THE ART OF ARTIFICIAL INTELLIGENCE:
I. THEMES AND CASE STUDIES OF KNOWLEDGE ENGINEERING

by

E. A. Feigenbaum

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY
The knowledge engineer practices the art of bringing the principles and tools of AI research to bear on difficult applications problems requiring experts' knowledge for their solution. The technical issues of acquiring this knowledge, representing it, and using it appropriately to construct and explain lines-of-reasoning, are important problems in the design of knowledge-based systems. Various systems that have achieved expert level performance in scientific and medical inference illuminate the art of knowledge engineering and its parent science, Artificial Intelligence.
THE ART OF ARTIFICIAL INTELLIGENCE:
I. Themes and Case Studies of Knowledge Engineering

STAN-CS-77-621
Heuristic Programming Project Memo 77-25

Edward A. Feigenbaum

Department of Computer Science
Stanford University
Stanford, California

ABSTRACT

The knowledge engineer practices the art of bringing the principles and
tools of AI research to bear on difficult applications problems requiring
experts' knowledge for their solution. The technical issues of acquiring
this knowledge, representing it, and using it appropriately to construct
and explain lines-of-reasoning, are important problems in the design of
knowledge-based systems. Various systems that have achieved expert level
performance in scientific and medical inference illuminate the art of
knowledge engineering and its parent science, Artificial Intelligence.

The views and conclusions in this document are those of the author and
should not be interpreted as necessarily representing the official
policies, either express or implied, of the Defense Advanced Research
Projects Agency of the United States Government.

This research has received support from the following agencies: Defense
Advanced Research Projects Agency, DAHC 15-73-C-0435; National Institutes
of Health, 5R26-RRO0612, RR-00785; National Science Foundation, MCS 76-
11649, DCR 74-23461; The Bureau of Health Sciences Research and Evaluation,
NS-DL344.
THE ART OF ARTIFICIAL INTELLIGENCE

1. Themes and case studies of knowledge engineering

Edward A. Feigenbaum
Department of Computer Science,
Stanford University,
Stanford, California, 94305

Abstract

The knowledge engineer practices the art of bringing the principles and tools of AI research to bear on difficult applications problems requiring experts' knowledge for their solution. The technical issues of acquiring this knowledge, representing it, and using it appropriately to construct and explain line-of-reasoning, are important problems in the design of knowledge-based systems. Various systems that have achieved expert level performance in scientific and medical inference illustrate the art of knowledge engineering and its parent science, Artificial Intelligence.

1 INTRODUCTION: AN EXAMPLE

This is the first of a pair of papers that will examine emerging themes of knowledge engineering, illustrate them with case studies drawn from the work of the Stanford Heuristic Programming Project, and discuss general issues of knowledge engineering art and practice.

Let me begin with an example new to our workbench: a system called PUFF, the early fruit of a collaboration between our project and a group at the Pacific Medical Center (PMC) in San Francisco.

A physician refers a patient to PMC's pulmonary function testing lab for diagnosis of possible pulmonary function disorders. For one of the tests, the patient inhales and exhales a few times in a tube connected to an instrument/computer combination. The instrument acquires data on flow rates and volumes, the so-called flow-volume loop of the patient's lungs and airways. The computer measures certain parameters of the curve and presents them to the diagnostician (physician or PUFF) for interpretation. The diagnosis is made along these lines: normal or diseased; obstructive lung disease or restrictive lung disease or a combination of both; the severity; the likely disease type(s) (e.g., emphysema, bronchitis, etc.) and other factors important for diagnosis.

PUFF is given not only the measured data but also certain items of information from the patient record, e.g., age, sex, number of pack-years of cigarette smoking. The task of the PUFF system is to infer a diagnosis and print it out in English in the normal medical summary form of the interpretation expected by the referring physician.

Everything PUFF knows about pulmonary function diagnosis is contained in (currently) 55 rules of the IF...THEN...form. No textbook of medicine currently records these rules. They constitute the partly-public, partly-private knowledge of an expert pulmonary physiologist at PMC, and were extracted and polished by project engineers working intensively with the expert over a period of time. Here is an example of a PUFF rule (the unexplained acronyms refer to various data measurements):

---

**RULE 31**

IF:  
1) The severity of obstructive airways disease of the patient is greater than or equal to mild, and  
2) The degree of diffusion defect of the patient is greater than or equal to mild, and  
3) The tlc(bdy)observed/predicted of the patient is greater than or equal to 110 and  
4) The observed-predicted difference in rvl/vol of the patient is greater than or equal to 10

THEN:  
1) There is strongly suggestive evidence (2.5) that the subtype of obstructive airways disease is emphysema, and  
2) It is definite (1.0) that "OAD, Diffusion Defect, elevated TLC, and elevated RV together indicate emphysema." is one of the findings.

---

1
in the post decade of the art of engineering knowledge-based intelligent agents.

In the remainder of this essay, I would like to discuss in detail that one research group, the Stanford Heuristic Programming Project, has taken, illustrating in case studies, and discussing themes of the work.

2 ARTIFICIAL INTELLIGENCE & KNOWLEDGE ENGINEERING

The dictionary that was used to classify the collected papers in the volume "Computer and Thought" still characterizes all the motivations and research efforts of the AI movement. In this essay, I will discuss how the "artificial intelligence community is being redefined by the work of the Stanford Heuristic Programming Project."

In this context, I will refer to the "rules of expertise" and the "rules of good judgment" of the expert practitioners of that domain that we seek to transfer to our programs.

2.1 Lessons of the Past

Two insights from previous work are pertinent to this essay.

The first insight is based on the power and generality of the inference engine used in the Stanford Heuristic Programming Project. (Some of the participants, in particular, have labeled the project "the power strategy"). We must be careful in our assumptions and in the limitations of what we can do. This is not a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system.

As the Stanford Heuristic Programming Project was being developed, there were two main concerns: that it could be used to solve problems in a specific domain, and that it could be used to solve problems in a general domain.

The second insight is based on the limitations of the inference engine used in the Stanford Heuristic Programming Project. (Some of the participants, in particular, have labeled the project "the power strategy"). We must be careful in our assumptions and in the limitations of what we can do. This is not a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system.

As the Stanford Heuristic Programming Project was being developed, there were two main concerns: that it could be used to solve problems in a specific domain, and that it could be used to solve problems in a general domain.

The second insight is based on the limitations of the inference engine used in the Stanford Heuristic Programming Project. (Some of the participants, in particular, have labeled the project "the power strategy"). We must be careful in our assumptions and in the limitations of what we can do. This is not a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system.

As the Stanford Heuristic Programming Project was being developed, there were two main concerns: that it could be used to solve problems in a specific domain, and that it could be used to solve problems in a general domain.

The second insight is based on the limitations of the inference engine used in the Stanford Heuristic Programming Project. (Some of the participants, in particular, have labeled the project "the power strategy"). We must be careful in our assumptions and in the limitations of what we can do. This is not a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system.

As the Stanford Heuristic Programming Project was being developed, there were two main concerns: that it could be used to solve problems in a specific domain, and that it could be used to solve problems in a general domain.

The second insight is based on the limitations of the inference engine used in the Stanford Heuristic Programming Project. (Some of the participants, in particular, have labeled the project "the power strategy"). We must be careful in our assumptions and in the limitations of what we can do. This is not a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system. It is a powerful tool that can be used to solve specific problems in a specific domain. But it does not have the generality of a "general" system.
areas in slight. A chief master is unlikely to be an expert algebraist or an expert mass spectroscopy. In this view, the expert is in the specialism, and a specialist's methods are heuristics. (Fleischer, Buchanan and Lederberg, 1971, p. 187)

Subsequent evidence from our laboratory and others has only confirmed this belief.

At researchers have dramatically shifted their view on generality and power. In the past decade, in 1963, the fundamental question about the DENDRAL program was: "Is there a paradigm shift in AI?" in 1977, Goldstein and Papert write of a paradigm shift in AI:

"Today there has been a shift in paradigm. The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a form that permits their effective use and interaction." (Goldstein and Papert, 1977)

The second insight from past work concerns the nature of this knowledge. That an expert brings to the performance of a task. Experience has shown it to be a highly heuristic, knowledge, expertise, uncertain — mostly "good reasons" and "good practice," in lieu of facts and rigor. Experience has also taught that much of this knowledge is private to the expert; not being to share publicly how he performs, but how he enacts it. He knows more than he is aware of knowing. His explicit knowledge is private to the expert, not being to share publicly how he performs.

3.2 The Knowledge Engineer

The knowledge engineer is that second party just discussed, determining what is implicit about the term. In the mid-60s, John McCarthy, for example, was describing Artificial Intelligence as "applied Epistemology," but it was not until Donald Michie in 1968, he reported that it was "epistemological engineering, a clever but ponderous and unpronounceable term of art that I simplified into "knowledge engineering."" In deference to my favorite knowledge engineer, he works intensively with an expert to acquire domain-specific knowledge and organizes it for use by a program. Simultaneously she is matching the tools of the AI workbench to the task at hand — program organizations, methods of symbolic inference, techniques for the structuring of symbolic information, and the like. If the tool she uses it. If not, gets created. She builds the early versions of the intelligent agent, guided always by her intent that the program eventually achieve expert levels of performance in the task. She refines or reconceptualizes the system as the increasing amount of acquired knowledge causes the AI tool to "break" or slow down intolerably. She also refines the human interface to the intelligent agent with several aims: to make the system appear "comfortable" to the human user in his linguistic transactional style with the machine; to make the system's inference processes understandable to the user; and to make the assistance controllable by the user when, and for whatever part of a real problem, she has an insight that previous was not elicited and therefore not incorporated.

In this section, I wish to explore (in summary form) some case studies of the knowledge engineer's art.

3 CASES FROM THE KNOWLEDGE ENGINEER'S WORKSHQIP

I will draw material for this section from the work of my group at Stanford. Much exciting work in knowledge engineering is going on elsewhere. Since my intent is not to write a literature survey, the risk of appearing partial I have used as case studies the work I know best.

My colleagues (Professors Lederberg and Buchanan) and I began in 1965 the development of the DENDRAL program. In 1965, we did deal with a scientific problem solving and discovery, particularly the processes of an unaided, scientist should use in inferring hypotheses and theories from empirical evidence. To conduct this study in such a way that our experimental program would yield one domain, working scientists, providing intelligent assistance on important and difficult problems. By 1970, we and our co-workers had gained enough experience that we felt comfortable using not a program of rules, but one that could be used by a group of scientists in a research laboratory. Our "expert system," DENDRAL, was designed to provide assistance on the structure and interpretation of organic mass spectrum data. The basic assumption was that some structural hypotheses would be generated by an expert, but not enough to be comprehensible or not large enough to be comprehensible to a human expert. The overall approach allows great flexibility for adding, removing, or changing knowledge in the system. The hallmark of the system is the large number of knowledge and experience that are used in the interpretation of a single mass spectrum.

Line-of-reasoning: A central organizing principle in the design of the knowledge-based intelligent agents is the maintenance of a line-of-reasoning, the development of which is not limited by the temporal sequence of sentences, but which can be used in an appropriate manner. This principle is, of course, not a logical necessity, but stems from the need to be an engineering principle of major importance.

Multiple Sources of Knowledge: The formation and maintenance (support) of the line-of-reasoning usually require the integration of many disparate sources of knowledge. The representational and inferential problems in achieving a smooth and effective integration are formidable engineering problems.

Explanation: The ability to explain the line-of-reasoning in a language convenient to the user is necessary for application and for system development (e.g. for debugging and for extending the knowledge base). Once again, this is an engineering principle, but very important. What constitutes an "explanation" is not a single concept, and considerable thought needs to be given to it in each case, to the structuring of explanations.

CASE STUDIES

In this section I will try to illustrate these themes with various case studies.

3.1 DENDRAL: Inferring Chemical Structures

3.1.1 Chemical Note

Begun in 1965, this collaborative project with the Stanford Linear Accelerator Laboratory has become one of the longest-lived continuous efforts in the history of AI. The project, in no small way, has contributed to its success. The basic framework of DENDRAL, large and inconsistent, has proven rugged and extensible. For us DENDRAL is a testbed for ideas which may have found their way, highly metamorphosed, in the work of other researchers. The long-standing commitment to rule-based representation, for example, our attempt to head off the imminent obsolescence of DENDRAL, has been based on the rapid accumulation of new knowledge in the area of organic chemistry.
3.1.7 Representations

Chemical structures are represented as node-link graphs of atoms (nodes) and bonds (links). Constraints on search are represented as subgraphs (atomic configurations) to be deleted or preferred. The search process of DENDRAL is represented by a set of rules of the general form:

\[
\text{Situation: Particular atomic configuration (subgraph)}
\]

\[
\text{If Probability, P, of occurring is greater than threshold, T}
\]

\[
\text{Then Action: Fragmentation of the particular configuration (breaking links)}
\]

Rules of this form are natural and expressive to mass spectrometrist.

3.1.8 Sketch of Method

DENDRAL's inference procedure is a heuristic search that takes place in three stages, without feedback: plan-generate-test.

"Generate" (a program called CONDOR) is a generation process for plausibly structured families. Its foundation is a combinatorial algorithm with heuristically proven properties of completeness and non-redundant generation) that can produce all the topologically legal candidate structures. Constraints supplied by the user or by the "Plan" process prune and steer the generation to produce the plausible set (i.e., those satisfying the constraints) and not the enormous legal set.

"Test" refines the evaluation of plausibility, discarding less worthy candidates and rank-ordering the remainder for examination by the user. "Test" first produc a predicted set of instrument data for each plausible candidate, using the rules described. It then evaluates the set of each candidate by comparing the predicted data with the actual input data. The evaluation is based on heuristic criteria of goodness-of-fit. Thus, "test" selects the "best" explanations of the data.

"Plan" produces direct (i.e., not chained) inference about likely substructures in the molecule from patterns in the data that are indicative of the presence of the substructure. (Fattor in the data triggers the left-hand-sides of substructure rules). Though composed of many atoms whose interconnections are given, the search space can be manipulated as atomically by "generate." Aggregating many units entering into a combinatorial process (data) reduces the size of the combinatorial search space. "Plan" helps to reduce the search space so as to be relevant to the input data. "Generate" is the inference tactic "Plan" and "Generate." User-supplied constraints enter this interface, directly or from user-assist packages, in the form of structures.

3.1.5 Sources of Knowledge

The various sources of knowledge used by the DENDRAL system are:

- Rules (rules of connections of atoms; stable and unstable configurations of atoms; rules for mass spectrometry fragmentation; rules for NMR shifts; expert's rules for planning and evaluation; user-supplied constraints (contextual)).

3.1.6 Result

DENDRAL's structure elucidation abilities are far-reaching, both very general and very narrow. In general, DENDRAL handles all molecules, cyclic and acyclic, simple and complex. DENDRAL's performance in structure elucidation with instrument data, DENDRAL's performance as a human performance only for a few of the chemical families for which the program has been given a degree of knowledge, namely the families of interest to our chemical collaborators. I will spare this computer science audience the list of names of these families. Within these areas of knowledge-intensity, DENDRAL's performance is usually not only much faster but also more accurate than human performance.

The statement just made summarizes thousands of runs of DENDRAL on problems of interest to our experts, their colleagues, and their students. The results obtained, along with the knowledge that has been obtained to DENDRAL to obtain them, are published in the scientific literature. In fact, 25 papers have been published there, under a series title "Applications in Chemical Intelligence: Specific Subjects" (see references).

DENDRAL is in everyday use by Stanford chemists, their collaborators at other universities and collaborating or otherwise interested chemists in industry. Users outside Stanford access the system over commercial computer networks. The problem-solving results are often difficult and novel. The British government is currently supporting work in the USA aimed at transferring DENDRAL to industrial user communities in the UK.

3.1.7 Discussion

Representation and expressibility. The representation chosen for the molecule, constraints, and rules of instrument data interpretation is sufficiently close to that used by humans in thinking about molecular structure elucidation that the knowledge base has been extended smoothly and easily, mostly by chemists themselves. The use of only one major reprogramming effort took place in the last 9 years—when our computer was created to deal with cyclic structures.

Representation and the integration of multiple sources of knowledge. The generally difficult problem of integrating various sources of knowledge has been made easy in DENDRAL by careful engineering of the representations of objects, constraints, and rules. We insisted on a common language of compatibility of the representations with each other and with the inference processes: the language of molecular structure expressed as graphs. This leads to a straightforward procedure for adding a new source of knowledge, say, for the knowledge associated with NMR data. The procedure is this: write rules that describe the effect on the physical process of the instrument on molecules using the situation = action form with molecular graphs on both sides; any special inference process using these rules must pass its results to DENDRAL's generator module(s) in the common graph language.

It is today widely believed in AI that the use of many different sources of knowledge in problem solving and data interpretation has a strong effect on the quality of performance. Now all sources of course, domain-dependent, but the impact of bringing together knowledge from many different sources of knowledge can be startling. In one difficult (but not unusually difficult) mass spectrometry analysis, the program using its mass spectrometry knowledge alone would have generated a very large set of plausible chemical candidates (over 1.25 million). Our engineering effort to make use of source of data and knowledge, proton NMR. The addition of a single interactive source of NMR data, from which the program could infer a few additional constraints, produces a much smaller set of plausible candidates to one, the right structure! This was an isolated case, but not an isolated one. The analysis of an acrylamide with formula C3H6N2O.
program. The inference is to be made from actual aspects recorded from known molecular structures. The output of the system is the set of facts discovered, summary of the evidence supporting each rule, and a summary of non-indicating evidence. User-supplied constraints can also be input to force the form of rules along desired lines.

3.2.3 Representations

The rules are, of course, of the same form as used by DENDRAL that was described earlier.

3.3 Sketch of Method

META-DENDRAL, like DENDRAL, uses the generation-and-test framework. The process is organized in three phases: (1) Retrieve the data and summarize evidence (INTSUM); generate plausible candidates for rules (RULEGEN); test and refine the set of plausible rules (RULEMOD).

INTSUM: gives every data point in every spectrum an interpretation as a possible (highly specific) fragmentation. It then summarizes statistically the "weight of evidence" for fragmentations and for atomic configurations that cause these fragmentations. Thus, the job of INTSUM is to translate in DENDRAL subgraphs and bond-breaks, and to summarize the evidence accordingly.

RULEGEN: conducts a heuristic search of the space of all rules that are legal under the DENDRAL rule system and the user-supplied constraints. It searches for plausible rules, i.e. those for which positive evidence exists. A search path is permissible provided there is no evidence for rules of the class just generated. The search tree begins with the (single) most general rule (loosely put, "anything" fragments from "anything") and proceeds level-by-level toward more detailed specifications of the "anything." The heuristic stopping criterion measures whether a rule being generated has become too specific, in particular whether it is applicable to too few molecules of the input set. Similarly there is a criterion for when an emerging rule is too general. Thus, the output of RULEGEN is a set of candidate rules for which there is positive evidence.

RULEMOD: tests the candidate rule set using more complex criteria, including the presence of negative evidence. It removes redundancies from the candidate rule set; merges rules that are supported by the same evidence; tries further specialization of candidates to remove negative evidence; and tries further generalization that preserves positive evidence.

3.3.1 Historical note

MYCIN originated in the Ph.D. thesis of E. Shortliffe (now Shortliffe, M.D. as well), in collaboration with the Infectious Disease group at the Stanford Medical School (Shortliffe, 1976). TEIRESIAS, the Ph.D. thesis work of C. Davis, arose from issues and problems indicated by the MYCIN project but generalized by Davis beyond the bounds of medical diagnostic applications (Davis, 1976). Other MYCIN-related themes are in progress.

3.3.2 Tasks

The MYCIN performance task is diagnosis of blood infections and meningitis infections and the recommendation of drug treatment. MYCIN conducts a consultation (in English) with a physician about a patient case, constructing lines-of-reasoning leading to the diagnosis and treatment plan.

The TEIRESIAS knowledge acquisition task can be described as follows:

In the context of a particular consultation, confront the expert with a diagnosis with which he does not agree, lead the expert systematically back through the lines-of-reasoning that produced the diagnosis to the points at which he indicates the analysis went awry. Interact with the expert to modify offending rules or to acquire new rules. Further the consultation to test the solution and gain the expert's concurrence.

3.3.3 Representations

MYCIN's rules are of the form:

IF conjunctive clause THEN implication

The following is an example of a MYCIN rule for blood infections:

\[ \text{IF} \quad (pseudomonas-aeruginosa) \quad \text{THEN} \quad (\text{mycetoma}) \]

3.3.4 Sketch of method

MYCIN deploys a generation-and-test procedure of a familiar sort. The generation of steps in the line-of-reasoning is accomplished by MYCIN's own inference engine. An if-then rule is either immediately true or false (as determined by patient data or by data entered by the physician in the consultation); or is to be decided by subgoaling. Thus, "tests" is interleaved with "generation" and serves to prune out incorrect lines-of-reasoning.

Each rule supplied by an expert has associated with it a "degree of certainty" representing the expert's confidence in the validity of the rule (a number from 1 to 10). MYCIN uses a particular ad-hoc but simple model of interacting reasoning to compute the degree of certainty of the rules used in an inference chain (Shortliffe and Buchanan, 1975).

It follows that there may be a number of "somewhat true" lines-of-reasoning -- some indicating one diagnosis, some indicating another. All (above a threshold) are used by the system as sources of knowledge indicating plausible lines-of-reasoning.

TEIRESIAS's rule acquisition process is based on a record of MYCIN's search. Rule acquisition is guided by a set of rule models that dictate the form and indicate the likely content of new rules. These models are not given in advance, but are inferred from the knowledge base of existing rules.

3.3.5 Language of Interaction

The language used looks like it might be English but is actually the direct "Doctorese" used by members of the tribe of healing arts practitioners. Reasoning processes and language processing methods suffice. When ambiguities of interpretation are encountered, they are fed back to the user for decisions about meaning.

3.3.6 Example diagnosis and treatment plan

The following is an example of a diagnosis reported by MYCIN to one of its users following a consultation. An example consultation will not be shown, but one can be found in Shortliffe's book (Shortliffe, 1976).

My therapy recommendations will be designed to treat for organisms that are either very likely or, although less likely, would have a significant effect on therapy selection if they were present. It is important to
3.3.7 Results

MYCIN is currently being prepared for clinical evaluation. In a preliminary evaluation, done outside of the clinical setting, a panel of expert judges was asked to rate MYCIN's performance along a number of dimensions (e.g., infecting organism identification, organism susceptibility). In 90% of the cases submitted to the judges, a majority of the judges said that the program's decisions were the same as or superior to the decisions they would have made.

3.3.8 Explanation

MYCIN can expose its line of reasoning by offering explanations of various kinds. "Why" and "How" questions can be asked by the user during the consultation. "Why" is the user's way of inquiring, "Why do you want to know that?" "How" is his way of asking, "How did you conclude that...?" In a post-consultation, he can similarly inquire about the diagnostic conclusions were reached, and inquire about the reasons why certain other conclusions were ruled out. Examples of the MYCIN-TEIRESIAS explanation capabilities follow:

**MYCIN**: Is the patient's illness with (organism-1) a hospital-acquired infection?
**USER**: Why?
**TEIRESIAS**: (i.e., why is it important to determine whether or not the infection with (organism-1) was acquired while the patient was hospitalized?)

[1.0] This will aid in determining the category of (organism-1).

It has already been established that

(1.1) the gram stain of (organism-1) is gram-positive,
(1.2) the morphology of (organism-1) is rod, and
(1.3) the aerobicity of (organism-1) is facultative

Therefore, if

(1.4) the infection with (organism-1) was not acquired while the patient was hospitalized then

there is strongly suggestive evidence (.8) that the category of (organism-1) is enterobacteriaceae

Also, there is weakly suggestive evidence (.3) that the identity of (organism-1) is pseudomonas-aeruginosa

**RULE 37**

Because (organism-1) is pseudomonas-aeruginosa

**MRI**: How can I?

**TEIRESIAS**: (i.e., how can I establish that the aerobicity of (organism-1) is facultative?)

The following were used:

(3.1) RULE 27 indicated there is strongly suggestive evidence (.8) that the aerobicity of (organism-1) is facultative
(3.2) RULE 27 indicated there is weakly suggestive evidence (.2) that the aerobicity of (organism-1) is anaerobic

Since this gave a cumulative CF of (.8) for facultative, and (.2) for anaerobic, it has been established that the aerobicity of (organism-1) is facultative.
3.4.3 Representations

The rules given by the expert about objects, their relationships, and the generation of signals from these objects are all represented in the system as action forms. The "situation" consists of the objects (which are processes that modify the current hypotheses, most unresolved, low quality, exception evaluations, etc.) the expert's knowledge of how to do analysis in the task is also represented in rule form. These strategy rules replace the normal executive program.

The situation-hypothesis is represented as a node-link graph, treelike in that it has distinct "levels," each representing a degree of abstraction (or aggregation) that is natural to the expert in his understanding of the domain. A node represents an hypothesis; a link to that node represents support for that hypothesis (as in hypotheses networks or "support from below"). "Lower" levels are concerned with the specifics of the signal data; "higher" levels represent symbolic abstractions.

3.4.4 Search method

The situation-hypothesis is formed incrementally. As the situation unfolds over time, the triggering of rules modifies or corrects existing hypotheses, adds new ones, or changes support values. The situation-hypothesis is a sequence of situations, or "blackboards," in SPARL jargon for all the rules.

In general, the incremental steps toward a more complete and refined situation-hypothesis are: Some of the rules are plausibility-move generators, generating either nodes or links. On the other hand, there exist test-action, teaching and modifying node descriptions.

In typical operation, new data is submitted for processing (as a set of new rules). This initiates a flurry of rule-triggering and concurrently rule-modifications (called "events"). Some events are direct consequences of the data; other events arise to a cascade-like fashion from the triggering of rules. Auxiliary symbolic data also cause events, usually affecting the higher levels of abstraction. For example, a consequence, support-from-above for the lower level processes is made available; and expectations of possible lower level events can be formed. Eventually all the rules that modify their own symbol and the system becomes quiescent, thereby triggering the input of new data to restart the inference activity.

The system uses the simplifying strategy of maintaining only one "best" situation-hypothesis at any moment, modifying it incrementally as required by the changing data. This approach is made feasible by several characteristics of the domain. First, there is the strong continuity over time of object behavior. (Specifically, they do not change radically over time, or behave radically differently over short periods.) Thus the appropriate explanation must be symbolic and relatively permanent.

Since the appropriate explanation is symbolized, it is possible to refer up to the current hypothesis. Thus contrast this with non-symbolic, full-fledged, high-level explanations of rule invocations in the construction of a reasoning chain.

3.4.5 Results

In the test application, using signal data generated by a simulation of a known process, where the data was not available, the program achieved expert levels of performance over a span of test periods. Under normal rules triggering, there was very little primary signal to support inference. Other rules tended to become too much signal induced a plethora of alternatives with much ambiguity.

A modified SUX design is currently being used as the basis for an application to the interpretation of X-ray crystallographic data, the CRYSALIS program mentioned later.

3.4.6 Discussion

The role of the auxiliary symbolic sources of data is of critical importance. They supply a symbolic model of the existing situation that is used to generate explanations of events to be observed in the data stream. This allows flow of inferential inferences from higher levels of abstraction to lower. Such a process, so familiar to AI researchers, apparently is almost unrecognized among signal processing engineers. In the application area investigated, the ability to generate knowledge is essential in controlling the combinatorial processing explosion at the lower levels, since it is the explosion that forces the traditional signal processing engineers to seek out the largest possible number-cruncher for their work.

The design of appropriate explanations for the user takes an interesting twist in SUX. The situation-hypothesis unfolds piecemeal over time, but the "appropriate" explanation for the user is one that is generated over the long period. Thus the appropriate explanation must be plausible, strongly supported ones that have been led up to the current hypothesis. Contrast this with the "appropriate" explanation for the user being held by the expert, symbolizing a "list" of rule invocations in the construction of a reasoning chain.

Since its knowledge base and its auxiliary symbolic data give it a model-of-the-situation that strongly constrains interpretation of the primary data stream, SUX is relatively unperturbed by errorful or missing data. These data consist of so many fluctuations in the credibility of individual hypotheses and/or the creation of the "adult-ante" events. But there is (but not yet been) use to control sensors. Since there is a "minimal sets" of evidence necessary to establish support, and since it is currently processing a complete hypothesis structure, it can request "critical readings" from the sensors. In general, this allows an efficient use of limited bandwidth and data-acquisition processing capability.

3.5 OTHER CASE STUDIES

Space does not allow more than just a brief sketch of other interesting projects that have been completed or are in progress.

3.5.1 AI: mathematical discovery

A knowledge-based system that conjectures interesting concepts is elementary mathematics. It is a discoverer of interesting theorems and theorem proving program.

Involving over a dozen researchers, was created and executed by D. Lenat for his Ph.D. thesis, and is reported by him in these proceedings ("An Overview of AM").

AM's knowledge is basically of two types: rules that allow the discovery of new concepts from previously conjectured concepts; and rules that allow the discovery of "interestingness" of a conjecture. These rules are not formally defined. The expertise of the professional mathematician at the task of mathematical discovery. Though Lenat is not a professional mathematician, he was able successfully to serve as his own expert in the built-in AM.

AM conducts a heuristic search through the space of concepts created from its rules. Its basic steps are: generate a set of possible new concepts; generate a set of possible new concepts. The test is the evaluation of "interestingness." Of the expertise of the professional mathematician at the task of mathematical discovery. Though Lenat is not a professional mathematician, he was able successfully to serve as his own expert in the built-in AM.

AM conducts a heuristic search through the space of concepts created from its rules. Its basic steps are: generate a set of possible new concepts; generate a set of possible new concepts. The test is the evaluation of "interestingness." Of the expertise of the professional mathematician at the task of mathematical discovery. Though Lenat is not a professional mathematician, he was able successfully to serve as his own expert in the built-in AM.
is a protein chemist's art. As always, capturing this art in heuristic rules and putting it to use with an inference engine is the project's goal.

For the CRYSTALS system, the basic strategy is to find a modification of the SU7X system design described above. The hypothesis formation process must deal with many levels of feasible aggregation and abstraction. For example, the map itself can be viewed as consisting of "peaks," "seas," "peaks and valleys," or "skeletons." The protein model has "atoms," "twelve planes," "active side-adecchar," and even massive substructures such as "helixes." Protein molecules are so complex that a systematic generation-and-test strategy like DENDRAL is not feasible. Incremental piecing together of the hypothesis using region-growing methods is necessary.

The CRYSTALS design (also SU7X in a different notation) is described in detail in a recent paper by Mill and Feigenbaum (1977).

4 SUMMARY OF CASE STUDIES

Some of the themes presented earlier need no recapitulation, but I wish to revisit three: generation-and-test; situation → action rules; and explanations.

4.1 Generation and Test

Aircraft come in a wide variety of sizes, shapes, and functional designs and they are applied in very many ways. But almost all that fly do so because of the fundamental physical principle of lift by airflow; the others are described by exception. So it is with intelligent agent programs and, the information processing psychologists generally agree, the computer scientists. One unifying principle of "intelligence" is generation-and-testing. It has been seen thoroughly in AI research.

In the case studies, generation is manifested in a variety of forms and processing schemes. There are legal move generators defined for chess; there are connectionist models (as in BRYANT'S work); there are simulator generation programs (as in DENDRAL). Generation is interleaved with testing (as in MCDIN, SU7X, and AI). Solution generation precedes testing (DENDRAL). One case (META-DENDRAL) is mixed, with some testing taking place during generation, soon after.

Text also shows great variety. There are simple tests (MCDIN: "Is the organism balloon?"; SU7X: "Has a spiral line appeared at position F7?"; Some texts are complex heuristic evaluations (AI: "Is the concept interesting?"; NODGEN: "Will the reaction actually take place?" Sometimes a complex test can involve feedback to modify the object being tested (as in META-DENDRAL).

The evidence from our case studies supports the assertion by Newell and Simon that generation-and-test is a law of our science (Newell and Simon, 1976).

4.2 Situation → Action rules

Situation → Action rules are used to represent experts' knowledge in all of the case studies. Always the situation part indicates the specific conditions under which the rule is relevant. The action part can be simple (MCDIN: conclude presence of particular organisms; DENDRAL: conclude break of particular bond). Or it can be quite complex (MCDIN: areola procedure). The overwhelming consideration in making design choices is that the rule form chosen be able to represent the situation and clearly and directly what the expert wishes to express about the domain. As illustrated, this may necessitate a wide variation in rule syntax and semantics.

From a study of all the projects, a regularity emerges. A salient feature of the Situation → Action rule technique for representing expert's knowledge is the modularity of the knowledge base. It allows the knowledge engineer to flexibly add or change the knowledge easily as the experts' understanding of the domain changes. Here too one must be pragmatic, not dogmatic. A technique such as this cannot represent as much knowledge as the unified approach (as if that modularity does not exist in the domain). The virtue of this technique is that it serves as a framework for discovering what modularity exists in the domain. The process of discovering and reformulation of the domain toward greater modularity is a case of the "helices." The flexi-modularity of the Situation → Action rule technique is an opportunity for the expert, the knowledge engineer, and the knowledge applications engineer to continue to work toward a more modular knowledge base.

Finally, our case studies have shown that strategy knowledge can be captured in rule form. In CRYSTALS, the metarules capture an explanation of how to deploy domain knowledge; in SU7X, the strategy rules represent the experts' knowledge of "how to analyze" in the domain.

4.3 Explanation

Most of the programs, and all of the more recent ones, make available an explanation capability for the user; the end-user or system developer. Our focus on end-users and applications for the knowledge has forced attention to human engineering issues, in particular the need for the explanation capability imperative.

The Intelligent Agent viewpoint seems to us to demand that the agent be able to explain its activity; else the question arises of who is in control of the agent's activity. The issue is not academic or philosophical. It is an engineering issue that has arisen in medical and military applications of intelligent agents, and will govern future acceptance of AI work in applications areas. And on the philosophical level one might even argue that there is a moral imperative to provide accurate explanations to end-users whose intuitions about our systems are almost nil.

Finally, the explanation capability is needed as part of the concerted attack on the knowledge acquisition problem. Explanation of the reasoning process is central to the interactive transfer of expertise to the knowledge base, and it is our most powerful tool for the debugging of the knowledge base.

5 CONCLUSION

What we have learned about knowledge engineering goes beyond what is discernible in the behavior of our case study programs. In the next paper of this two-part series, I will raise and discuss many of the general concerns of knowledge engineers, including these:

What constitutes an "application" of AI techniques?

Are there differences between a serious application and an application-flavored toy problem?

What are some criteria for the judicious selection of an application of AI techniques?

6 ACKNOWLEDGMENT

The work reported herein has received long-term support from the Defense Advanced Research Projects Agency. The National Institute of Health has supported DENDRAL, META-DENDRAL, and the SIM-IAM computer facility on which we compute. The National Science Foundation has supported research on CRYSTALS and MCDIN. The Bureau of Health Sciences Research and Evaluation has supported research on MYCIN. I am grateful to these agencies for their continuing support of our work.

I wish to express my deep admiration and thanks to the faculty, staff and students of the Computer Programming Project, and to our collaborators in the various worldly arts, for the creativity and enthusiasm they have made our work exciting and fruitful. My particular thanks for the help I have received in preparing this manuscript go to Randy Davies, Penny Mill, Reid Smith, and Carolyn Tayan.

The side benefits to the expert of his investment in the knowledge engineering activity.

Getting consensus among experts about the knowledge of a domain.

The consensus may be a more valuable outcome of the knowledge engineering effort than the building of the program.

Problems faced by knowledge engineers today.

The lack of adequate and appropriate computer hardware.

The difficulty of export of systems to end-users, caused by the lack of properly-sized and -packaged combinations of hardware and software.

The chronic absence of cumulation of AI techniques in the form of software packages that can achieve wide use.

The shortage of trained knowledge engineers.

The difficulty of obtaining and sustaining funding for interesting knowledge engineering projects.
REFERENCES

General

Feigenbaum, E.A. "Artificial Intelligence Research: What is it? What has it achieved? Where is it going?" invited paper, Symposium on Artificial Intelligence, Canberra, Australia, 1974.


DENDRAL and META-DENDRAL


MITH


PARTIAL


Davis, R., "Interactive Transfer of Expertise 2: Acquisition of New Inference Rules," these Proceedings.


MEC


GENE


MOLGEN


CRYSTALS