of modifying the event sequence envisioned above (to account for the hypothesized failure) is explicit in the protocol. It is also possible that GAT was used here, proposing each fallible event in the normal handoff sequence as a candidate and testing against each individually to find causal or teleological relationships between the candidate event and the disconnected state.

6.2. Deep Knowledge Cognitive Processes

An interesting aspect of the above analysis is that the fourth section (envisioning) uses a different problem-space to justify heuristically determined conclusions. Whereas preceeding inferences had developed a picture of the problem that was temporally disjoint, during envisioning, facts are reported in chronological order. The component of the problem-state that explicitly represents the actions and states of equipment also differs. The operators that are applied manipulate a composite picture of the system state. Envisioning allows the expert to introduce many constraints, both temporal and stative, that cannot be managed heuristically.

Following envisioning the envisioned sequence of events is available. Cognitive activities before and after envisioning permit consideration of constraints on states and more flexible reasoning about temporal relationships. For instance, in step (18), the approximate time of day enters into the picture: we know that all relevant events occurred during what is typically a time of low load on the telephone system. Reasoning processes may consider a series of steps, knowing only the endpoints, without considering intermediate steps. For example, assuming the failed handoff hypothesis, and knowing that the call reached a disconnected state, disconnection sequences are generated for consideration (19).

Although envisioning is restrictive regarding temporal inference, it is a powerful strategy for handling complex relationships between device states. For example, it is difficult to determine the outcome when multiple functions use the same piece of equipment. In the cellular telephone system above, a conference bridge would be required to complete a handoff operation. Other demands on the available pool of conference bridges are difficult to account for, without envisioning explicitly stepping the conference bridge through the handoff sequence.

Knowledge-states during envisioning are complex: a conjunction of state information relevant to a variety of system components must be maintained concurrently. Other envisioning cognitive activity is required to identify the equipment for envisioning, and for more directed consideration of alternatives.

Knowledge-states in non-envisioning phases permit manipulation of conjunctions of state information that are associated by relationships other than temporal concurrency. Envisioning is thus "at 90 degrees" to other reasoning; concurrent relationships between components are strongly bound, whereas non-envisioning processes deal with simple inter-component relationships and a wider variety of temporal relationships. Envisioning provides a good opportunity for analysing causal relationships, since they depend on the information flow between components. In this way, the domain model is applied as deep knowledge to confirm and add detail to the conclusions reached by earlier heuristic phases.

7. A Preliminary Computer Implementation

7.1. Character of the Implementation

A program was written in the OPS4 language [Forgy and McDermott 1977] to mimic the strategies discussed above. An example of the program's execution is presented in Appendix A. A description of the program design and an explanation of its operation follow.

7.2. Program Design

According to our model of cognition there are three parts to the system. These parts are 1) working memory, 2) the representation system, and 3) the cognitive processes. Encoding these in OPS4 and LISP is straightforward: working memory is encoded in OPS4 working memory, the representation system is realized as a simple frames-based representation system written in LISP, and the cognitive processes are implemented as OPS4 rules. The OPS4 production memory initially contains the basic man-machine interface production rules in addition to those needed to retrieve rules from the representation system. Thus, all rules need not be loaded in production memory simultaneously but are built as they are needed.

The description of the program design offered here should give some insight into the structure of the representation system and working memory, showing how the implementation maps onto the knowledge structures discussed in earlier sections. The OPS rules are divided into groups, according to the cognitive strategies evident in the previously-analyzed report.

7.2.1. The Representation System

The program's representation system is organized around conceptual frames in the task environment. Frames are composed of roles where each role describes a relation between frames or specifies rules that are relevant within the context of that frame. Much of the reasoning in the scenario was associated with RF properties, requiring representation of geographic concepts such as regions and cells. An aggregation relation holds between cells and regions, so that part of the definition of a "cell" concept describes the regions covered it. This is a one-to-many relation and is encoded in the representation system as a single role in a cell's frame, presented as follows:

```
**********************************************************

    ;
    ; Cells -- zones of radio coverage
    ;
    (frame cell1
         cell1
         (coverage (places
                 (downtown)
                 (north-hill) (properties))))

**********************************************************
```

As this example shows, "cell1" has a list of values denoting geographic regions ("downtown", "north-hill", and "properties") stored under its "coverage" slot. The inverse of the "coverage" role is the "serving-cell" role, defined as a role of the "region" concept. The "serving-cell" role can be differentiated into "primary" and "secondary" serving cells. This allows overlapping regions to be described more accurately. For example, the "downtown" location has "cell1" listed as its primary serving-cell and "cell2" as its secondary serving-cell. Role differentiation must currently be handled explicitly by role-associated inference rules, although work is underway to allow such knowledge to be encoded declaratively.

7.2.2. Working Memory Structure

Little structure is imposed on elements in working memory. Representation were more or less based on first-order predicate logic. The notation used in working memory is based on first-order predicate logic: zero-place predicates are represented as atoms, and n-ary predicates are described by lists of length n+1; where the CAR of the list is the predicate symbol (as shown below).

```
**********************************************************

         (terminal-type PartyA mobile)

**********************************************************
```

Working memory elements serve three purposes: One set to explicitly identify the progress of the currently-active strategic phases, a second set to realize various programming control mechanisms, and a third set to encode a representation of the problem. Most of the elements are of the third type. For example, an assertion such as "(terminal-type =P mobile)" should be interpreted to mean "the type of terminal used by party =P was a mobile telephone", where the symbol "=P" is a variable that can be replaced by either "Party A" or "Party B".

7.2.3. Cognitive Processes

The phases of the written protocol as described above are associated with distinct sets of OPS rules. Progress through the phases is controlled by strategic control elements in working memory that are turned on and off by rules associated with each strategy. Once all processing within a phase has been exhausted, the termination rules take effect to terminate the current phase and initiate the next phase.

An example of a rule that activates the representation system is given below. This rule is used during the "IDENTIFY-BOUNDARIES" phase. If the call setup was determined (earlier) to have been successful and a party involved in the call used a mobile telephone (terminal), then the "primary-serving-cell" for the geographic location of the party at call setup is retrieved from the representation system. OPS-4 does not permit the use of functional forms of access to working memory on the right-hand side of rules, thus an extra left-hand side term is needed to retrieve the location of the concerned party from working memory. The RHS function "frrecall" recalls information stored under a specified frame, role, and facet in the the representation system's database.

```
********************************************************

        CHECK-SERVING-CELL  ; Rule name

        (IDENTIFY-BOUNDARIES        ; Rule antecedent (LHS)
         (setup-succeeded yes)
         (terminal-type =P mobile)
         (location =P =L)
         - (=L serving-cell primary =)

         -->

         (<frrecall> =L serving-cell primary)) ; RHS

********************************************************
```

The first phase (lines 8-44 in the Appendix) acquires information about the problem in a non-descript manner. The first action taken requests information about basic call roles from the user. No inferential reasoning occurs.

The second action in the first phase classifies the problem from the information given initially. The problems that the program can diagnose are grouped according to the type of failure. This section acquires additional call-related information in an attempt to specialize the problem-related reasoning that follows. Some of the possible problem-type frames that could be used would be: "setup-failed", "Call-not-maintained", "intermittent-loss", and

The second phase (lines 47-55) identifies the equipment used in connecting the call. It is assumed here that there are few options within the switch (the possibility of a blocking switch matrix is ignored) so that only the points of connection to the switch are needed. Geographic information is used to determine the RF equipment that was used in the call.

The handoff-failure hypothesis is selected in the next phase (lines 62-110) and a script for analyzing this hypothesis is chosen. A condition necessary in using this script is that that one (or both) of the parties were connected via RF channels. To confirm the feasibility of this hypothesis additional facts must be assembled. In addition these facts establish an initial state for the next phase.

The envisioning phase follows (lines 113-119), performing a step-by-step simulation of the telephone system according to the constraints imposed by the handoff script and known facts. The handoff script is assumed to allow only one path, thus no backtracking or other control mechanisms are needed. If an action in the script cannot be applied, the script is abandoned. The model used is based on a Petri-net formalism for expressing procedure. Transitions between successive states simultaneously change the states of all system components that play a role in the action. For instance, when a message is sent from one component to another, the states of both components change simultaneously. The temporal characteristics of real-world communication actions may be introduced into the model by expanding transitions; although this is currently not done. As the simulation progresses a set of rules monitors the envisioning process, saving information about states and events that appears relevant to the hypothesis raised above. Only the grossest state information is reported during envisioning.

Following envisioning the output of the monitor rules is evaluated in considering alternative solutions. The first hypothesis evaluated is the handoff-hypothesis (lines 121-131). More information is gathered from the user when not enough inferences can be made from the available data to establish the validity of the hypothesis. The evaluation of alternative hypotheses is done purely heuristically without any benefit from the knowledge explicit in the simulation rules. A set of consistent fault-hypotheses is proposed as the result of this phase. Among the hypotheses considered are: 'hole-in-RF-coverage', 'parameter-tuning-error', 'communication-error' (at various points in the simulation), and 'all-trunks-busy'.

Finally, an explanation is generated by adding "canned" causal links from the hypothesis frame (lines 134-145). This would not be reasonable if the set of hypotheses available and the relationships to inferred system activity became large. Special-purpose rules dominate the inference process throughout this section.

8. Directions for Further Research

Avenues of research that may be pursued to achieve the goal of integrating deep-knowledge with standard expert system paradigms can probably be better understood in the light of the points raised in this paper. The steps leading to this are:

- Organize procedures (experiment design) for knowledge acquisition. The results presented in this paper permit the adoption of a stronger model for knowledge acquisition. This effort could include development of automated tools to aid encoding of acquired knowledge.
- Collect and analyze talk-aloud protocols. Using a stronger hypothesis conduct a detailed analysis of problem-solving behavior in the diagnostic domain. This requires development of an inference engine compatible with the knowledge acquisition tools, so that the results can be validated against experimentally-collected protocol data.
- Validate the expert system against a realistic base of problems.

9. Review of Methods and Results

We believe that this example permits examination of several points that are relevant to further research in this vein. Many decisions were made in reaching this stage; by re-evaluating some of them, we hope that a valuable perspective can be gained.

9.1. Knowledge-Acquisition in a Rich Task Environment

This problem exhibits a rich task environment. The subject's cognitive strategies are aimed at reducing cognitive strain by consideration of a selected subset of the global knowledge at any one time. Assessing a subject's state-of-knowledge at any time is accordingly a difficult exercise. The goal of an expert system, as outlined in the introductory paragraghs, is to mimic at a strategic level the behavior of human experts in domain-specific problem-solving. Problem-solving is thus treated as a dynamic activity. Many other expert systems have been content with trying to mimic only the final outcome of an expert's reasoning process (eg. PROSPECTOR), ignoring the "how" part of the expert's success.

In less complex task environments (eg. classification problems) the state of a subject's knowledge can be summarized simply; relationships between concepts do not excessively complicate the problem-state. This is seen in the 'flat' problem-spaces characteristic of the structure-selection expert system shells: all that is really needed is a set of hypotheses with associated certainty factors. In structured-selection problems, relationships between domain entities are not heavily problem-dependent, and inferred relationships do not figure prominently in the reasoning. Relationships between beliefs derived by the program are maintained only for purposes of explanation. In these cases knowledge acquisition can focus on the expert's conceptual framework (eg. [Shaw and Gaines] rather than on his reasoning processes.

In a task such as interpreting problem reports in a cellular telephone system the structure of the evolving knowledge-structures is important. Typically, only a small part of this structure is evident in the final results; to achieve performance at expert standards the knowledge engineer must be attentive to the expert's cognitive processes. The importance of understanding the final conceptual structure is not diminished. But we do not believe that an adequate analysis is obtained by statical analysis of domain concepts.

9.2. Deep Reasoning

Deep reasoning based on teleology reasoning and envisioning were demonstrated in the example, along with shallow heuristic strategies. Questions about the role of deep knowledge together with shallow knowledge have not been answered conclusively; the IPS model of cognition does not provide a suitable handle for dealing with such questions.

9.3. Choice of a Suitable Expert

A problem in developing an expert system in a domain such as cellular telephony is that there are no "old hands" with compiled knowledge resulting from dozens of years of field experience. The expert in this study had no experience diagnosing field problems but did have extensive design experience. This probably introduced a bias toward reasoning based on teleology and envisioning models as these are prominent during the design process. In future work other subjects will be used to avoid strong biases in the encoding effort. However the subjects should have a design background so that they will have the required knowledge, although it will not be operationalized for diagnostic purposes. Such subjects are more likely to draw on deep reasoning strategies than subjects with less theoretical understanding of system.

9.4. Difficulties with Protocol Analysis

Protocol analysis is laborous. Few generalities about problem-solving behavior can be made without intensive analysis. Unfortunately, it is difficult to amass enough data to confirm or refute a hypothesis. In rich task environments such as this protocol analysis alone may not be adequate. The use of verbal probes may be advantageous although a stronger reference model is needed for knowledge acquisition. Some degree of automation would be desirable (as in [Simon and Ericsson 1984]), particularly if the task of producing and validating a programmed realization for observed cognitive processes was addressed.

As discussed above, attention to process is necessary. Protocol analysis addresses process explicitly. Attention to process also helps the production of a computer program eg. an expert system.

10. Appendix A -- An Annotated Execution Trace

```
This is an expert system designed to diagnose operational
problems in a Cellular Telephone System

==================          ===========================

            First I need an initial description of the fault situation
Type of calling telephone? (Enter mobile or land)
*mobile
Type of telephone called? (Enter mobile or land)
*land
Was the call setup successfully?
*yes


=====================================================


            The next step is to determine the type of problem
            Some basic facts are needed to make this determination

What is the location of partyA ?
(Enter downtown, hwy-1, mcleod-tr, deerfoot-tr)
*hwy-1
Did either party hang-up intentionally?
*no

            The call must have failed because of a switching
            event or RF problem while the call was in progress

How was the initial connection quality? (Enter good or bad)
*good

            If the call quality deteriorated prior to the failure it might
            indicate a change in RF propagation characteristics

Was the connection quality degraded immediately prior to disconnection?
*yes


            Now I need to identify the equipment involved

Which cell provides radio coverage for hwy-1 ? (Enter 1 or 2)
*1

            partyA used a radio-channel


            partyB was routed through the public telephone switch




=====================================================

Before simulating the systems behavior I should
select some scripts to consider
```

If there was a predictable radio propagation problem
there should have been a handoff attempt
Was partyA in the fringe area for cell 1 ?
*yes
Was partyA moving at the time the call failed?
*yes

A handoff attempt for partyA might have caused the failure

There may have been a handoff locate sequence for partyA that succeeded

There may have been a handoff transfer sequence for partyA that failed

A handoff should have been attempted

Next I should establish the initial conditions
before simulating the telephone system according to
these scripts

Cell 1 is now serving partyA

Cell 1 is currently serving partyA

partyA is now in conversation state

partyA is currently in conversation state

partyB is now in conversation state

partyB is currently in conversation state

That establishes a reasonable initial state to
begin simulation

==

partyA moves into a region of poor RF coverage ...
The system locates a better cell for partyA ...

==

That completes the simulation
Now to evaluate and compare the candidate explanations
Were more than 90% of the circuits in cell 1 in use ?
*yes

Is there a hole in the RF coverage for cell 1 at or near hwy-1
*yes

Have there been problems tuning the handoff parameters for
cell 1 ?
*no

==

The most-likely candidate explanations are ...

It is possible that a locate request for partyA was received
from cell 1 but no better location was found

A hole in the RF coverage in cell 1 could have resulted in no better cell being
selected for partyA

Heavy usage of voice circuits in cell 1 could have blocked a handoff attempt for partyA

==

...and that concludes our diagnosis session.

YOUR BILL IS ALREADY IN THE MAIL !!!!!

end -- explicit halt

11. References

Allen, James F. 1984. Towards a General Theory of Action and Time. Artificial Intelligence 23.

Barr, Avron, and Feigenbaum, Edward A. 1981. The handbook of artificial intelligence, volume 1. Los Altos, California: William Kaufmann, Inc.

Barr, Avron, and Feigenbaum, Edward A. 1983. The handbook of artificial intelligence, volume 2. Los Altos, California: William Kaufmann, Inc.

Borgida, Alexander, Mylopoulos, John, and Wong, Harry K.T. Generalization/Specialization as a Basis for Software Specification, in Brodie et al 1984.

Brodie, Michael L., Mylopoulos, John, and Schmidt, Joachim W. (Ed.) 1984. On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages. New York, NY: Springer Verlag Inc..

Bundy, A. 1983. The computer modelling of mathematical reasoning. New York: Academic Press.

Bobrow, D. G., (ed.) 1985. Qualitative Reasoning about Physical Systems. Cambridge, Mass.: MIT Press.

de Kleer, J. 1985. How Circuits Work, in [Bobrow 85].

de Kleer, J. and Brown, J. S. 1985. A Qualitative Physics Based on Confluences, in [Bobrow 85].

Duda, R., Gaschnig, J., and Hart, P. Model design in the prospector consultant system for mineral exploration.

Duda, R. O., Hart, P.E., and Nilsson, N. J. Subjective bayesian methods for rule-based inference systems.

Dyer, Michael G. 1983. In-depth understanding: a computer model of integrated processing for narrative comprehension. Cambridge, Mass.: MIT Press.

Forgy, Charles, and McDermott, J. 1977. OPS, a domain-independent production system language. IJCAI 5.

Gentner, Dedre, and Stevens, Albert L. (Ed.) 1983. Mental Models. Hillsdale, N.J.: Lawrence Erlbaum.

Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. 1983. Building Expert Systems. Reading, Mass.: Addison-Wesley.

Kuipers, Benjamin, and Kassirer, Jerome P. 1984. Causal Reasoning in Medicine: Analysis of a Protocol. Cognitive Science 8.

Lenat, D. B. 1982. The nature of heuristics. Artificial Intelligence 19.

McCammon, R. B. 1983. Operation manual for the muPROSPECTOR consultant system. USGS Publication.

McDermott, Drew 1982. A Temporal Logic for Reasoning About Processes and Plans. Cognitive Science 6.

Moszkowski, Benjamin Charles 1983. Reasoning About Digital Circuits. Ph.D. Thesis, Stanford University.

Newell, A. and Simon, H. A. 1972. Human problem solving. Englewood Cliffs, N. J.: Prentice-Hall.

Nilsson, N. J. 1980. Principles of artificial intelligence. Palo Alto, California: Tioga Publishing Co.

Prigogine, Ilya, and Stengers, Isabelle 1984. Order out of Chaos, Man's New Dialogue with Nature. New York, NY: Bantum Books.

Sacerdoti, E. D. 1977. A structure for plans and behavior. New York: American Elsevier.

Sharman, Duane. 1985. Inference techniques for analyzing faults in communications systems. University of Calgary CPSC 670 Paper

Shaw, M. L. G., and Gaines, B. R. Knowledge engineering tools for expert systems.

Simon, H. A. 1981. The sciences of the artificial, second edition. Cambridge, Mass.: MIT Press.

Simon, Herbert A. and Ericsson, K. Anders. 1984. Protocol analysis, verbal reports as data. Cambridge, Mass.: MIT Press.

van Melle, W. J. System aids in constructing consultation programs. Ann Arbor, Michigan: UMI Research Press.

Vere, S. 1985. Temporal scope of assertions and window cutoff. IJCAI 13.

Williams, Michael D., Hollan, James D., and Stevens, Albert L. 1983. Human Reasoning About a Simple Physical System. In [Gentner and Stevens 83].

Winston, Patrick Henry. 1984. Artificial intelligence, second edition. Reading, Mass.: Addison-Wesley Publishing Company.