EXPERT SYSTEMS FOR ENGINEERING DESIGN:
PROBLEMS COMPONENTS, TECHNIQUES, AND PROTOTYPES

by

M.D. Rychener

December, 1983

DRC-05-02-83

Expert Systems for Engineering Design:

Problem Components, Techniques, & Prototypes

26 March 1984

Michael D. Rychener

Design Research Center & Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213

Table of Contents

Abstract. An expert system embodies a human expert's domain-specific knowledge and skill, acquired and refined over years of experience. A number of problems in diagnosis and engineering design can be solved by using current expert-system techniques. This paper enumerates the main components of such problems and the steps that are taken in solving them. A few prototypical artificial intelligence systems embody techniques that can be applied to engineering problems. These are surveyed, and their relevance to components of design problems is discussed. Some expert systems in design domains are summarized, with emphasis on aspects that can illustrate wider applicability of the techniques. A number of avenues of further research are evident, and the area of engineering design offers rich opportunities for advancing the state of the art in expert systems.

Acknowledgments. The author wishes to thank the following people for their encouragement, pioneering spirit, general support, and technical advice: Art Westerberg, Steve Fenves, Allen Newell, John Gaschnig, and Rene Reboh. Considerable credit goes also to the co-workers on the HEATEX, CONPHYDE, and MOVER systems: Lewis Grimes, Rene Banares-Alcantara, Jacobo Bielak, Dan Rehak, Mary Lou Maher, and 0. Sriram. Ed Szewczak developed the automobile trouble-shooting system that I have used in several examples here. This research has been supported in part by the CMU Design Research Center, with funding from the Alfred P. Sloan Foundation (grant no. 80-4-1); by the CMU Chemical Engineering, Civil Engineering, and Computer Science Departments; by the CMU Robotics Institute; and by the Westinghouse Electric Corporation, Transportation Division. Portions of this paper have been copyrighted by the Institute for Electrical and Electronics Engineers, in the Proceedings of the 1983 Trends and Applications Conference, "Automating Intelligent Behavior Applications and Frontiers[99] (May, 1983), and are reprinted with permission.

1. Background on Expert Systems

The term "expert system" refers to a computer program that is largely a collection of heuristic rules (rules of thumb) and detailed domain facts that have proven useful in solving the special problems of some technical field. Expert systems to date are an outgrowth of artificial intelligence (AI), a field that has for many years been devoted to the study of problem-solving using heuristics, to the construction of symbolic representations of knowledge about the world, to the process of communicating in natural language, and to learning from experience. Expertise is often defined to be that body of knowledge that has grown up over many years of experience with a certain class of problem. One of the hallmarks of an expert system is that it is constructed from the interaction of two very different people, one a practicing expert in some technical domain and the other well-versed in AI and, in particular, in the process of studying an expert's problem-solving processes and encoding them in a computer system. The best human expertise is the result of years, perhaps decades, of practical experience, and the best expert system is one that has profited from contact (via the knowledge engineer) with a human expert.

1.1. Expertise: definition, advantages and costs

What are the defining characteristics of an expert system? Foremost among them is excellent performance - accuracy, speed, and cost-effectiveness of information-gathering techniques. But expert systems are also typified by a collection of other properties, many of which are taken for granted in human experts:

- The ability to explain and justify answers, either on the basis of theory, or by citing relevant heuristic rules, or by appeal to past case histories;
- The closeness of reasoning procedures to those used by human experts (the system is not a mysterious black box using obscure mathematical formulas);
- The ability to deal with uncertain or incomplete information about the current problem situation;
- The ability to summarize and point out features of the problem situation that were most important in leading to an answer, related is information about which *ether* factors might still have an effect if they were to become known;
- The use of verbal or symbolic encodings for knowledge, most of which is readily communicated in natural language;
- The ability to grow gradually by adding new pieces of knowledge, usually in the context of solving an unfamiliar problem.

These qualities make the expert system more effective as a consultant and in other expert roles, since there is some way of backing up answers and of building confidence in the system's abilities. Also included are the possibilities of improving the system by conversational means, and of using the system as a tutor or trainer.

Why would someone go to the trouble to build an expert system? Inherent complexity of a problem area and scarcity of good human experts are prime motivating factors. Often building an expert system can help to systematize a body of knowledge, so that it can be widely dispersed. Some expert systems are applied in hazardous or uncomfortable surroundings, such as nuclear reactors. Retirement of key personnel can spark interest in industrial settings. An expert system is often a good means for pooling the expertise of a number of specialists, to produce a system that is more effective than any of them working alone. Fully automated systems can often use the capabilities of an expert system to avoid the need for human intervention in many of the routine day-to-day failures and emergencies.

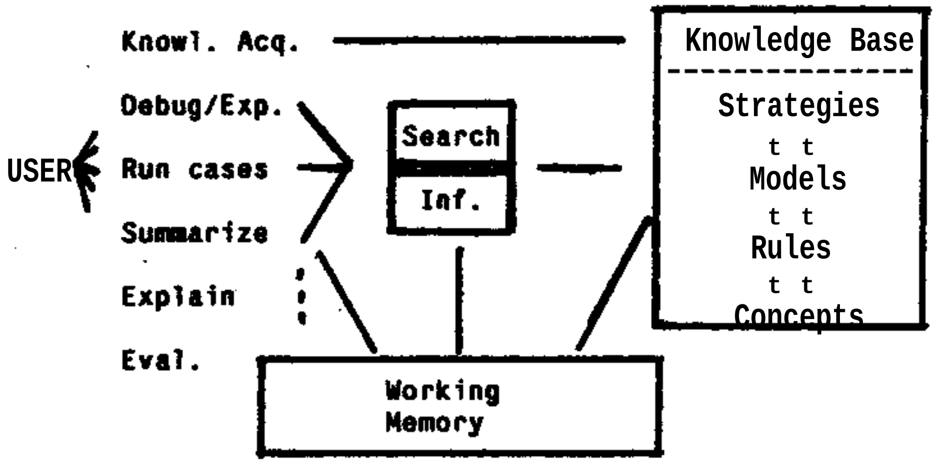
Which kinds of problems are most amenable to this type of approach? Those requiring knowledge-intensive problem solving, i.e., where years of accumulated experience produce good human performance. Such domains have complex fact structures, with large volumes of specific items of information, organized in particular ways. Often there are no known algorithms for approaching these difficulties, and the domain may be poorly formalized. Strategies for approaching problems may be diverse and depend on particular details of a problem situation. Many aspects of the situation need to be determined during problem solving, usually selected out of a much larger set of possible data readings * some may be expensive to determine, so that an expert needs to weigh carefully how serious a particular need is.

The advantages of an expert system are significant enough to justify a major effort to build them, though. Decisions can be obtained more reliably and consistently. Explanation of the final answers is an important side product A problem area can be standardized and formalized through the process of building an expert system for it An expert system may be especially useful in a consultation mode on difficult cases, where humans may overlook obscure factors. An expert system can often serve as an example of good strategy in approaching a problem, which might be useful in training situations. Expert systems can be more easily expandable than conventional software, so that they can be gradually improved as their problem domain evolves. Expert systems are often implemented in an interactive, decentralized environment, taking advantage of emerging, cost-effective personal computing resources. Ready availability of an expert consultant program can improve the training environment in industrial settings.

To summarize and re-emphasize a few points from above, to be considered a proper expert system, a system must encode knowledge from a human expert. This expert knowledge is much more than just the basic facts, but has been organized and 'compiled' through its intensive use on practical problems over a period of many years. Often, in fact, a novice cannot follow the reasoning steps of an expert, because the expert's process of organization and compilation has advanced very fan a psychological presentation of this can be found in [Anderson 83].

1.2. The components of expert systems

The key aspects that distinguish expert systems can be summarized by looking at the key components in the accompanying block diagram.



On the left side of the figure, six components are shown, reflecting the capabilities for knowledge acquisition, debugging and experimenting with the knowledge base, running test cases (perhaps systematically, from a library), generating summaries of conclusions, explaining the reasoning that led to a conclusion (or to a question by the system), ami evaluating system performance (including sensitivity of an answer to particular data items, present or absent). The main computation engine is in the center of the diagram, the searching and inferring component It searches the knowledge base for applicable knowledge, and makes inferences on the basis of current problem data. The Working Memory is a store of the current problem data, e.g., answers to questions about the problem and results of diagnostic tests.

The knowledge base is the main repository for domain-specific heuristics. It is considered to be in four levels, each one built out of elements of the next lower level:*

- Concepts - declarative representation of domain objects, with both abstract classes and concrete instances; complex interrelationships can usually be represented and used in making inferences and in constructing similarities. Usually this knowledge can be obtained from textbooks, and includes the basic terms of the problem domain.
- Rules - empirical associations linking: causes and effects; evidence and likely hypotheses to be concluded; situations and desirable actions to perform; etc. This level of knowledge is the main form that is obtained from an expert, and is based on experience. The

*Some of this terminology follows Reboti's [Reboh 81].

knowledge is empirical (difficult to obtain from textbooks), and may have associated with it "certainty factors" indicating degrees of belief in its applicability. Experts may not agree on knowledge at this level.

- Models - collections of interrelated rules, usually associated with a particular problem hypothesis or overall diagnostic conclusion. Sometimes this represents a subsystem within a complex mechanical or natural structure. Rules within models interact much more strongly with each other than with rules in other models, in a way similar to Simon's weakly decomposable systems [Simon 69].
- Strategies - rules and procedures to aid in the use of the rest of the knowledge base, e.g. guiding search and resolving conflicts when several equally plausible rules apply to a given situation.

These can be illustrated with an example from the domain of automobile trouble-shooting.*

Concepts include such things as brakes (which can be subdivided into disc and drum types), master cylinder, and brake lines. The concepts might include classification information (as in the case of brake types) and also basic knowledge about how the parts interrelate and interconnect, both in terms of physical structure (the brake line is connected to the master cylinder, etc.) and in functional terms (the master cylinder controls the pressure in the lines).

Heuristic rules are diagnostic connections between observed symptoms and probable causes, as in the following:

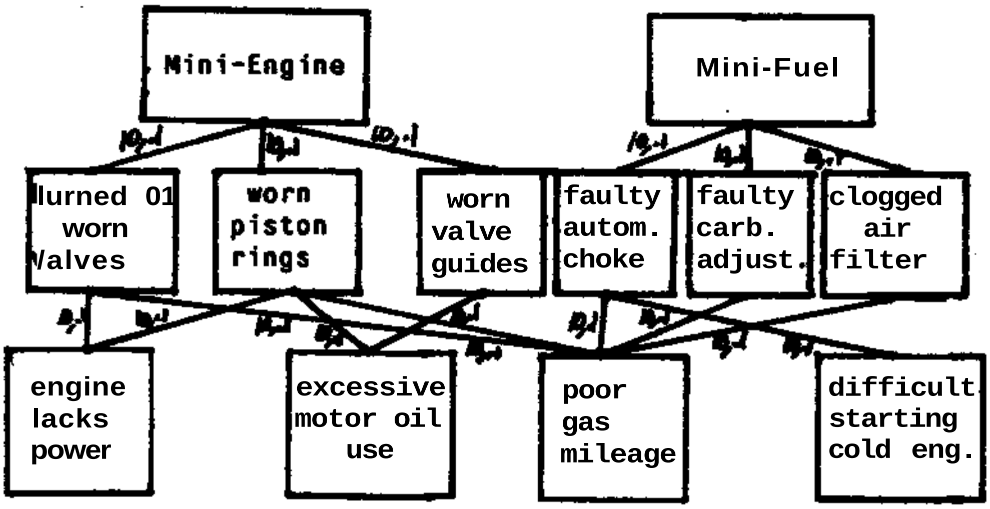
```
IF   there is excessive play in pedal,
THEN (certainty 0.2) a probable cause is low fluid level.
```

Models in this domain are subsystems of automobiles, namely things like: engine, fuel system, cooling system, brakes, transmission, and electrical system. *Strategies* include meta-rules such as:

```
IF   car performs poorly,
THEN check fuel and electrical systems first.
```

A couple of mini-models in this domain are the following:

*The author is not an expert in this domain, nor has he consulted one in connection with these examples, but the domain was chosen for its likely familiarity to many readers. The details are meant only to be illustrative, not correct as a real-world system.



These two models are arranged with observable symptoms at the bottom, probable causes in the middle, and subsystem names at the top. Each box in the diagram is a node in the network, and the arrows between the boxes represent rule connections, with associated numbers being estimates of the strength of the rule connections in both directions: one represents the likelihood of the upper box given the lower one, the other represents the likelihood of the lower box given the upper one, or alternatively the likelihood that the *absence* of the condition in the lower box would refute the hypothesis in the upper one. The observable symptoms in this case are shared between the two models. Usually in cases like this, varying strengths of the rule connections would help to discriminate between the two models when overlapping evidence is present but this example has been simplified for purposes of illustration.

An example of this mini-system running, within the PROSPECTOR / KAS system from SRI [Reboh 81], is the following:

```
•• run

This version contains the following models:

1 mini-fuel system
2 mini-engine system ____________________

Different symptoms are associated with trouble in different
models.  To help in selecting a model to pursue, it would
be useful for me to know the types of symptoms that might
be present.  Do you want to volunteer any symptoms?   n

1 — To what degree do you believe that the engine lacks
power  ? 2

2 — To what degree do you believe that there is poor gas
```

```
mileage ? 3

3 — To what degree do you believe that there is excessive
motor oil use ? 5

I have nothing more to ask about this hypothesis.
    Conclusion: my certainty in mini-engine system is now
    4.73338
--------------------
        Presently, there is only the following model:

mini-fuel system (certainty 3.053928)

        Do you want to pursue this hypothesis?  y
        OK.  (Use STAY or SWITCH commands to intervene.)
--------------------
To what degree do you believe that there is difficulty
starting when engine is cold ? -3


I have nothing more to ask about this hypothesis.
    Conclusion: my certainty in mini-fuel system is now
    2.36648
--------------------
The following models have been considered:
1) mini-engine system (certainty 4.733384)
2) mini-fuel system (certainty 2.366484)   •• Current •*

For which of the above do you wish to see additional
information?  1
--------------------
I suspect that
1) mini-engine system (certainty 4.733384)

There are several factors, in order of importance:
1) there are worn piston rings (certainty 4.714125)
2) there are burned or worn valves (certainty 3.046123)
3) there are worn valve guides (certainty 2.368421)


For which of the above do you wish to see additional
information? . . .
```

The user expresses degrees of belief in answers with numbers ranging between -5 (negative answer, strong certainty) and 5 (positive answer, strong certainty). The ordering of questions by the system is determined by *a priori* probabilities associated with the nodes, and by answers to previous questions, according to a heuristic weighting formula. Answers are combined using Bayesian methods, and values are automatically propagated among connected nodes whenever new information is obtained, so that at each step the top-level nodes are being updated according to the current information. The outcome of a session is a rating for which system is most likely to contain a fault. The user then asks for additional information and obtains ratings on which system component seems to be causing the problem.

1.3. Building an expert system

The steps to be taken to build an expert system involve building up the knowledge base from the simplest elements to the most complex, i.e., building up the concepts first, then rules, then models and strategies [Reboh 81]. During the beginning phase, a small number of test cases are used, to establish the desired system behavior on a range of typical problems. The knowledge engineer can use the test cases to build up an initial (very incomplete) set of rules and to establish the overall model organization. When these preliminary items are implemented, and the domain expert has approved, the work of filling in more and more details can begin, primarily with the acquisition of rules. Continuing progress in this second phase results in better system coverage of problems in the domain, and also can involve filling in other aspects of the knowledge base that are necessary for advanced kinds of user interaction with the system.

After the rule building phase is fairly complete, field evaluations can be carried out, and the task of the knowledge engineer shifts to adding sources of knowledge in addition to the empirical rules. For instance, in some domains it is useful to have a historical database, to help distinguish among various possible occurrences that are similar when only present diagnostic tests are used. Another source of knowledge that may be needed in some domains is background theory, a 'deeper[1] kind of knowledge of the domain. These and other sources of knowledge are not always required, and it is usually best to build up a good knowledge base of empirical rules before deciding to embark on other aspects of expertise, given that the rules are often easiest to elicit from a human expert and implement Many domains have been found to be adequately modelled by a purely empirical rule base.

Experience in building expert systems so far indicates that the first phase of the activity takes on the order of six staff-months to one staff-year. Usually a limited working prototype can be constructed and demonstrated within a few months of the start of a project, allowing managers or research sponsors to get an early idea of how the system might look when completed. The next phase can take from two to five staff-years, as the system is gradually expanded and refined to handle more and more domain problems. If further deepening phases are needed, building the additional sources of knowledge can take a similar amount of effort

2. The Components of Engineering Design

It is fruitful to examine the subproblems involved in engineering design, in order to isolate areas where expert system techniques are applicable. The main approach here is to look first at specialized areas within design disciplines, in order to apply current expert system technology. When a number of areas within a discipline have been explored in this way, we will be in a better position to integrate the results into a more comprehensive, "automated" design system. We expect the integration to pose significant challenges in the area of tool-building, and thus there is good reason to want to look at the entire problem, but it is necessary first to work on the pieces to be integrated.

Engineering design is, for the scope of this paper, restricted to problems in Chemical Engineering and Civil Engineering. In Chemical Engineering [Westerberg 82], the designed object is a chemical processing plant, and the process of design involves complex predictions and evaluations, in addition to planning and layout (flowsheeting) of a proposed process. Civil Engineering [Fenves 81, Sriram et al 82] involves a wide variety of systems and structures, including transportation systems, buildings, bridges, and dams. Civil engineers are concerned not only with design of new structures, but with their context (soil, earthquakes, environment) and with their operation and maintenance.

In order to elucidate the subproblems within design that might be amenable to expert systems, we can utilize the knowledge that the field of AI research has gained about broad types of problems that exist in real-world domains. The approach here is to enumerate and compare the components of various types of problems, when they are viewed as taking place in a problem space [Newell 80, Newell 82], i.e., when they are formulated in the way developed in heuristic-search problem solving systems. These components give rise to a classification scheme, of which three types of problems are considered in this section. A similar analysis, with more examples of types of problems, appears in [Stefik et al 82]. The section after this will use the components developed here to point out a variety of AI systems that might be applied to design problem components.

2.1. Diagnosis

The dominant paradigm of expert systems has been a diagnostic one: weighing and classifying complex patterns of evidence, to evaluate a situation that is either abnormal (as in diseases or faults) or developable in new ways (as in mineral prospecting). Diagnosis involves applying a standard set of tests, whose extensiveness or cost usually allows only a small portion of them to be performed. Thus selectivity an important aspect of a diagnostic program. Another constraint is that results may be unreliable or approximate. Expert systems have demonstrated the ability to infer possible causes of symptoms (evidence), to gather data efficiently, and to discriminate competing hypotheses.

The major components of diagnosis can be summarized as follows:

- Givens:
  - o a case of malfunctioning, unusual "symptoms";
  - o a standard set of diagnostic tests.

- Goals:
  - o to fit case into known "disease" categories;
  - o to find probable causes of symptoms;
  - o to recommend treatment methods,

- Constraints:
  - o the set of tests *are* large: selectivity is important;
  - o the tests may be very expensive (time or money);
  - o the tests may not be reliable or precise.

- Operations:
  - o infer possible causes of symptoms;
  - o gather data about symptoms and about the characteristics of the case, i.e., ask questions and do tests;
  - o classify (group) possible causes into disease (fault) hypotheses;
  - o discriminate competing hypotheses;
  - o take account of the interactions of several causes;
  - o take account of the history of the system;
  - o reason on the basis of general causal knowledge of the system, or on the basis of theory.

2.2. Design

The process of design involves some of the same constraints as diagnostic processes, in that tests may be costly, imprecise, and difficult to select But a design problem involves a different objective: to construct a system or object satisfying a given specification. Design can be broken down into several phases (see, e.g., [Fenves 81]):

1. Preliminary design, involving selection of overall forms, environment, and functional requirements (this and the next step are sometimes described as the synthesis phase);
2. Preliminary component design, where components are selected and elaborated;
3. Detailed design, in which further refinement takes place;
4. Analysis, evaluation, and optimization, to verify the various aspects of the design;
5. Documentation and detailed project planning.

Usually there are analytic tests or simulations that can be performed on a proposed design, and the components from which the construction is to be done are known and have known properties and interrelationships. Selection and connection of components are important operations in designing, as are deducing and testing properties of subsystems of the proposed result. As proposals are generated, they must be checked for consistency with the specifications. Designs undergo evolution and updating operations after being formed, and a system to aid in design must be able to track such changes and check that new variations are correctly designed and updated. New components are sometimes created, along with constraints that apply to them. A variety of guidelines and environmental regulations have to be followed. A number of these facets of design can be seen to be appropriate for the application of expert system techniques.

The main components of design can be summarized as follows:

- Givens:
  - o the specification of the desired object or system, giving its features, functions, constraints, budget, etc.;
  - o standard analytic tests on systems and components that are proposed or designed;
  - o possible components, their properties, their interrelations.
- Goal: to produce an object or system that meets the specifications.
- Constraints: (the same ones as for diagnosis)
  - o the set of tests is large: selectivity is important;
  - o the tests may be very expensive (in time or money);
  - o the tests may not be reliable or precise.
- Operations:
  - o select overall forms;
  - o select, and specify details of, components;
  - o infer properties of the desired system from the givens (to aid selection and construction);
  - o put components together into (sub)systems (this may involve planning, or searching thru alternative constructions, to meet functional specifications);
  - o check specifications (features, constraints, costs);
  - o perform analytic tests (e.g., predict behavior);
  - o evolve and update the designed system, using feedback from tests, recording reasons for decisions and inter-dependencies of details, and maintaining constraint satisfaction;
  - o create, represent and utilize new components and new constraints;
  - o observe design guidelines for efficient procedure;
  - o apply optimization procedures;
  - o consider non-economic criteria (safety, environmental protection, esthetics).

2.3. Planning

The operation of planning a sequence of actions can be viewed as a design process, in which the result has more dynamic variation than is ordinarily considered in design (Stefik, et al., have expressed similar views [Stefik 81, Stefik et al 82]). That is, planning involves the design of a structure within which complex series of actions are to take place. It must include contingencies accounting for various results of actions, and for contingencies in the environment. Restrictions may be placed on certain sequences of actions. But for the most part, planning can benefit in the same ways that design can, with respect to expert systems.

Planning can be summarized as follows:

- Givens:
  - o specifications of the results of executing a procedure (cf. designing a procedure);
  - o individual actions (corresponding to physical components).
- Goals: similar to those for design.
- Constraints:
  - o contingencies in actual procedure operation;
  - o restrictions on how procedural steps interact (e.g., inputs and outputs must match, orderings may be critical).
- Operations: similar to those for design.

3. Prototypical Artificial Intelligence Systems

This section sketches a number of existing AI systems that have some relevance to engineering design domains. In most cases, the system is not directly useable as a design tool, but it contains important ideas that could be adapted to the design domain. The ideas in these systems can be expected to be incorporated into future expert system tools. For each system, there is a list of the design problem components that it is conceptually appropriate to deal with.

3.1. The SRI Prospector / KAS system

The Prospector system [Duda&Gaschnig&Hart 79] is essentially a diagnostic system, piecing together evidence to make a conclusion about the presence or absence of one of a number of minerals. It asks the user questions that can be answered from field evidence and geological data, such as types of rocks and dgree of erosion. The evidence is processed with respect to about a dozen different models, each one organizing empirical knowledge about particular desirable deposits, e.g. copper and molybdenum. Uncertainties in information and rules are treated with a Bayesian probabilistic approach. The system can make inferences on the basis of semantic knowledge about rock types, geological forms, and other domain concepts.

The main computational and representational tools within Prospector have been separated into a separate system called KAS [Reboh 81]. It has in addition a number of facilities for entering new models, for checking their completeness, for editing existing models and for experimenting and debugging a knowledge base. The KAS system offers a very powerful environment for rapid development of applications, but the type of application suitable for it is rather narrow.

The components of diagnostic problems that KAS addresses: all except history and causality. Diagnostic abilities are important in the following design problem components:

- Component selection and specification;
- Selection of overall system design approach;

• Selection of analytic tests.

3.2. The R1 expert configurer

The R1 system [McDermott 80] does the configuration of VAX computer systems, and is currently used in production mode at Digital Equipment Corp. factories. It is used to ensure that all the required parts have been included in orders, and that when assembled they will work together as expected. The expertise consists of thousands of special-case details of the individual parts and how they interact with each other in systems. The overall configuration follows a prespecified plan, with R1 executing the plan and filling in all the details. R1 is written in a standard production-rule system language, a dialect of the OPS language [Forgy&McDermott 77]. Since it is a programming language, it is suitable for a wide range of applications, but more effort is required than is the case for a specialized system such as KAS.

The components of design problems that an R1-like approach might address:

• Put components together;
• Compare system to specifications;
• Follow specific strategic guidelines.

3.3. Sussman and Steele, Constraints

In the domain of electrical circuit design, a formulation of basic knowledge in the form of constraints has proven successful [Sussman&Steete 80]. Constraints on elements in the design are expressed symbolically and check for validity throughout the design by a method known as constraint propagation. Each circuit element is represented in a flexible way such that constraints can propagate through the element in a variety of directions. The overall result is that a model of causality within a circuit is built up by interconnecting models of the elements, in such a way that properties of the entire assembly can be predicted and manipulated by the designer. Rieger and his colleagues have developed similar approaches for other physical systems [Rieger&Grinberg 77].

The components of design problems that a symbolic constraints approach might address:

• Compare system to specifications;
• Analytic tests and predictions;
• Evolve and update proposed system.

3.4. The HEARSAY-II speech understanding system

The Blackboard model of problem solving has been developed and applied to the difficult problem of speech, in [Lesser&Erman 77]. In this model, a group of expert modules, or knowledge sources, is coordinated by way of a centralized data base of current hypotheses and other information, called the blackboard. Strategies about how the knowledge sources might best interact to produce a good hypothesis are encoded in another knowledge source, the focus of attention module. The AGE system [Nii&Aiello 79] is an example of an expert system tool that attempts to incorporate the blackboard idea, to allow several expert systems to work together on a complex problem.

The components of diagnosis problems that a blackboard approach might address:
- Infer chains of causes;
- Classify and discriminate hypotheses;
- Handle interactions.

The components of design problems that a blackboard approach might address:
- Infer properties;
- Construct systems;
- Compare and maintain constraints;
- Observe procedural guidelines;
- Maintain non-economic criteria.

3.5. Designing molecular genetics experiments

The MOLGEN system [Stefik 81] is a prototype expert system illustrating the power of a multi-level planning approach. Expertise at various degrees of abstraction is embedded into the system at different program levels. Communication between levels is done with carefully designed classes of messages, so that different levels cannot assume how things are being done in another level, but only what is being done there. This forces a very effective form of modular organization on the program. The most abstract program level is the strategy level, which knows how to manipulate tasks of various sorts; the design level knows about steps such as refinement of design operators and proposing new goals; the most detailed knowledge is at the laboratory level, where particular molecular genetics operations are encoded. The layered planning approach is combined with operations on symbolic constraints, as discussed above.

Stefik's approach has possible application to these design problem components:
- Construction;
- Evolve and update;
- Create and represent new constraints.

The conclusion section below will summarize the overall impact that the above AI systems might have on engineering design.

4. Some Expert System Experiments

Expert systems have been developed that illustrate the application of expert systems techniques to some of the main operations in the design process. These examples span a variety of domains.

Most of the systems described here make use of the KAS system, a domain-independent, knowledge-base editor for the SRI PROSPECTOR system [Reboh 81]. It was important for us and for the experts involved to be able to build up very rapidly initial prototype systems with interesting behavior. The KAS system was ideal for this task, since it requires little more than a graph-structured model of interrelated rules and logical definitions. (This contrasts with some other systems which require detailed semantic definitions of the concepts involved before proceeding with rules.) The KAS approach involves the expert in the process very soon after beginning, and showing an expert an initial demonstration is a very effective way of eliciting more details of the expertise (if in fact the system is in the right ballpark at all). The rule-based nature of the knowledge is well-known to be an advantage, and KAS extends this by having the rules structured into larger wholes called models. KAS includes a rudimentary knowledge-base editor and some debugging commands that allow on-line experimentation and revision.

Engineering design as a domain has brought out some limitations, though, which are discussed later. One class of limitation has to do with minor computational variations that can be made to fit into the existing expert-system framework. It is also apparent that AI techniques alone are not appropriate for all of the problems within this domain, and that connection must be provided to allow access to existing analytical tools of the various engineering disciplines.

The KAS system has very satisfactory knowledge base structure, inference mechanism, explanation facility and search strategy. Its knowledge base is somewhat weak in the meta-rule area, judging from some difficulties in controlling the flow of questioning (its "context" mechanism is not always effective, possibly due to bugs in our particular implementation version). The knowledge acquisition and experimentation facilities are adequate though they need improvement, since they seem to be involved with the biggest bottleneck in the process, namely with acquiring knowledge from the experts.

The need for greater generality has led us to use an OPS-based approach [Forgy&McDermott 77, McDermott 80, McDermott 82] in two of the systems discussed below. This will be touched on further in the concluding section, where we suggest that OPS should be utilized more widely as an implementation tool.

4.1. Heat exchanger networks

The first example of an expert system in engineering design is a Chemical Engineering system, HEATEX, for aiding in the construction of networks that minimize energy requirements by allowing the exchange of heat among various process streams [Grimes&Rychener&Westerberg 82]. An expert has rules of thumb for arranging the network and the heat exchangers, and the program embodies some of this expertise. This sort of analysis comes into play after the overall process has been detailed, as one of a number of evaluative criteria for estimating operating and capital costs. OPS* is used in this case, for Hs general-purpose programmability. The mode is mixed: in an initial phase of execution, the program alone produces an initial feasible configuration and evaluation of a given network. Then there is an interactive phase, where the human user can suggest evolutionary changes that might improve the overall objective of minimum energy use. HEATEX updates the network according to the user's suggestions, but it has no rules for choosing to make such evolutions itself. The resulting ensemble performs a hill-climbing search, where enthalpy constraints are used to restrict the operations. The following paraphrases a rule from the first, constructive phase of HEATEX:

```
P88: IF the goal is to do step 0 in creating a
          network,
        AND there are two streams of opposite
          types, with compatible heatioads,
     THEN create a possible match between the two
        streams, computing the mean, the
        coefficient of the match, and the delta-
        min value of the match.
```

HEATEX turned out to be a useful tool for developing the theory and methods for problems of its type, resulting in the formulation of a better algorithm and especially of evolutionary rules for improvement of networks. It grew to a system of 115 production rules. The system did not, however, execute rapidly enough to be practical for further work on the problem (much faster OPS dialects have come along in the meantime). One obvious continuation would incorporate more rules into the program to suggest where the existing evolutionary rules might be applicable, further automating the improvement of networks, but there is currently no plan to carry this out

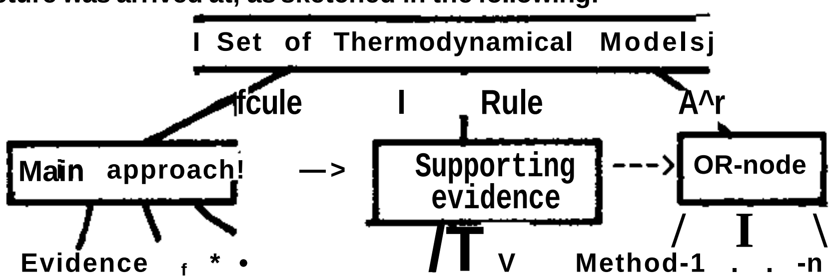
*Actually, an experimental, extinct dialect called OPS3RX.

4.2. Banares' CONPHYDE

A second Chemical Engineering system is one for selecting the appropriate analytic program to use to evaluate a design, with respect to its physical properties, by R. Banares [Banares 82]. There are apparently only a handful of experts in the world with a significant degree of expertise in this domain, an important motivation for the research. This kind of problem arises in preliminary stages of design, where the engineer is beginning to sketch a flowsheet and wants to check feasibility by looking at physical properties (eg, temperature, pressure and phase) of substances that might be involved. The expertise was first expressed in terms of informal rules such as,

```
(25) IF   operating conditions are near the
             critical region,
     THEN Peng-Robinson equation of state is
             advised, for easier convergence
             (computational reason).
```

The CONPHYDE system is composed of 5 KAS models, with a total of about 70 spaces (nodes in KAS inference graphs). Some difficulty was experienced at first in trying to organize the expertise to fit into the KAS framework, since the rules were highly overlapping and interconnected. Finally an overall structure was arrived at, as sketched in the following:



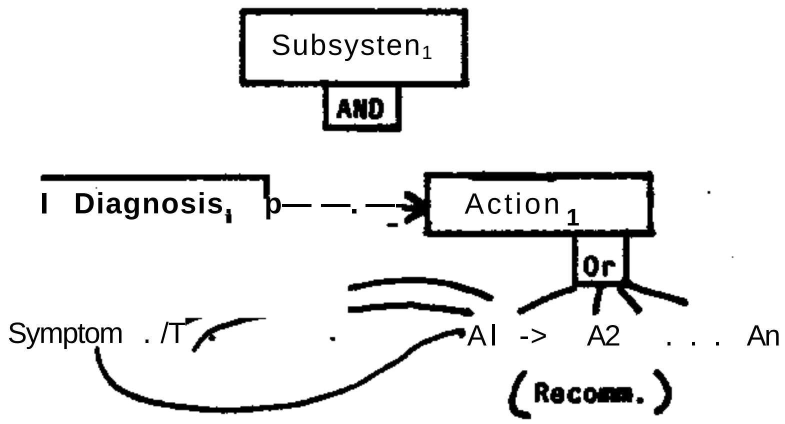
Here, the terminal nodes in the tree are actually subtrees consisting of questions for the user to answer. The overall inference direction is from left to right, and from bottom to top: from evidence to support a set of models (i.e., that determines the overall approach to be taken), to supporting evidence for that set, to questions that allow the selection of the right member of the set. Since the PROSPECTOR control structure is such that a choice is being made among models (whose top nodes are sets of possibilities), the user sees the system switching among those models rather than among particular methods. The PROSPECTOR explanation mechanism is used, after a run, to explore details of the best-rated set of models, including the method chosen and the reasons for choosing it

CONPHYDE has been successful in its domain, with respect to proving that expert system techniques are applicable in the chosen setting. Extending it will require using more advanced features of KAS, namely, accessing taxonomies of concepts, storing numerical parameters in the taxonomy, and computing values alongside the inferences (based in part on values stored in the taxonomy). We are currently deciding on whether to expand CONPHYDE itself or to switch to a new domain in which industrial support is available.

4.3. Fenves, et al, MOVER

A third expert system, involving Civil Engineering, diagnoses faults and recommends repairs in an airport transportation system [Fenves&Bielak&Rehak&Rychener&Sriram&Maher 83]. Though it is a diagnostic system, two facts suggest that some results will carry over to design domains: design itself includes some trouble-shooting, namely trouble-shooting of proposed designs (this is in the checking and analysis operations); and the issues raised in building this expert system, and any solutions proposed, are vitally relevant to design systems. These issues are discussed below. In addition, from the general analysis above of diagnosis and design, diagnostic abilities are very similar to those used in selecting an overall design approach, in selecting and specifying components, and in selecting analytic tests.

The transportation system to be diagnosed contains a number of interacting subsystems, both electronic and mechanical. The MOVER system has models for a number of those subsystems, specifically 19 models with over 400 inference nodes. The design of each subsystem network is shown in the accompanying sketch.



Each subsystem model contains a diagnostic part, including external data and confirming evidence, linked by a context link to an action portion, where conditional sequences of recommendations are represented in a network of logical interconnections. Some of the MOVER diagnostic models use probabilistic rules (with uncertainties indicating such things as frequency of failure) while others are purely logical. This system was designed and implemented over a span of about five months, with a group of six knowledge engineers (some novices).

The system as developed so far is still a prototype, as far as the real application is concerned.

Two subsystems were developed to considerable depth, to demonstrate concept validity. The KAS approach was found to be lacking in some mechanisms that would aid the system in understanding rarer and more intricate problems. In particular, database techniques are needed to handle knowledge of the history of the system, and of its intended behavior as specified by its blueprints (as opposed to empirical knowledge about how it has performed and failed). Efficiency of operation would be enhanced by giving direct access to some of the electronic components (e.g., diagnostic memories) of systems that are being diagnosed. We are exploring these issues in detail for the next phase of the project

4.4. Other experiments at CMU

Expert systems have been constructed or initiated in several other interesting areas, using techniques similar to those in the preceding subsections. These all support the usefulness of the existing techniques for building an initial approximation very rapidly, from which basis deeper explorations and interactions can occur. A small diagnostic system for trouble-shooting automobiles was built as a vehicle for comparing various expert system techniques (E. Szewczak, U. of Pittsburgh, unpublished term paper). In the public affairs area, a system of several models and over a hundred spaces has been built to perform welfare eligibility screening (E. Subrahmanian, CMU School of Urban and Public Affairs, unpublished working paper). Another Civil Engineering project involves building an expert system to aid in the preliminary structural design of buildings (Maher, M. L., CMU Civil Engineering Dept, unpublished thesis proposal). Systems in medical domains (in collaboration with 0. Servan-Schreiber, A. Rappaport, B. Mulsant, R. Nahmias and J. Chauvet) have been explored. Finally, the availability of powerful personal computers can be exploited to build an intelligent user interface to a complex design tool (A. Westerberg and M. Rychener, research proposal). The expertise in this case is both in designing chemical processes and in using effectively the design tool (which is a collection of Fortran programs).

Some other closely-related work is being carried out in electrical engineering and in decision support for industrial management A system for synthesis of digital VLSI circuits [Director&Parker&Siewiorek&Thomas 81] combines a variety of tools within a hierarchical design methodology. Its approaches to managing the databases involved and to managing the complexity of the design process, with a number of loosely interconnected modules, provide lessons for expert design systems. The set of programs described by McDermott [McDermott 82] include expert system approaches to component selection and to configuration of components into working systems, in the domain of computer hardware systems at the macroscopic level. Fox's work [Fox 81] in building tools for scheduling, process diagnosis, and other industrial management tasks provides a demonstration of approaches to a variety of design operations, including optimization, construction (planning), and evolution and updating of designs.

5. Conclusions

Expert systems to aid in engineering design have up to now illustrated approaches to the following subareas of the design process:

- selection of components;
- selection of analytic programs;
- selection of parameters for preceding;
- trouble-shooting and diagnosis of systems (and thus, presumably, of designs);
- intelligent interfaces to complex tools;
- overall strategic guidance;
- and fulfilling the functions of various levels within multi-level systems.

One general conclusion is that current expert system techniques are effective in starting to attack a wide variety of domains where expertise is known to exist and to be in demand. They are effective in providing a starting basis for ongoing projects that can make significant impacts in industrial and commercial settings. In many cases they can be completely sufficient to the task at hand, while in others, they lead to challenging research problems. Some expert system tools, e.g. KAS, are very suitable for narrow domains with diagnostic flavor, of which some components of design problems are examples. Other tools, e.g. OPS, are effective programming languages that incorporate the features of specialized expert system tools, and are suitable for a wider range of problems.

A number of limitations of current approaches have been discovered in the course of the above experiments. They can be grouped as follows:

- Limitations of overall approach, which might be alleviated by adding new modules to existing systems, and by organizing the module structure differently. Examples include:
    - The acquisition and evolution of models, a major bottleneck in the process of building an expert system. New approaches include domain-specific aids and learning (evolution) by example;
    - Help in utilizing and coordinating fuzzy interconnections and inferences (especially, maintaining consistency);
    - Use of causal and historical databases;
    - Use of sensor data directly;
    - More complete explanatory, analogical, case-history and theoretical background knowledge;
    - Adaptability to a variety of modes of operation, e.g., full automation, consultant, supervisor, assistant, and trainer;
    - Moving towards multi-level systems, involving coordination among various levels of expertise.
- Limitations in current software at behavioral level, which can be alleviated by incremental

fixes within the current framework: computing side values along with uncertainty ratings, accessing (external) databases, and interfacing with analytical programs in alien languages.

- Limitations of underlying implementation, including its extensibility, adaptability to new computational demands, internal documentation and self-description, and data structures at a useful level of abstraction; our current thinking leads us to believe that LISP is inappropriate, while a rule-based and knowledge-representation-language-based system is feasible and desirable.
- Limitations of existing architectures for flexible organization.
- Limitations of knowledge of how to incorporate background theory of the domain.

We are currently examining the possibility of developing a variety of rule-based expert system tools, to help remedy a number of these limitations. Making them rule-based will improve their adaptability, their conciseness, their understandabHity, and their portability. We hope that these tools will have some of the positive attributes of expert systems, namely the ability to explain their own reasoning and to assist in their own improvement by having an understanding of their own knowledge structure. We are also encouraged by developments in the computer hardware area. The emergence of powerful personal computer workstations will make it possible to have an expert system in a peronal computer, enabling expert systems on other computers, linked together in a local network, to cooperate on solving large design problems, using the blackboard model described above.

6. A Selective Bibliography

In the interest of acquainting a wider audience with a representative sampling of expert systems, and especially with the approaches they embody, the following selective reading list is proposed. These readings are not overly technical, and usually make their main points by way of examples of natural language dialog between a user and the expert AI system. Papers usually contain further references to related work and background material. The intended audience includes engineers (especially designers), social scientists, computer specialists without AI background, and other professionals.

The first version of this bibliography was published in June, 1981, and since that time a number of excellent expert system papers have appeared, mainly in national and international conferences on AI and in the *Artificial Intelligence Magazine* [IJCAI 81, AAAI 82, IJCAI 83, AAAI 83]. The classics still stand out, and I haven't recommended the newer papers for novices or outside reviewers of the field, though some are of high quality. For the most part, the conference papers discuss specific applications and do not discuss substantially new issues or techniques.

In the following, I have grouped papers roughly into categories, some of which contain specific systems while others are broader overviews or even general introductions to the wider field of AI. In the interest of being representative, a number of systems of approximately equal significance to those given here have been omitted. The papers here do in fact refer to these others.

6.1. Expert systems in general and overviews of AI

[Feigenbaum 77]*
[Waterman&Hayes-Roth 78]
[Erman&Lesser 78]
[Michie 79]
[Nii 80]
[Newell 81]
[Barr&Feigenbaum 82]*
[Newell 82]*
[Stefik et al 82]*
[Hayes-Roth&Waterman&Lenat 83]*
[Simon 69]*
[Boden 77]
[Winston 77]
[McCorduck 79]
[Nilsson 80]

*Highly Recommended.

6.2. Electronics and computers

[Brown&Burton 75]
[Sussman 77]
[deKleer79]
[Borning 79]
[Sussman&Steete 80]
[McDermott80]
[Grinberg 80]
[Director&Parker&Siewiorek&Thomas 81]
[Davis 82]*
[McDermott82]

6.3. Engineering design in general

[Freeman&Newell71]
[Powers 72]
[Rieger&Grinberg 77]*
[Bennett&Engelmore 79]
[Fenves&Norabhoompipat 78]
[Eastman 81]
[Fenves81]*
[Stefik81]*

6.4. Other expert systems

[Davis&Buchanan&ShortKffe 77]*
[Buchanan&Feigenbaum 78]
[Weiss&Kulikowski&Amarej&Safir 78]
[Duda&Gaschnig&Hart 79]*
[Genesereth 79]
[Nii&AieHo 79]
[Waterman&Peterson 80]
[Lindsay&Buchanan&Feigenbaum&Lederberg 80]
[Pople81]*
[Reboh81]*

* Highly rfcconviiisnGBii*

6.5. Other specific AI techniques of interest

[Lenat75]*
[Sacerdoti 75]
[Lesser&Erman 77]
[Clancey 79]
[Davis 80]
[Hayes&Ball&Reddy81]
[Sacerdoti 81]
[Stefik81]
[Teitelman&Masinter 81]

6.6. References

[AAAI82] American Association for Artificial Intelligence.
Proc. National Conference on Artificial Intelligence.
William Kaufmann, Inc., Los Altos, CA, 1982.
See pp. 278-376.

[AAAI83] American Association for Artificial Intelligence.
Proc. National Conference on Artificial Intelligence.
William Kaufmann, Inc., Los Altos, CA, 1983.

[Anderson 83] Anderson, J. R.
The Architecture of Cognition.
Harvard University Press, Cambridge, MA, 1983.
See especially Ch. 6, Procedural Learning; also a 1982 paper in Psych. Review (89:4).

[Banares82] Banares-Alcantara, R.
Development of a Consultant for Physical Property Predictions.
Master's thesis, Dept of Chemical Engineering, Carnegie-Mellon University, May, 1982.

[Barr&Feigenbaum 82]
Barr, A. and Feigenbaum, E. A., eds.
The *Handbook of Artificial Intelligence, Volume II.*
W. Kaufmann, Inc., Los Altos, CA, 1982.

*rHQniy Recommended.

References

[Bennett&Engelmore 79]
Bennett, J. S. and Engelmore, R. S.
SACON: A knowledge-based consultant for structural analysis.
In *Proc. Sixth International Joint Conference on Artificial Intelligence*, pages 47-49. Tokyo, 1979.
Advises users of a program, MARC, for analysis of physical structures

[Boden 77] Boden, M.
Artificial Intelligence and Natural Man.
Basic Books, New York, 1977.
Chapters 1, 10, 12, 15 are pertinent to expert systems

[Borning 79] Borning, A. H.
THINGLAB - A Constraint-Oriented Simulation Laboratory.
PhD thesis, Stanford University, 1979.

[Brown&Burton 75]
Brown, J. S. and Burton, R. R.
Multiple representations of knowledge for tutorial reasoning.
In Bobrow, D. G. and Collins, A., editor, *Representation and Understanding: Studies in Cognitive Science*, pages 311-349. Academic, New York, 1975.
SOPHIE electronics lab simulation

[Buchanan&Feigenbaum 78]
Buchanan, B. G. and Feigenbaum, E. A.
DENDRAL and Meta-DENDRAL: their applications dimensions.
Artificial Intelligence 11:5-24, 1978.
Chemistry (mass spectroscopy) and scientific inference

[Clancey 79] Clancey, W. J.
Tutoring rules for guiding a case method dialogue.
International Journal of Man-Machine Studies 11:25-49, 1979.
Turning expert systems into teaching / tutorial ones; also 6th IJCAI (1979), pp. 155-161

[Davis 80] Davis, R.
Meta-rules: reasoning about control.
Artificial Intelligence 15:179-222, 1980.
There is a companion paper in the same issue, pp. 223-239

[Davis 82] Davis, R.
Expert Systems: Where are we? And where do we go from here?
AI Magazine 3(2):3-22, Spring, 1982.

[Davis&Buchanan&Shortliffe 77]
Davis, R., Buchanan, B. and Shortliffe, E.
Production rules as a representation for a knowledge-based consultation program.
Artificial Intelligence 8:15-45, 1977.
Discusses MYCIN, an expert on infectious diseases and treatments; see 6th IJCAI (1979), pp. 923-925 (EMYCIN)

References

[deKleer79] deKleer.J.
Causal and Teleological Reasoning in Circuit Recognition.
PhD thesis, Massachusetts Institute of Technology, 1979.

[Director&Parker&Siewiorek&Thomas 81]
Director, S. W., Parker, A, C, Siewiorek, D. P. and Thomas, D. E.
A design methodology and computer aids for digital VLSI systems.
IEEE Trans, on Circuits and Systems To appear, July, 1981.

[Duda&Gaschnig&Hart 79]
Duda, R. O., Gaschnig, J. and Hart, P. E.
Model design in the Prospector system for mineral exploration.
In Michie, D., editor, *Expert Systems in the Micro Electronic Age,* pages 153*167. University of Edinburgh, Scotland, 1979.
See also [Waterman&Hayes-Roth 78], pp. 203-222

[Eastman 81] Eastman, C. M.
Recent developments in representation in the science of design.
In *Proc. 18th Design Automation Conference,* IEEE Computer Society and ACM, June, 1961.
Building design, geometric modelling, and integrity of design databases

[Erman&Lesser 78]
Erman, L D. and Lesser, V. R.
System engineering techniques for artificial intelligence systems.
In Hanson, A. and Riseman, E., editor, *Computer Vision Systems.* Academic, New York, 197a
General system-building ideas

[Feigenbaum 77] Feigenbaum, E. A.
The art of artificial intelligence: I. Themes and case studies of knowledge engineering.
In *Proc. Fifth International Joint Conference on Artificial Intelligence,* pages 1014-1029. Massachusetts Institute of Technology, 1977.
Overview of a number of Stanford systems and their methods

[Fenves 81] Fenves, S. J.
Computer-aided design in Civil Engineering.
Proc. of the IEEE 69(10):1240-1248, October, 1961.
This entire number was devoted to CAD.

[Fenves&Bielak&Rehak&Rychener&Sriram&Maher83]
Fenves, S., Bielak, J., Rehak, D., Rychener, M., Sriram, D. and Maher, M. L.
An expert system for diagnosis and repair of an automated transportation system.
Technical Report, Robotics Institute, Carnegie-Mellon University, January, 1983.
In preparation.

References

[Fenves&Norabhoompipat 78]
Fenves, S. J. and Norabhoompipat, T.
Potentials for artificial intelligence applications in structural engineering design and detailing.
In Latombe, J.-C, editor, *Artificial Intelligence and Pattern Recognition in Computer Aided Design,* pages 105-119. IFIP Working Conference, Grenoble, France, March, 1978.
A view from outside AI

[Forgy&McDermott 77]
Forgy, C. and McDermott, J.
OPS, a domain-independent production system language.
In *Proc. Fifth International Joint Conference on Artificial Intelligence,* pages 933-939. Cambridge, MA, 1977.
The latest versions are OPS5 and OPS83, by Forgy

[Fox 81] Fox, M.S.
The Intelligent Management System: An Overview.
Technical Report CMU-RI-TR-81-4, Carnegie-Mellon University, Robotics Institute, Pittsburgh, PA, August, 1981.

[Freeman&Newell 71]
Freeman, P. and Newell, A.
A model for functional reasoning in design.
In *Proc. Second International Joint Conference on Artificial Intelligence,* pages 621-640. London, 1971.
Somewhat general, problem-solving approach

[Genesereth 79] Genesereth, M. R.
The role of plans in automated consultation.
In *Proc. Sixth International Joint Conference on Artificial Intelligence,* pages 311-319. Tokyo, 1979.
User consultant for MACSYMA

[Grimes&Rychener&Westerberg 82]
Grimes, L E., Rychener, M. 0., and Westerberg, A. W.
The synthesis and evolution of networks of heat exchange that feature the minimum number of units.
Chemical Engineering Communications 14,1982.
Design Research Center, Carnegie-Mellon University, Tech. Rpt DRC-06-22-81, September, 1981.

[Grinberg 80] Grinberg, M. R.
A knowledge based design system for digital electronics.
In *Proceedings of the First Annual National Conference on Artificial Intelligence,* pages 283-285. AAAI, 1980.
Semi-Automatic Digital Designer (SADD)

[Hayes&Ball&Reddy 81]
Hayes, Phil, Ball, E. and Reddy, R.
Breaking the man-machine communication barrier.
Computer 14(3):19-30, March, 1981.

[Hayes-Roth&Waterman&Lenat 83]
Hayes-Roth, F., Waterman, D. A. and Lenat, D. B. (Eds.).
Building Expert Systems.
Addison-Wesley, Reading, MA, 1983.

[IJCAI 81] IJCAI-81.
Proc. Seventh International Joint Conference on Artificial Intelligence.
William Kaufmann, Inc., Los Altos, CA, 1981.
See pp. 815-912.

[IJCAI 83] IJCAI-83.
Proc. Eighth International Joint Conference on Artificial Intelligence.
William Kaufmann, Inc., Los Altos, CA, 1983.
See pp. 100-271.

[Lenat 75] Lenat, D. B.
BEINGS: Knowledge as interacting experts.
In *Proc. Fourth International Joint Conference on Artificial Intelligence*, pages 126-133. Tblisi, 1975.

[Lesser&Erman 77]
Lesser, V. R. and Erman, L. D.
A retrospective view of the HEARSAY-II architecture.
In *Proc. Fifth International Joint Conference on Artificial Intelligence*, pages 790-800. Boston, 1977.

[Lindsay&Buchanan&Feigenbaum&Lederberg 80]
Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A. and Lederberg, J.
Applications of Artificial Intelligence to Chemistry: The DENDRAL Project.
McGraw-Hill, New York, 1980.

[McCorduck 79] McCorduck, P.
Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence.
Freeman, San Francisco, 1979.
Chapter 12 is pertinent to expert systems

[McDermott 80] McDermott, J.
R1: An expert configurer.
Technical Report CMU-CS-80-119, Carnegie-Mellon University, April, 1980.
Configures VAX computer systems; OPS language

[McDermott 82] McDermott, J.
Domain knowledge and the design process.
Design Studies 3(1):31-36, 1982.
Also appeared in Proc. 18th Design Automation Conference, Nashville, TN, 1981, ACM/IEEE.

[Michie 79] Michie, D. (ed.).
Expert Systems in the Micro Electronic Age.
Edinburgh University Press, Scotland, 1979.
A recent collection of papers

References

[Newell 80] Newell, A.
Reasoning, problem solving and decision processes: the problem space as a fundamental category.
In Nickerson, R., editor, *Attention and Performance VIII.* Lawrence Erlbaum Associates, HillsdaJe, NJ, 1980.

[Newell 81] NewellirAm
How to view the computer.
In *Foundations of Computer-Aided Chemical Process Design.* Engineering Foundation, New York, 1981.

[Newell 82] Newell, A.
The knowledge level.
Artificial Intelligence 18:87-127,1982.

[Nii80] Nii, H. P.
Heuristic Programming Project 1980.
Stanford University, Department of Computer Science, 1980.
Overview of a number of expert systems projects

[Nii&Aiello 79] Nii, H. P. and Aiello, N.
AGE (Attempt to Generalize): A knowledge-based program for building knowledge-based programs.
In *Proc. Sixth International Joint Conference on Artificial Intelligence,* pages 645-655. Tokyo, 1979.

[NiissonSO] Nilsson.N.J.
Principles of Artificial Intelligence.
Tioga, Palo Alto, CA, 1980.
Chapters 1,8,9 and 10 are especially relevant to expert systems

[Pople81] Pople, H. E. Jr.
Heuristic methods for imposing structure on ill-structured problems: the structuring of medical diagnosis.
In *Artificial Intelligence in Medicine.* American Association for the Advancement of Science, 1981.
see also 5th IJCAI (1977), pp. 1030-1037

[Powers 72] Powers, G. J.
Non-numerical problem solving methods in computer-aided design.
In *IFIPS Conference on Computer-Aided Design.* Eindhoven, The Netherlands, 1972.
Discussion of AI techniques applied to design problems

[Reboh81] Reboh,R.
Knowledge Engineering Techniques and Tools in the PROSPECTOR *Environment.*
Technical Report 243, SRI International, Menlo Park, CA, June, 1981.

References

[Rieger&Grinberg 77]
Rieger, C. and Grinberg, M.
The declarative representation and procedural simulation of causality in physical mechanisms.
In *Proc. Fifth International Joint Conference on Artificial Intelligence,* pages 250*256. Massachusetts Institute of Technology, 1977.
Complex interconnected systems simulated

[Sacerdoti75] Sacerdoti, E. D.
The nonlinear nature of plans.
In *Proc. Fourth International Joint Conference on Artificial Intelligence,* pages 206-214. Tblisi, 1975.
Flexible problem-solving and planning

[Sacerdoti81] Sacerdoti, E. D.
Problem-solving tactics.
A/Magazine 2(1), 1981.
Also in Proc. Sixth IJCAI, pp. 1077-1085, Tokyo, Japan, August, 1979.

[Simon 69] Simon, H. A.
The *Sciences of the Artificial.*
MIT Press, Cambridge, MA, 1969.
Rationalizing the process of design

[Sriram et al 82] Sriram, D., Maher, M. L, Bieiak, J. and Fenves, S. J.
Expert Systems for Civil Engineering - A Survey.
Technical Report R-82-137, Carnegie-Mellon University, Dept of Civil Engineering, Pittsburgh, PA, July, 1982.

[Stefik81] Stefik,M.J.
Planning with constraints (MOLGEN: part 1); Planning and meta-planning (MOLGEN:part2).
Artificial Intelligence 16:111-170,1961.

[Stefiketal82] Steftk, M., Atkins, J., Balzer, R., Benoit, J., Birnbaum, L, Hayes-Roth, F. and Sacerdoti, E
The Organization of Expert Systems: A Prescriptive Tutorial.
Technical Report VLSI-82-1, Xerox Palo Alto Research Centers, Palo Alto, CA, 1982.
Reprinted in Hayes-Roth&Waterrnan&Lenat83.

[Sussman 77] Sussman, 6. J.
Electrical design: a problem for artificial intelligence research.
In *Proc. Fifth International Joint Conference on Artificial Intelligence,* pages 894-900. Massachusetts Institute of Technology, 1977.
Includes some general pointers; see also *Artificial Intelligence* 9: pp. 135-196, 1977 (EL)

[Sussman&Steele 80]
Sussman, Q. J. and Steele, G. L. Jr.
CONSTRAINTS - A language for expressing almost hierarchical descriptions.
Artificial Intelligence 14:1-39,1980.

References

[Teitelman&Masinter 81]
Teitelman, W. and Masinter, L
The INTERLISP programming environment.
Computer 14(4):25-33, April, 1981.
A flexible, responsive environment; issues of personal computing and user interfaces

[Waterman&Hayes-Roth 78]
Waterman, D. A. and Hayes-Roth, F. (Eds.).
Pattern-Directed Inference Systems.
Academic, New York, 1978.
Especially the first and last chapters; exhaustive references

[Waterman&Peterson 80]
Waterman, D. A. and Peterson, M.
Rule-based models of legal expertise.
In *Proceedings of the First Annual National Conference on Artificial Intelligence,* pages 272-275. AAAI, 1980.
Legal decision-making system (LDS); RITA and ROSIE languages

[Weiss&Kulikowski&Amarel&Safir 78]
Weiss, S. M., Kulikowski, C. A., Amarel, S. and Safir, A.
A model-based method for computer-aided medical decision-making.
Artificial Intelligence 11:146-172,1978.
CASNET expert on glaucoma; see also 6th IJCAI (1979), pp. 942-947 (EXPERT)

[Westerberg82] Westerberg, A. W.
Design research - both theory and strategy.
Chemical Engineering Education Winter/Spring: 12-16, 62-66,1981&82.
Amer. Society for Engineering Education, ChE Div. Award Lecture.

[Winston 77] Winston, P. H.
Artificial Intelligence.
Addison-Wesley, Reading, MA, 1977.
Chapters 1 and 5 are the most relevant