OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

16178

Distr.
LIMITED

IPCT.41(SPEC.)
23 September 1987

ENGLISH

UNITED NATIONS
INDUSTRIAL DEVELOPMENT ORGANIZATION

EXPERT SYSTEMS:
PROSPECTS FOR DEVELOPING COUNTRIES*

prepared by

Anil K. Jain**
UNIDO Consultant

---

* The views expressed in this paper are those of the author and do not necessarily reflect the views of the secretariat of UNIDO. Mention of firm names and commercial products does not imply an endorsement by UNIDO. This document has been reproduced without formal editing.

4.66

** Professor, Department of Computer Science, Michigan State University, East Lansing, Michigan, USA.

SUMMARY

Expert systems or knowledge-based systems are important and useful products of Artificial Intelligence (AI) research that offer tremendous possibilities for developing countries.

An expert system is a computer program that can provide effective solutions to problems in a limited domain normally solved by a human expert. The potential uses of expert systems are intriguing and exciting. Expert systems have been used to diagnose, analyze, design, plan, learn, monitor, instruct and manage. In many domains, expert systems have performed tasks as well as, or better than human experts.

This report introduces the topic of expert systems to a layman. Several case studies illustrate how expert systems are developed and used. Potential uses of expert systems in developing countries are outlined along with a project proposal for an expert system for transfer of technology agreements. Commercially available expert systems are described and an extensive list of references is also provided.

PREFACE

The idea for writing a report on Expert Systems and its potential applications in developing countries originated during my brief visit to UNIDO Headquarters, Vienna, in July 1986. I am grateful to Dr. Venkataraman and Dr. Fialkowski for their encouragement and support. I spent one week at the Instituto de Investigaciones Eléctricas (IIE), Mexico, in December 1986 to understand the needs and implications of developing expert systems for solving various problems in developing countries. I want to thank the staff of IIE for their hospitality.

Joe Miller provided invaluable assistance in collecting the background information and references for this report. R.D. Ramaswami assisted in editing an earlier version.

Reference to dollars ($) in this report is to United States dollars.

CONTENTS

ANNEXES

I. WHAT IS AN EXPERT SYSTEM ?

The field of Artificial Intelligence (AI) has interested computer scientists, engineers and system designers for the past 25 years. The major goal of AI research has been to develop computerized systems that can perform tasks and make decisions considered to be intelligent by humans. In spite of vigorous research, currently no AI system possessing intelligence comparable to that of, say, a five-year-old child is available. However, AI systems have outperformed humans in several specific tasks and domains. For example, computer programs that play checkers or chess can easily defeat an expert human player.

The popularity of AI is due to the tremendous opportunities it offers in increasing productivity, reducing design and manufacturing costs, improving the quality of decisions, and relieving humans from routine or hazardous tasks. AI technologies have been applied successfully towards solving problems in robotics, image understanding, VLSI circuit design, natural language processing, operations planning, trouble shooting, speech understanding, decision support systems, software development, data base, and of course, expert systems. This report introduces the expert system technology that has received significant attention in the marketplace. However, readers should be aware of these words of caution:

(a) Even though expert systems technology has been applied to a variety of task domains, it is not suitable and may even be inappropriate for other domains.

(b) The goal of research and development in the area of expert systems is to develop computerized knowledge-based systems whose performance is comparable with or exceeds that of a human expert. Commercially available expert systems have not yet replaced human experts; they assist human experts in routine tasks and provide training to novices.

An expert system is a computer program that is analogous to a human expert. Human experts are doctors, systems analysts, real estate agents, brokers, cotton farmers, geologists and others who possess a large background of knowledge in a field of expertise. Expert systems have been developed to perform tasks as well as, or better than the human experts in these fields. The possibilities for using expert systems are almost limitless. "They can be used to diagnose, monitor, analyze, interpret, consult, plan, design, instruct, explain, learn and conceptualize" [GEVA85].

Design and development of expert systems is currently the hottest area in AI research. With the decreasing cost of computer hardware, the number of areas where expert systems can be applied economically is increasing. An expert system can solve problems normally solved by human experts, in a limited domain. Guidelines and rules of thumb, along with knowledge about the subject, are used by an expert system to come up with a plausible solution for the problem. Depending on the application, the answer an expert system comes up with can range from being the "best" answer (e.g., MYCIN) to being a "non-optimal", yet viable solution (e.g., R1). MYCIN, one of the most well-known expert systems, diagnoses bacterial infections in humans and recommends the smallest combination of antibiotics that a) will cover all bacterial infections that a patient's symptoms indicate, and that b) produces the fewest bad side effects. The expert system R1 consists of over 1,200 rules that configure Digital Equipment Corporation's VAX series computer systems [GEVA85]. The configuration is done in stages while complying with customers' orders. For example, in one stage various computer components are put in boxes and these boxes are then placed in cabinets. There are many ways to assemble the final product. However, R1 picks the first solution that satisfies all the known constraints (e.g., volume of objects placed in a cabinet must be less than the total cabinet size) without trying all possible configurations, and then picks the one that costs the least. In some domains all the known facts or hypotheses may not be accurately available and so the expert system may also report a confidence level with its conclusions.

A. Components of an Expert System

Expert systems differ from other computer programs (e.g., compilers) in that they generally consist of four main parts or modules:

(i) global database
(ii) knowledge base
(iii) inference engine
(iv) user interface

The advantage of having four separate modules in an expert system is that it keeps the data and program control structure separate from one another. Figure 1 shows the interaction between the four modules of a typical expert system.

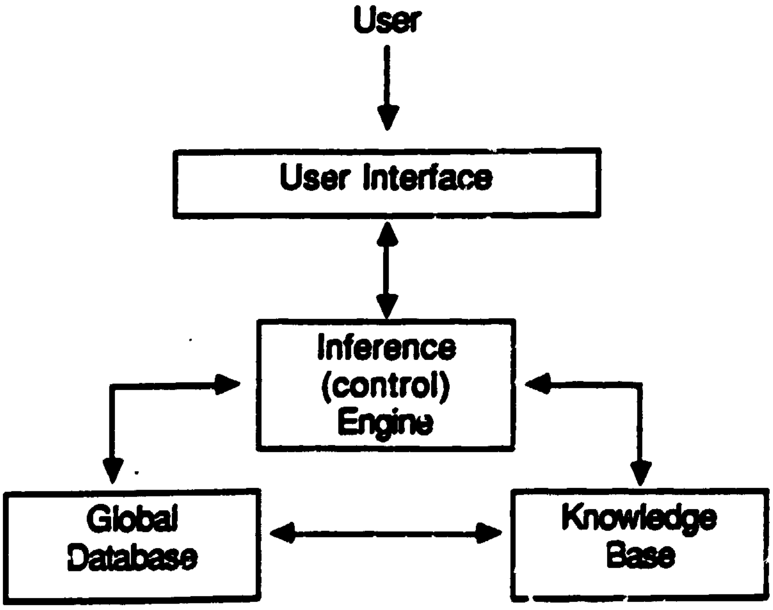

Figure 1: Information flow among the four modules in an expert system

In Figure 1, the knowledge base is the set of rules and guidelines containing the human expert's knowledge about the subject. The global database contains only facts about the current case being considered and is updated on a case by case basis. The inference engine is the control structure that determines which rules are checked and how the expert system arrives at its conclusion. The user interface handles data entry about particular cases and user queries to the database. Often it includes a natural language interface that translates internal data and rule representations to the user's native language. By keeping the four modules separate, it is easy to modify old rules or add new rules to the database without having to make any revisions to the control structure. For many applications, it is possible to use a standard or commercially available expert system shell (containing the inference engine) with the appropriate knowledge base (rules).

Comparing an expert system to a human expert, say a physician treating a patient, leads to the following analogies: Information about the patient such as temperature, blood pressure, weight, height, allergic reactions to various drugs and other personal facts, the patient's condition, and the current diagnosis would be kept in the global database. Questions a physician would ask the patient are learned from experience and education. These questions and guidelines would be represented in the knowledge base. The inference engine is similar to a doctor's line of reasoning. A doctor has considerable experience with patients and possible illnesses. He may start out by asking the patient if he is experiencing any pain, and if so, where the pain is located. The sequence in which the doctor asks the questions is influenced by the patient's answers. Similarly, the inference engine in the expert system determines which rules in the knowledge base to look at, and decides if the rules are satisfied according to various criteria.

B. A Simple Example

Consider a hypothetical expert system that can diagnose a patient as having a cold, a fever, or as being healthy. This example is simplified so that a small number of facts and rules can be used to demonstrate how an expert system can be implemented. A more sophisticated expert system to diagnose other ailments would require many more rules.

This example considers a patient named Joe Smith. The global database will start out containing only the facts relevant to the patient Joe Smith.

PATIENT IS JOE SMITH

TEMPERATURE IS 100 °F

Thus the global database starts out containing two facts about the current patient. Facts will be added to the global database as the expert system proceeds with its diagnosis. In more complicated cases facts will be modified and even deleted at times.

The knowledge base contains rules of thumb and guidelines that a physician would use in diagnosing a patient. These rules can be expressed as IF-THEN rules. This representation is easy for humans to understand and easy for computers to work with. Users should be able to understand how the expert system comes to its conclusion just as one would expect a doctor to be able to explain his diagnosis. The general format of IF-THEN rules is:

IF <statement> THEN <action>.

If the <statement> is true, then a specific <action> will be taken. For example, the IF-THEN rule:

1) IF 100 °F $\leq$ TEMPERATURE $\leq$ 101 °F THEN ASSERT "DIAGNOSIS

IS MILD FEVER"

would be evaluated by the expert system as follows: The value of TEMPERATURE in the global database would be checked. In this example JOE SMITH has a temperature of 100 °F, thus the statement portion of the IF-THEN rule is true and the action DIAGNOSIS IS FEVER is added to the global database.

This simple expert system would also need a few more rules in the knowledge base to deduce how severe the fever is, if the patient has a cold, or if the patient is healthy. The following rules could accomplish this:

2) IF $101\,^\circ F < \text{TEMPERATURE} \leq 103\,^\circ F$ THEN ASSERT "DIAGNOSIS IS MODERATE FEVER"

3) IF $\text{TEMPERATURE} \geq 103\,^\circ F$ THEN ASSERT "DIAGNOSIS IS SEVERE FEVER"

4) IF NOSE IS STUFFY AND COUGHING IS TRUE THEN ASSERT "DIAGNOSIS IS COLD"

5) IF NOSE IS STUFFY AND THROAT IS SORE THEN ASSERT "DIAGNOSIS IS COLD"

6) IF NO DIAGNOSIS EXISTS THEN ASSERT "DIAGNOSIS IS HEALTHY"

The inference engine must have a procedure to deal with missing information. For example, when the fourth rule is looked at, no value for NOSE or COUGHING is available in the global database. The expert system could ask the doctor to supply the necessary information. Thus, the questions "IS JOE SMITH'S NOSE STUFFY ?" and "IS JOE SMITH COUGHING ?" would be asked by the expert system. If the doctor replies "YES" to the first question, then the fact "NOSE IS STUFFY" would be entered into the global database; otherwise, the value "NOSE IS NOT STUFFY" could be entered into the database. This is not necessarily the best approach because it may be desirable to have values other than STUFFY and NOT STUFFY for NOSE (STUFFY, SORE, RED, etc.), but for this simple model it will work. Once the value for the NOSE and COUGHING conditions are known the inference engine could evaluate the appropriate IF-THEN rules.

In rule 6, the statement NO DIAGNOSIS EXISTS is interpreted by the inference engine to mean that no entry for DIAGNOSIS appears in the global database. Thus if rules 1 through 5 were not true, DIAGNOSIS would not appear in the global database, and rule 6 would be true, causing a value for DIAGNOSIS to be placed in the global database. The following problem would also need to be addressed: How does the inference engine know when to stop evaluating rules? It could evaluate all the rules once and then stop to report all the known facts to the user. However, it is generally better to loop through the rules in the knowledge base until none of the rules add anything new to the global database, or until a rule is satisfied that has an action part specifying that the expert system has made a diagnosis for this particular patient.

In this example, let us add two additional rules:

7) IF DIAGNOSIS EXISTS THEN REPORT DIAGNOSIS

8) IF DIAGNOSIS EXISTS THEN STOP

Now this set of eight rules will insure that the expert system will report at least one of the five possible states of a patient before stopping execution. Note, however, that this system assumes that the patient cannot have both a TEMPERATURE >= 100 'F and the symptoms for a cold. If this were the case, then the inference engine would first add DIAGNOSIS IS FEVER to the global database;

after the user enters the facts that NOSE IS STUFFY and COUGHING IS TRUE to the global database, the inference engine would replace DIAGNOSIS IS FEVER with DIAGNOSIS IS COLD. In order for the expert system to report the correct results to the user, it would have to save the previous FEVER diagnosis as well as DIAGNOSIS IS COLD. Then only could all occurrences of DIAGNOSIS that are in the global database of facts be reported.

The inference engine may come to conclusions about the specific facts using the rules in the knowledge base in several different ways. It could attempt to check each rule one at a time to see if the statements or conditions are true, and then follow the action described. This is known as forward chaining from the known facts to the final desired conclusion (in this example a value for DIAGNOSIS is the desired conclusion). Another method is to assume that one of the actions specifying a value for DIAGNOSIS is true and then try and show that all the statements for the IF-THEN rule are true. This backward chaining from possible conclusions to known facts is desirable if the number of possible conclusions is small and one wishes to find the first conclusion that is supported by the known facts. A combination of these two methods may be used in the hope that by proper application of both methods, the two lines of reasoning will meet somewhere in the middle, resulting in a speedier execution of the program.

A user should be able to question or quiz an expert system about its line of reasoning. This is very useful during the debugging phase of the design of an expert system. If the line of reasoning used by an expert system is not similar to the one commonly used by human experts, then it indicates either an incomplete knowledge base or inadequate inference engine. Expert systems will become more acceptable if the users can understand how the expert systems arrive at their conclusions. This ability of an expert system to explain its line of reasoning can also be used to provide training to students. For example, after the expert system reported that JOE SMITH had a fever, JOE SMITH wanted to know why he was diagnosed as having a fever. The expert system should have the capability to keep track of what rules created which facts in the global database. An acceptable answer will be:

The DIAGNOSIS IS FEVER because rule 1) IF 100°F $\leq$ TEMPERATURE $\leq$ 101°F THEN ASSERT "DIAGNOSIS IS MILD FEVER" was true. Note that this fact was entered by the user.

II. EXPERT SYSTEM APPLICATIONS

Expert systems have been used to diagnose, analyze, design, plan, learn, interpret signals, monitor, instruct, manage, create programs and understand images. Expert systems are particularly useful in domains where quantitative models are not available. Annex I lists more than 100 expert systems and shells, classified according to function and domain of use. This table (see p. 39) has been supplemented with expert systems from other sources. Uses of expert systems range from diagnosis and anaiysis to knowledge acquisition and learning. Annex I, under the category of "expert system construction", also lists expert system shells. A shell usually contains all the parts of an expert system except the knowledge base. This allows the user to build an expert system without having to create the inference engine, language interface, debugging tools, and explanation subsystem that are domain-independent parts of an expert system.

Unfortunately, promoters of expert system technology have created misconceptions about the capabilities of such systems. Expert systems do not have intuition or the ability to learn. So they do not solve novel problems in imaginative ways. However, expert systems can capture various rules of thumb that an experienced human expert uses to solve fairly routine problems in a specific domain or problem area. Since an expert system can solve a problem faster, it can help a specialist work more efficiently by enabling him to examine many alternative solutions. Similarly, it can also help a non-specialist learn problem-solving techniques in specific domains.

Expert systems development was initially limited to medicine, chemical synthesis, oil exploration, and other big money commercial ventures and research areas. This was due primarily to the cost of creating an expert system. The two main expenses incurred in creating an expert system are the computer hardware (memory, disks, cpu, etc.), and the computer software (both the inference engine and the knowledge base that must be acquired from human experts). The hardware costs have been falling dramatically over the years, so it is the software dominating the total cost of an expert system. Complex expert systems require months and even years to create the inference (control) structure and knowledge base. Two factors have been helpful in cutting these costs. The evolution of knowledge engineers who specialize in getting the information from human experts in the form of rules for the computer's knowledge base has cut down the time and effort needed. With the availability of low-cost expert system shells for personal computers (PCs) and advances in language interfaces, it is now possible to build expert systems for many areas of expertise where it was not economically feasible to do so before. This is not to say that expert systems are inexpensive to develop; it may take several man-years to develop sophisticated systems.

Expert systems have been useful in:

(a) Capturing the knowledge of human experts: The Campbell Soup Company implemented an expert system recently for trouble-shooting sterilizers - or cookers - in soup plants. If the problem in the cookers is not diagnosed quickly, the soup must be thrown out. Because Campbell's sterilizer expert was retiring with no replacement available, a 151-rule expert system was developed to capture his knowledge [WALL86]. The expert system is implemented on a Texas Instruments Professional Computer using Texas Instrument's Personal Consultant development system.

(b) Providing training in specialized fields: An expert system may be used as a teacher in domains with a shortage of human experts. Examples of this type of system include SOPHIE for electronic trouble shooting and STEAMER for steam propulsion plant operators [GEVA85]. By studying how an expert system makes its decisions, students can learn how to solve similar problems. Expert systems may also function as an aid to human experts by relieving them of monotonous tasks.

(c) Disseminating technical expertise in developing countries: A simple expert system is being tested in Tunisia and Egypt that assists paramedicals in rural villages. The expert system helps determine if a patient with an eye infection needs treatment, if treatment is necessary, and whether the patient can be treated locally or at a health center [DALY85].

(d) Monitoring and interpreting complex processes: The Westinghouse Electric Company has an expert system that monitors the performance of steam turbine generators [WALL86]. Called Gen-aid, the expert system monitors 250 sensor inputs continuously 24 hours a day for conditions that may signal a potential breakdown. Gen-aid monitors the generator conditions and proposes solutions when a problem occurs. Human operators cannot handle the large amounts of data being generated continuously in applications like this. Nuclear power plant monitoring is another important application area for expert systems.

(e) Building up and changing large databases: For example, The Digital Equipment Corporation is introducing new computer hardware constantly. To meet each customer's needs the best way possible, the knowledge base in R1 is updated continually as new hardware is introduced and older items become obsolete. R1 frees the salesman from having to memorize the changing variety of computer peripherals available for the different computer systems. The knowledge base and control structures are separate in expert systems, allowing for the easy addition and modification of rules in the knowledge base. As new developments and changes occur in the expert system's domain, appropriate changes to the knowledge base in the expert system can be made. These changes can allow for the best possible solution using the most up-to-date methods. By consulting many different human experts, it is possible to combine each individual's knowledge and come up with a more informed expert system than any of the experts' individual knowledge. Unfortunately, this task is complicated by possible disagreements among human experts on the proper way to diagnose problems.

One of the first expert systems was created in 1965 at Stanford University in California. DENDRAL was able to determine the molecular structure of an organic compound from data supplied by a mass spectrometer. DENDRAL actually has two rule-based subsystems [WINS84a]. One resides in the structure enumerator that generates all the feasible combinations of molecular structures that may make up the compound. It uses the spectrogram information to augment a must-have and must-not-have list of substructures for the unknown structure of the compound. These lists are used to limit the possible number of molecular structures from thousands to dozens. The second rule-based subsystem is located in the spectrogram synthesizer, which uses IF-THEN type rules to come up with the type and quantity of each possible decomposition product [WINS84a]. After synthetic spectrograms are produced, they are compared with the experimental spectrogram to obtain a final list of valid chemical structures. One of the main problems with DENDRAL was that the knowledge and inference mechanism were written together as one chunk of code. Thus, to modify or add a set of knowledge about chemical reactions was time-consuming and expensive.

DENDRAL was very successful. It produced analyses that have been published as original research results [RICH83]. This demonstrated that expert systems could function competently in a field made up of human experts. Often it is hard to compare the performance of an expert system with that of a human expert. Generally the results of an expert system are considered similar to those of a human expert if they agree on most of the important cases and the expert system does not make any costly mistakes. This problem is not unique to man-machine comparisons. In fact, in some fields the recommendations or conclusions of two different human experts can vary to the point of being opposing views even though both experts are given the same initial set of data.

III. CASE STUDIES

This section gives a brief description of expert systems in three different domains. MYCIN is a well known expert system for diagnosing infectious diseases. COMAX gives recommendations on growing and harvesting cotton crops. The final example describes DCLASS and PROPLAN, expert systems for creating manufacturing process plans from part descriptions.

A. MYCIN: Medical diagnosis of infectious diseases.

An important requirement for an expert system in the medical domain is that it be able to explain its recommendations. This is necessary to help doctors understand and accept the recommendations reached by the expert system. MYCIN was developed to "reason" in a way similar to humans so that it could explain its conclusions. A program called TEIRESIAS acts as the user interface with MYCIN. TEIRESIAS has two functions: First, it allows a knowledgeable user to add or modify data in the knowledge base. Second, it interacts with the user during the diagnosis of a patient. The user enters data about the patient's condition and can ask questions about MYCIN's reasoning by using TEIRESIAS.

A typical MYCIN production rule is as follows [NAU83]. The rules are written in the LISP programming language.

```
PREMISE ($AND (SAME CNTXT INFECT PRIMARY-
    BACTEREMIA)
        (MEMBF CNTXT SITE STERILESITES)
        (SAME CNTXT PORTAL GI))
ACTION  CONCLUDE CNTXT IDENT BACTEROIDES
    TALLY .7)
```

This means:

If (1) the infection is primary-bacteremia,
  (2) the site of the culture is one of the sterilesites,
    and
  (3) the suspected portal of entry of the organism is the
    gastro-intestinal tract.
then there is suggestive evidence (.7)
that the identity of the organism is bacteroides.

MYCIN is a rule-based system consisting of about 500 rules of the form:

IF <statements> THEN <actions>

The system starts by looking at a rule with some conclusion about an infectious disease. For this conclusion to be true, the premise or statements in the IF part of the rule must be true. Thus the system tries to find the values of these statements in the global database. If no value for a statement is known, MYCIN checks all the actions of the other rules to see if this statement can be determined by finding another set of statements to be true. When a value does not appear as a conclusion to any rule and the statement is a result of a test or value that the user may know, the system asks the user to supply this information. MYCIN stops after no further conclusions can be made from the currently known data. It then prescribes a treatment consisting of the smallest and safest set of drugs that will cover all diseases that are likely candidates for the known condition of the patient.

Because of the uncertainty involved in diagnosing diseases, MYCIN assigns a certainty factor to all the facts in the global database and to conclusions of rules. The certainty factor is a number between -1 and 1. A value of 1 indicates that the system is certain about the conclusion or fact being stated, whereas a value of -1 indicates that the value or statement is not true. Values in the interval [-1,1] indicate how certain the statement is.

MYCIN produces treatments similar to what a doctor would prescribe. Doctors do not always agree among themselves about the proper treatments for certain symptoms, but in 1977 MYCIN was in agreement with the medical experts 72% of the time [NAU83]. This value has been improved since then. Problems can arise from unexpected rule interaction in MYCIN, so care must be taken in formulation and ordering of the rules.

One of the most important features of MYCIN is its ability to give reasons why it prescribes certain drugs for the patient. This is possible because the system keeps track of the backward chaining done in going from the conclusions of a rule to the precedents that caused the conclusion to be asserted. This is helpful in the debugging stage. The following example shows how TEIRESIAS/MYCIN interacts with a user (doctor) [RICH83].

MYCIN: Is the patient's illness with ORGANISM-1 a hospital-acquired infection?

(ORGANISM-1 denotes unknown organism number 1 that MYCIN is trying to identify and treat. Note: It is possible that more than one organism is present in the patient. MYCIN can handle the presence of many different organisms in its diagnosis.)

USER: Why ?

(The user wishes to know why it is important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was in the hospital.)

MYCIN: [2.0] This will aid in determining the category of ORGANISM-1. It has already been established that:

[2.1] the gram stain of ORGANISM-1 is gramneg, and
[2.2] the morphology of ORGANISM-1 is rod, and
[2.3] the aerobicity of ORGANISM-1 is facultative
Therefore, if
[2.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized, then
there is strongly suggestive evidence (0.8) that the category of ORGANISM-1 is enterobacteriaceae
[ALSO: there is weakly suggestive evidence (0.1) that the identity of ORGANISM-1 is pseudomanas-aeruginosa] [RULE037]

USER: How [2.3] ?

(The user wants to know how MYCIN established that the aerobicity of ORGANISM-1 is facultative.)

MYCIN: The following were used:

[3.1] RULE027 indicated there is strongly suggestive evidence (0.8) that the aerobicity of ORGANISM-1 is facultative.
[3.2] RULE033 indicated there is weakly suggestive evidence (0.2) that the aerobicity of ORGANISM-1 is not facultative.
Since this gave a cumulative Certainty Factor of (0.6) for facultative, the evidence indicates that the aerobicity of ORGANISM-1 is facultative.

MYCIN showed that it could be a valuable assistant to doctors in diagnosing bacterial infections. However, MYCIN is limited to a narrow domain and suffers from severe performance degradation when faced with something outside of its narrow domain. If a patient is suffering from an allergic reaction to pollen, the system will prescribe tests for various bacterial infections. MYCIN does not have the knowledge to notice that the symptoms are due to an allergic reaction. An important spinoff from the MYCIN project was EMYCIN. EMYCIN (Empty MYCIN) contains only the control structure of MYCIN and none of the domain specific knowledge about bacterial infections. By adding an appropriate knowledge base for allergies to the EMYCIN shell, one would have an expert system that could diagnose allergic reactions instead of bacterial infections. Using EMYCIN would make the development task of this new expert system for allergies much easier.

B. COMAX: Cotton Crop Manager

Expert systems have been used in agricultural areas as well. One particular expert system is COMAX, used for cotton crop management. COMAX uses a cotton plant growth simulator program called GOSSYM, which simulates the growth of a cotton plant with respect to the local soil properties and daily weather conditions. COMAX determines the best strategy for irrigation, fertilization, and spraying of the cotton fields. COMAX is run everyday to re-evaluate its strategy based on the new weather conditions and weather history.

COMAX consists of about 50 rules, the inference engine about 6,000 lines of code, and GOSSYM approximately 3,000 lines of FORTRAN code [LEMM86]. COMAX was written in Common LISP on a Symbolics 3670 machine and then downloaded, unchanged, to a personal computer. Some sample facts and a rule used in COMAX are given below [LEMM86]:

```
FACTS
   (run-number 1)
   (hypothesized-weather hot-dry)
   (irrigation amount 1)
   (irrigation application-time 4)
    ...

RULE find-water-stress-day

IF
  (run-number ?number)
  (hypothesized-weather ?weather)
THEN
  (printout "Finding water stress day")
  (run-gossym ?number ?weather)
  (assert (set-hypothesized-irrigation)
```

The rule "find-water-stress-day" is one of the set of rules that determines the optimum irrigation schedule. The rule is true or fired whenever a value for run-number and hypothesized weather appears in the fact base. The symbols "?number" and "?weather" are variables that take on the values of run-number and hypothesized-weather in the data base. In this example, "?number" and "?weather" would take on the values of 1 and "hot-dry" respectively. When this IF-THEN rule is invoked, it runs the GOSSYM program to get the desired simulation results. The assert statement in the THEN part of the above rule is used to allow another rule "set-up-hypothesized-irrigation" to be true, so that it may be executed.

The simulation program GOSSYM is what gives COMAX most of its data for making decisions on the crop's water and fertilizer needs. GOSSYM was developed over a 12-year period with contributions from 10 scientists from four institutions. It simulates the growth and development of the entire cotton plant on an organ-by-organ basis: roots, stems, leaves, blooms, squares, and bolls [LEMM86]. The model must be given a detailed analysis of the hydrologic, fertility, and other properties of the soil as well as its water release curves and bulk density. After supplying the model with the initial soil data, the system is given daily weather information such as the minimum and maximum temperatures, rainfall, and solar radiation. Using this information, the simulation is run daily by COMAX to determine how and when the cotton plants should be watered or sprayed, if they need fertilizer, and when the cotton crop should be harvested. The simulation from emergence to harvest takes 60 to 90 minutes on an IBM PC with a math coprocessor. [LEMM86]

COMAX was tested on an actual 6,000-acre cotton farm. In July 1985, the system called for applying 50 pounds of nitrogen (fertilizer) per acre to increase the crop yield at harvesting time. The grower, who had not intended to use more fertilizer, applied an extra 20 pounds of nitrogen per acre, except on several test areas accounting for a total area of 6 acres. The resulting increase at harvest time was 115 pounds of cotton per acre where additional nitrogen was used. Based on the cost of \$11/acre cost of fertilizing the field and harvesting the extra cotton, this represents an increased profit of \$60 per acre. The increase in yield from this system resulted in approximately \$360,000 of additional profit. A microcomputer system and weather station hardware can be purchased for \$10,000 to \$13,000 not including the software. In this case, the system more than paid for itself due to the increase in yield.

C. DCLASS and PROPLAN: Process Planners

Process planning is an important stage in Computer Integrated Engineering (CIE). CIE refers to the use of computer technology to integrate the product design, specification, and production engineering functions [RICH86]. Process planning involves taking product specifications and producing a set of manufacturing plans. A process planner is typically a production engineer or an expert machinist with many years of experience. The task of process planning is complex and there is no guarantee of an optimal manufacturing plan being produced. An optimal process plan is defined as the least expensive plan in terms of time and money needed to produce the desired part. Computers are being used with increasing frequency in this domain to maintain consistency and to optimize the resulting manufacturing plans.

Several expert systems have been developed to automate the task of process planning. DCLASS is one such system, which was first used for sheet metal parts and later for a class of simple rotational parts [RICH86]. The system queries the user via selection menus about the part to be created. Information about the raw materials to be used for the part, its shape, number of holes, slots, and other part features are obtained in this manner. Similarly, other process information such as heat treatment and painting must be obtained to create a process plan for the part. The knowledge base in DCLASS is contained in hierarchical decision trees representing logical groupings of information. The decision trees are used as production rules by the forward chaining inference engine in DCLASS to produce the process plan for the part.

After eight months of development, a system for sheet metal parts was tested in a manufacturing environment. After a process plan was generated for a part, the human process planner was able to modify the plan if necessary. A productivity increase of 3:1 was realized in the sheet metal process planning effort itself

[RICH86]. Another benefit of the generative process approach was an increase in the consistency of the quality of the process plans. In several instances, design flaws or mistakes in the design specifications were made apparent because it called for an operation that did not appear on the selection menu presented to the user. For example, the painting of an aluminum part would not appear as a valid choice in the user's display menu [RICH86].

Encouraged by these results, the next step was to create a system for a small subset of simple rotational parts. The increase in the number of special features such as traverse holes, bevels, chamfers, threads, countersinks, counterbores, spherical ends, tapers, and knurls was staggering. So many questions were asked of the users that they became annoyed and frustrated. The number of parameters associated with more complex parts and their production made further attempts at generative process planning with DCLASS unfeasible. Another problem encountered was the development and maintenance of the complex decision logic used in DCLASS. Changes in the factory environment or class of parts that can be addressed by the system have to be reflected in the hierarchical decision structure. It takes time and careful thought to change the decision-making logic in such a way that the validity of the resulting process plans is ensured.

Although DCLASS was found to be unfeasible for simple rotational parts, another expert system, PROPLAN, has proved to be less demanding on the users. PROPLAN is a knowledge-based approach to generative process planning that creates process plans from Computer Aided Design (CAD) data to be used in a Computer Aided Manufacturing (CAM) environment. This system derives the necessary part specifications from a CAD database containing the part. PROPLAN is capable of analyzing the part geometry in its internal symbolic form to come up with a logical sequence of operations to be performed on some raw material to produce the final result. The resulting process plan specifies what machines and tools (drill bits, cutters) should be used, what type of coolant is needed, and the speed, feed, and depth of cut for each operation. PROPLAN is capable of explaining the line of reasoning that it used to come up with the final process plan. This is helpful for debugging and developing the expert system and knowledge base. The expert system's rules can be changed to accommodate different machinery and different types of tools.

Examples of PROPLAN Production rules [PHIL85] are:

(a) On choice of machine:

IF <OPERATION is TURNING>
and <Maximum LENGTH of PART is less than 30 inches>
and <Maximum DIAMETER of PART is less than 10 inches>
THEN <recommended MACHINE is LATHE001>

(b) On choice of tool:

IF <OPERATION is DRILLING>
and <PART-MATERIAL is HIGH-CARBON-STEEL>
and <DIAMETER of HOLE is less than 2 inches>
and <HARDNESS of PART-MATERIAL is between 200 & 2400 BHN>
THEN <recommend DRILL with HELIX 240 degrees>
or <recommend DRILL with POINT 118 degrees>

(c) On choice of tool-material:

IF <the PART-MATERIAL is ALUMINUM>
and <OPERATION is TURNING>
THEN <recommended TOOL-MATERIAL is HIGH-SPEED-STEEL>

(d) On choice of coolant:

IF <the PART-MATERIAL IS PLAIN-CARBON-STEEL>
and <TOOL-MATERIAL is HIGH-SPEED-STEEL>
THEN <recommended COOLANT is SOLUBLE-OIL>

(e) On choice of cutting speed:

IF <OPERATION is TURNING>
and <DEPTH-OF-MATERIAL to be removed is greater than 2 inches>
and <DIAMETER of PART is less than 4 inches>
and <PART-MATERIAL is CASTIRON>

and <TOOL-MATERIAL is CARBIDE-TIP>
THEN <recommended SPEED is 250 rpm>

Process plans were developed for several parts and then compared with the process plans currently being used in a factory environment. The plans were similar. However, some plans differed because of local constraints existing in the factory that were not accounted for in the expert system. The computer-generated process plans had a more consistent level of detail than those from the human process planners. This is desirable to make sure the resulting product meets the minimum standards. The results showed that the PROPLAN system is efficient and has the potential for practical application [PHIL85].

IV. DEVELOPMENT OF AN EXPERT SYSTEM

The development of expert systems involve symbolic and list manipulations. Therefore, expert systems are not usually written in FORTRAN, Pascal, BASIC, or COBOL programming languages. The LISP language is tailored for symbolic manipulation, list representation and associative memory retrieval. Writing an expert system from scratch is very time-consuming. However, the user interface and inference engine of an expert system do not need to be written for each new domain because they are commercially available. These expert system tools are referred to as expert system shells. Knowledge and rules for a specific domain can be combined with an expert system shell to produce an expert system.

In general, the development of an expert system is not a trivial task. A toy expert system, as discussed in Chapter I(B), can be written fairly easily as a rule-based system. However, more complex expert systems involve many heuristics and rules of thumb and thus require more development time. The development of an expert system requires both a domain expert and a knowledge engineer. Domain experts are knowledgeable about the area of expertise being encoded into an expert system. Knowledge engineers are familiar with ways of representing knowledge, artificial intelligence techniques, and expert system construction.

The cost of developing an expert system varies from a few thousand dollars for simple systems to several hundred thousand dollars for large projects. Hardware costs range from $1,000 for a personal computer to about $100,000 for specialized LISP work stations. Software costs include the cost of LISP or Prolog languages and/or expert system shells. Depending on the domain it may take several man-months to several man-years of time to develop a complete expert system.

Expert systems are developed in six more or less independent phases [HARM85]:

(a) Selection of an appropriate problem;

(b) Development of a prototype system;

(c) Development of a complete expert system;

(d) Evaluation of the system;

(e) Integration of the system; and

(f) Maintenance of the system.

A. Selection of an Appropriate Problem

Expert systems are not appropriate for all problem areas. If the domain consists primarily of numerical attributes with stable rules of thumb and heuristics, then it may be more appropriate to use a spreadsheet, database, or other conventional programs to solve the problem. Solutions requiring considerable background information or common sense knowledge may be too unwieldy and time-consuming to encode into an expert system to obtain the desired performance. One should not try to apply the expert system technology to situations where there is no previous experience.

Candidate problems to be solved using expert systems usually involve knowledge that is subjective, changing, symbolic or partly judgemental [HARM85]. By encapsulating the knowledge in a set of IF-THEN rules it is possible to add, delete, or change knowledge (rules) by simply modifying the rule base. Uncertainties in the original data and in rule conclusions can be accounted for by using certainty factors and a method for propagating certainty values. Symbolic data can be manipulated by IF-THEN rules that check for certain symbolic values and/or assert symbolic facts in the database.

Care must be taken to ensure that the benefits of the proposed expert system will outweigh the required man-months or years of development work. If after a long development time the targeted end users of the system do not use the system as expected, then developing an expert system may be a waste of time. Potential users may mistrust, fear, or be skeptical of the expert system. Consideration of how the domain experts feel about imparting their knowledge to an expert system is also important. The ultimate goal should be to enhance the expert's job performance, train students, provide help to the expert in tiresome tasks, or to fill a demand for the expert's domain knowledge. It is essential that the domain expert be able and willing to work with the knowledge engineer in creating a viable end product.

B. Development of a Prototype System

After an appropriate domain has been selected for the expert system, the next step is to develop a prototype system. Typically, the knowledge engineer questions the domain expert about the expert's problem-solving techniques. Next the domain expert runs through several typical example problems and shows how to solve them. The knowledge engineer must determine an appropriate data representation and control structure that will allow the expert system to represent the domain expert's facts and rules of thumb used in solving the problems.

Both the knowledge engineer and domain expert must specify exactly what performance they expect from the expert system. The desired performance must be defined in advance so that this can serve as a goal during the development stage. Typical goals are having the expert system produce solutions or recommendations similar to the human expert in certain predefined cases under the operating conditions of a human expert. When these goals are met, the knowledge engineer will have completed the task successfully.

Once a knowledge representation scheme and control structure are found that can meet the predefined goals, the knowledge engineer can choose an expert system shell or tool to be used in creating the final expert system. Using a commercially available shell saves a considerable amount of time. Next the knowledge and rules of thumb from the domain are encoded into the prototype system. The knowledge engineer and domain expert test the prototype with sample cases. By following the expert system's use of the IF-THEN rules, they can determine whether the machine is solving the problem incorrectly. If so, then the rules can be modified to obtain the desired results.

The main purpose of the prototype stage is to determine whether the knowledge representation and expert system shell selected by the knowledge engineer are adequate for the desired system. It may be necessary to start the prototype stage over again using a different expert system shell and/or knowledge representation scheme if the prototype performs badly. During the creation of the prototype system the domain expert becomes familiar with how the domain knowledge and rules of thumb are used by the expert system to make its conclusions. This enables the expert to formulate the domain knowledge into a form better suited for creating the complete expert system in the next phase.

C. Development of a Complete Expert System

Once a successful prototype system has been developed, it is time to start developing the complete expert system. Generally it is best to throw away the prototype, since the prototype often contains many special problem fixes that are awkward or inefficient. The knowledge engineer usually finds that a slightly different data representation performs better in the larger complete expert system than the one used in the prototype. Starting the final version of the expert system from scratch allows fine-tuning of the data representation and better coding of the rule base for the final product. In addition to allowing for more general cases, the complete expert system should have rules that take care of special cases and provide better coverage of the problem area being tackled than does the smaller prototype. As before, the system performance must be monitored and the rule base debugged when incorrect solutions are given.

D. Evaluation of the Expert System

Once the complete expert system meets the original design requirements, it is time to evaluate the system extensively. Other domain experts are called upon to test the system with both general and special cases to see if the system performs well. It is important that the domain experts agree that the system is performing well. If the experts feel that the system gives bad solutions or advice, then the system may have to be redesigned. Minor flaws in the expert system or special cases that the expert system cannot correctly solve can be fixed at this time. When the domain experts feel that the system does an adequate job, then it can be integrated into the work environment.

E. Integration of the Expert System

Integrating a system into its work environment involves interfacing the program with outside data sources (e.g., measurement systems and the user) and enhancing the user-friendliness of the system. The end users of an expert system usually know very little about computers. Thus, the data entry and user interaction should be as simple and user friendly as possible. The use of selection menus and specific user prompts can provide users with enough information to make the user interaction easy to handle.

PUFF, a pulmonary diagnosis system, is an example of an expert system that is well integrated into its environment. After PUFF was accepted as performing well, it was recoded from Lisp to Basic and then transferred to a Digital

Equipment Corp. PDP-11 Computer already being used in a hospital [HARM85]. To use the system, a patient breathes into the pulmonary machine, the data from this machine is passed to the PDP-11 computer where PUFF analyzes the data, reports the patient's condition, and recommends a possible treatment. The physician does not have to interact with PUFF at all.

F. Maintenance of the Expert System

Maintenance is the final phase of developing an expert system. This phase is often ongoing for domains where new data or knowledge about the domain is being acquired. This is especially true of applications involving changing technical fields such as medicine or machine repair (new models every year). Maintenance includes adapting the system to changing environments where new diagnostic machines or new computers become available.

V. COMMERCIALLY AVAILABLE EXPERT SYSTEMS

A large variety of expert systems, expert system shells, and development tools are commercially available. This section lists some companies that supply these products, with a brief description of one or more products available from each company. More information can be obtained by directly writing to the companies. Because of the substantial interest in expert systems, new companies are being formed every year in the United States and Western Europe.

1. Radian Corporation
   RuleMaster
   8501 Mo-Pac Blvd.
   P.O. Box 9948
   Austin, Texas 78766

   Phone: (512) 454-4797

RuleMaster - For developing large, practical expert systems. Can be used on most computers with a "C" compiler.

2. Micro Data Base Systems, Inc.
   Marketing & Sales
   P.O. Box 248
   Lafayette, IN 47902

   Phone: (317) 463-2581

GURU - An expert system designed especially for business. Considers uncertainties, asks for more information when needed, and explains its recommendations. Available on a personal computer.

3. Logicware Inc.
   Suite 3000 West Tower
   5000 Birch Street
   Newport Beach, CA 92660

   Phone: (714) 476-3634

MPROLOG - Contains a powerful inference engine that provides automatic, system-driven reasoning with the rules and facts in the programmed knowledge base. The same application can be run on a mainframe or a personal computer.

4. Artelligence, Inc.
   14902 Preston Road
   Suite 212-252
   Dallas, Texas 75240

   Phone: (214) 437-0361

OPS5 - Commonly used language in expert system development. It is available on SUN and APOLLO systems as well as IBM-PC, XT, and AT's.

5. Software Architecture and Engineering, Inc.
   1500 Wilson Blvd.
   Suite 800
   Arlington, VA 22209

   Phone: (703) 276-7910

KES II - A knowledge engineering system written in the C programming language. May be used on many computers including IBM PC.

6. Silogic, Inc.
   6420 Wilshire Blvd.
   Suite 2000
   Los Angeles, CA 90048

   Phone: (213) 653-6470

The Knowledge WorkBench - Supports a variety of natural language applications, expert data bases, and other knowledge-based applications. Available on personal computers.

7. Intelliware, Inc.
   4676 Admiralty Way
   Suite 401
   Marina del Rey, CA 90291

   Phone: (213) 305-9391

Expertteach - A comprehensive guide to Expert System Technology consisting of Expert System tutorials, case studies, on-line teaching programs, Expert System building tools with source code and Artificial Intelligence languages for use on IBM Personal Computers.

8. ExperTelligence, Inc.
   559 San Ysidro Road
   Santa Barbara, CA 93108

   Phone: (805) 969-7871

ExperFacts - A flexible expert systems building tool. Written in ExperLisp for Macintosh Personal Computers.

9. EXSYS, Inc.
P.O. Box 75158
Contr. Sta. 14
Albuquerque, NM 87194

Phone: (505) 836-6676

EXSYS - An expert system development tool for the IBM PC, XT, AT and compatibles. Uses IF-THEN rules and is written in C. About 5000 rules can be run in a PC with 640k. A low cost $15 demonstration system is available from the company.

10. KDS Corporation
934 Hunter Road
Wilmette, IL 60091

Phone: (312) 251-2621

KDS - Expert system development software which runs on an IBM-PC or PC compatible computer. KDS is given case histories and from these case histories it figures out which rules to apply automatically.

11. Texas Instruments
Phone: (800) 527-3500 U.S.
(416) 884-9181 Canada

Personal Consultant - Expert systems development software for TI and IBM Personal Computers.

VI. POTENTIALITIES FOR DEVELOPING COUNTRIES

Expert systems offer tremendous possibilities for developing countries. A recent Symposium on "Microcomputers for Developing Countries", held in Lisbon, Portugal, and organized by Board on Science and Technology for International Development, National Research Council, Washington D.C., USA, identified the following domains for applying expert system technology in developing countries:

(a) Health (Diagnosis, Simulation, Health Management)
(b) Agriculture
(c) Electricity Power Control
(d) Water Resources
(e) Real-Time Manufacturing Control
(f) Economic Analysis and Planning Tools
(g) Equipment Repair or Fault Diagnosis

It is important to keep in mind that the problem domains of interest and the corresponding knowledge base in developing countries are different from those in the developed countries. For example, an expert system software for fault-diagnosis in electric motors designed in a developed country may not work in a typical developing country environment. Voltage fluctuations, temperature, and humidity conditions severely affect the performance of electric motors. The repair strategy of the expert system should also take into account the shortage of critical spare parts in developing countries. Therefore, one should not try to apply the expert system technology to situations where there is no previous experience. The knowledge base for the expert system has to be developed in the environment where it will be used. Further, an expert system should never be used in a problem outside its domain.

Scientists, engineers, planners, and designers in developing countries should be made aware of the potential of expert systems. After all, an expert system offers an alternative and, often, a more efficient and economical method for monitoring, diagnosing, designing, problem-solving or trouble-shooting. Expert systems are particulary useful for problems involving scheduling (machines, people, trucks), diagnosis (patients, engine trouble) and process control (power plant, mining operation). "Expert systems" has become a technological buzz-word. There are misconceptions associated with expert systems, namely, that they can solve novel problems in an imaginative way. It is a human expert who provides the rules or knowledge base that an expert system follows. Since human experts are in short supply in many developing countries, it makes sense to build expert systems and place them in these countries' remote areas. An expert system can even help a specialist explore many more options, and can be used for educational and training purposes. The cost of computer hardware has come down to such a level that a PC-based expert system can be made available for less than $2,000.

Food production is vitally important to developing countries. Thus, expert system applications in agriculture should interest scientists in developing countries. Whittaker [WHI86a] classifies various applications in the agriculture domain into one of the following three categories:

(a) planning (e.g., pest management, fertilizer applications, facilities selection and design);

(b) diagnostics (e.g., crop problems, equipment problems, animal health); and

(c) management (e.g., resource conservation, marketing).

Three expert systems that have been developed in the agriculture domain are described below. Similar systems can be developed for other domains in developing countries. For instance, an expert system to select a mainframe computer for a state agency can be developed along the lines of the timber harvesting equipment selection in the first example.

(a) <u>Expert system for timber harvest equipment selection.</u>

Gibson and others [GIB86b] have developed an expert system that assists a manager to select timber harvesting equipment for various environmental and site conditions. This expert system receives input from the user about the slope, soil structure, concern for erosion and post-harvest aesthetics, crew skills, and road limitations. Based on this information, the expert system recommends whether an "overhead" harvesting system or a "ground" based harvesting system will be needed. This expert system also responds to the "how" and "why" queries of the user. It is these types of queries that make an expert system useful for training purposes. The timber harvesting equipment selection expert system has 63 rules and is running on an IBM PC using Texas Instruments Personal Consultant IQ-LISP-based shell. Other applications of expert system technology in the forestry domain include forest road layout, wildfire suppression prescribed burning, insect identification control, herbicide and pesticide application and integrated decision support [MILS86], [THIE86], [GIB86a].

(b) Grain Combine expert system.

Schueller and others [SCHU85] have developed an expert system for the adjustment, repair and maintenance of grain combines. A suboptimum performance of a grain combine can lead to grain loss, grain damage, and harvesting delays. This PC-based expert system determines problem causes and suggests potential corrective actions. While an operator can look through the manuals to find the solution to the problems, an expert system reduces the time taken to diagnose the problems and to find the maintenance schedule and the information about dealers and parts suppliers. If the farm is located in a remote area, a trained mechanic may not be readily available, and the expert system may also assist the farmer to fix the problem. A useful feature of this expert system is that it uses a speech synthesizer to present spoken queries to the user. As a future enhancement, the system will have the speech recognition ability to accept speech input so that the user does not have to enter the information through a keyboard.

(c) Expert system for soil erosion modeling.

A computer model for soil erosion is useful for studying the effect of factors like runoff, soil detachment, and soil deposition on long-term average erosion rates. Formal mathematical models are based on some simplifying assumptions and are valid only for a range of conditions. These models are difficult to use in situations where the information or the measurements are non-numeric (symbolic, e.g., temperature is "high" versus temperature equals 56 °F), incomplete, unreliable or inconsistent. To overcome these problems, Whittaker and others [WHI86b] are developing an expert system to model the soil erosion.

Since shortage of electric power is a common and serious ailment of many developing countries, efficient operation and safety of power generating units is critical. The need for trained technical persons also exceeds the supply in these countries. Operators are having to make more difficult and frequent decisions because of the increased complexity of power plants. Thus, expert systems can be very useful for the operation, control, safety, reliability and efficiency of power plants. The Instituto de Investigaciones Eléctricas, Mexico, is developing several expert systems for the diagnosis and operation of power plants [CAST86], [ROD86a], [ROD86b]. These systems will find use in other developing countries as well.

Expert systems can be utilized in all aspects of the manufacturing industry in developing countries. For developing countries to be more competitive in the international market, they must manufacture goods of high quality at lower costs. Expert systems can be used during the product design process and can assist in intelligent manufacturing [RYCH84], [BENA85]. Product design process involves expertise about a product and its intended use. Expert systems can be used to get the requirements directly, taking an order that is complete and manufacturable, and placing an order into production planning that can be built. Inventory management, capacity planning, and floor-loading throughput management are critical issues during manufacturing that can be handled by expert systems.

In summary, developing countries should investigate the applicability of expert systems to some of their problems. They should make an effort to build a critical mass of researchers and computing facilities. It is still too early to tell which applications of expert systems will prove to be most useful. But expert systems are receiving substantial attention in the developed countries and are being used in several manufacturing, decision-making, and financial domains. There should be no fear that expert systems will replace human experts. Human experts will be needed to update the systems' knowledge base as additional information becomes available.

ANNEX I : EXISTING EXPERT SYSTEMS BY FUNCTION

Domain	Name of system	Institution where developed
1. Diagnosis		
Medicine	PIP	MIT
Medicine	CASNET	Rutgers Univ.
Medicine	Internist/Caduceus	Univ. of Pittsburgh
Medicine	MYCIN	Stanford Univ.
Medicine	PUFF	Stanford Univ.
Medicine	MDX	Ohio State Univ.
Computer faults	DART	Stanford Univ./IBM
Computer faults	IDT	Digital Equipment Corp.
Nuclear reactor accidents	REACTOR	EG&G Idaho, Inc.
Telephone lines	WAVE	Bell Labs
Diesel locomotives	CATS	GE Corp. [WALL86]
Multiple pair/telephone wires	ACE	AT&T Bell Labs [WALL86]
Continuous Process Diagnosis	PDS (shell)	GE Corp. [WALL86]
Diagnostic Shell from CATS	GEN-X (shell)	CMU and Westinghouse [WALL86]
Telephone switching system	Compass	GTE Corp. [WALL86]
Plant diseases	AQ11	[NAU83]
2. Data Analysis and interpretation		
Oil well logs	Dip-Meter Advisor	MIT/Schlumberger
Petroleum	LITHO	[BONN86]
Petroleum	DRILLING ADVISOR	[BONN86]
Chemistry	DENDRAL	Stanford Univ.
Chemistry	GA1	Stanford Univ.
Geology	Prospector	SRI
Protein crystallography	Crysalis	Stanford Univ.
Causal relations in medicine	RX	Stanford Univ.
Causal relations in medicine	ABEL	MIT
Oil well logs	ELAS	AMOCO
Steam turbine generators	Gen-aid	Westinghouse Elec.[Wall86]
3. Analysis		
Electrical circuits	EL	MIT
Symbolic mathematics	MACSYMA	MIT
Symbolic mathematics	REDUCE	Stanford [BONN86]
Symbolic mathematics	SNARK	[BONN86]
Mechanics problems	MECHO	Univ. of Edinburgh
Naval task force threats	TECH	Rand/Naval Ocean Systems
Earthquake damage of structures	SPERIL	Purdue Univ.
Digital circuits	CRITTER	Rutgers Univ.
4. Design		
Computer system configurations	R1/XCON	CMU, DEC Corp
Circuit synthesis	SYN	MIT
Chemical synthesis	SYNCHEM	State Univ. of NY

..continued

ANNEX I : continued

Domain	Name of system	Institution where developed
5. Planning		
Machining of mechanical parts	GARI	[BONN86]
Chemical synthesis	SECHS	U. of Calif. Santa Cruz
Robotics	NOAH	SRI
Robotics	ABSTRIPS	SRI
Planetary flybys	DEVISER	JPL
Errand planning	OP-PLANNER	Rand
Molecular genetics	MOLGEN	Stanford Univ.
Mission planning	KNOBS	MITRE
Job shop scheduling	ISIS-II	CMU
Molecular genetic experiments	SPEX	Stanford Univ.
Medical diagnosis	HODGKINS	MIT
Naval aircraft operations	AIRPLAN	CMU
Tactical Targeting	TATR	Rand
Process planning	Proplan	U. of Illinois [PHIL85]
6. Learning from experience		
Chemistry	META-DENDRAL	Stanford Univ.
Heurisitics	EURISKO	Stanford Univ.
7. Concept formation		
Mathematics	AM	CMU
8. Signal interpretation		
Speech understanding	Hearsay II	CMU
Speech understanding	HARPY	CMU
Machine acoustics	SU/X	Stanford Univ.
Ocean surveillance	HASP	System Controls, Inc.
Sensors on board naval vessels	STAMMER-2	Naval Ocean Systems
Heart performance	ALVEN	Univ. of Toronto
Military situations	ANALYST	MITRE
9. Monitoring		
Patient respiration	VM	Stanford Univ.
10. Use Advisor		
Structural analysis	SACON	Stanford Univ.

..continued

ANNEX I : continued

Domain	Name of system	Institution where developed
11. Computer-aided instruction		
Electronic troubleshooting	SOPHIE	Bolt, Beranek and Newman
Medical diagnosis	GUIDON	Stanford Univ.
Mathematics	EXCHECK	Stanford Univ.
Steam propulsion plant operatn.	STEAMER	Bolt, Beranek and Newman
Diagnostic skills	BUGGY	Bolt, Beranek and Newman
Causes of rainfall	WHY	Bolt, Beranek and Newman
Coaching of a game	WEST	Bolt, Beranek and Newman
Coaching of a game	WUMPUS	MIT
Coaching of a game	SCHOLAR	Bolt, Beranek and Newman
12. Knowledge acquisition		
Medical diagnosis	TEIRESIAS	Stanford Univ.
Medical consultation	EXPERT	Rutgers Univ.
Geology	KAS	SRI
13. Expert system construction		
	ROSIE	Rand
	AGE	Stanford Univ.
	Hearsay III	Univ. of So. Calif. / ISI
	EMYCIN	Stanford Univ.
	OPS 5	Carnegie-Mellon Univ.
	YAPS	Maryland Software Distr.[CROS86]
	LOOPS	XEROX [ERIC84]
	KEE	XEROX [CROS86]
	ARS	[NAU83]
	YES/MVS	IBM
	RuleMaster	Radian Corporation [CROS86]
	Expert Ease	Human Edge Software [MICH85]
	KS 300	Teknowledge Inc. [MICH85]
	KES	Intelligenetics Inc. [MICH85]
	Personal Consultant	TI Inc. Dallas Texas [MICH85]
	AL/X	Univ. of Edinburgh [MICH85]
Medical diagnosis	KMS	Univ. of Maryland
Medical consultation	EXPERT	Rutgers Univ.
Medical expert systems work	PORTAL	Univ. of Victoria [CROS86]
Electronic systems diagnosis	ARBY	Smart Systems Technology
Medical consult. with time data	MECS-AI	Tokyo Univ.
Process planning	DCLASS	Texas Instr. [RICH86]
14. Consultation/intelligent assistant		
Battlefield weapons assignments	BATTLE	NRL AI Lab
Medicine	Digitalis Therapy Adv.	MIT
Radiology	RAYDEX	Rutgers Univ.
Computer sales	XCEL	CMU / DEC Corp.
Medical treatment	ONCOCIN	Stanford Univ.
Nuclear power plants	CSA Model-Based NPP Con	Georgia Tech.
Diagnostic prompting in med.	RECONSIDER	U. of Calif San Fran.

.. continued

ANNEX I : continued

Domain	Name of system	Institution where developed
15. Management		
Automated factory	IMS	CMU
Project management	CALLISTO	DEC Corp.
Cotton crop management	COMAX	USDA Agricultural Research Service [LEMM86]
16. Automatic programming		
Modeling of oil well logs	NIX	Schlumberger-Doll Res.
	CHI	Kestrel Inst.
	PECOS	Stanford Univ.
	LIBRA	Stanford Univ.
	SAFE	U. of So. Calif. / ISI
	DEDALUS	SRI
	Programmer's Apprentice	MIT
	PSI	[BONN86]
17. Image understanding		
	VISIONS	U. of Massachussetts
	ACRONYM	Stanford Univ.
	MAPS	CMU (ETL expanded) [MOON83]

Source: W.B. Gevarter, Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics (Prentice Hall Inc., New Jersey, 1985), pp. 46-66.

ANNEX II: PROJECT PROPOSAL FOR AN EXPERT SYSTEM FOR TRANSFER OF TECHNOLOGY AGREEMENTS

Part One - Basic Data

COUNTRY: Mexico

PROJECT TITLE: Issues in the Design of an Expert System for Transfer of Technology Agreements.

STARTING DATE: 1 August 1987

COMPLETION DATE: 31 December 1987

UNIDO FUNDS: US $20,000

Part Two

I. Objectives.

A. Development Objectives

To strengthen the capabilities of Instituto de Investigaciones Eléctricas (IIE), Mexico, in the area of design and applications of expert systems.

B. Immediate Objectives

1. To establish whether expert systems technology is useful for teaching contract negotiation skills to engineers, managers and lawyers in developing countries.

2. To investigate practical issues and

constraints (e.g., number of man-years, cost, data base, domain) for developing such an expert system.

II. Background and justification.

The industrial sector of virtually every developing country relies on the transfer of technology from developed countries. A substantial portion of the foreign exchange balance of a developing country is spent on acquiring this technology. Thus, the choice and the terms of contract of technology transfer are important to the economic well-being of a developing country. Negotiators in a developing country need information about the technology they are buying and need to develop negotiating skills. For information, they are often at the mercy of consultants and multinationals. A user-friendly, computerized data base can provide the information in a timely fashion. Multinationals are more interested in selling turnkey systems. A successful negotiator is one who can negotiate a contract with the maximum amount of local-content.

Contract negotiation is an art. It is not easy to specify which negotiating skills are essential, let alone figuring out how one acquires them. Personality traits of negotiators is very important in the outcome. The negotiation strategy depends on the industrial sector. Developing countries have an upper hand in the negotiation of conventional technology, but developed countries dominate the discussion for high technologies. Most of the time the design technology is available but the manufacturing technology is lacking in the developing country. Therefore, it is important to check that the "package deal" does not contain redundant and obsolete technologies.

Before initiating contract negotiations, the core of technology that is not available in the home country should be identified. Suppose machining capability to manufacture turbines is not available. Then instead of going to, say, Westinghouse (a manufacturer of turbines), the developing country should contact, say, Cincinnati Milicron (a manufacturer of machinery). Going to the right source for acquiring the technology is essential. Large U.S. companies do not have the expertise to manufacture low-volume production. Many companies in developed countries cannot tailor their technologies to meet the requirements of developing countries. For example, IIE had to approach a Spanish company to get fossil fuel generators converted for use in geothermal power generation.

UNIDO has a long-standing interest in establishing some common guidelines for evaluation of transfer of technology agreements [UNID79]. It has a commitment to provide practical information to negotiators in developing countries in preparing and negotiating technology transfer contracts. This report and the attached project proposal is an outcome of this commitment to explore the use of the emerging technology of expert systems in the preparation of these contracts. An expert system can verify if the technology transfer contracts have the right pro forma, if they conform with national policy, and if the benefits of acquiring technology exceed the cost of buying it.

III. Project outputs.

A report with proposals for action regarding the development of expert systems for transfer of technology negotiations will be developed. The report will evaluate the need for UNIDO to invest any additional resources to develop such a system. In addition to the expert systems, the report will also investigate the benefits of a CAI (Computer-Aided Instruction) system to assist developing countries in these negotiations. This could be a "how-to" guide or manual with a section for each country. Often, developing countries are at the mercy of consultants and multinationals about various technology options, so a CAI system containing a profile of each industry is desirable.

IV. Project activities.

Development of an expert system in a new domain is not easy. Contract negotiation is an art and it is difficult to establish how one acquires these skills. Negotiation skills and strategy used depends upon the specific technology. The staff at the IIE has substantial experience in importing technology from developed countries as well as in exporting technology to less developed countries. A consultant in expert systems will stay at the institute for one month to understand the contract negotiation process and to capture the sector knowledge, strategy, and national policies utilized. The consultant will also work closely with the UNIDO staff to identify UNIDO's role in these negotiations. The implementation schedule is as follows:

Phase	Duration
I. Major areas of technology transfer agreements	2 weeks
II. Identify a few technological areas for detailed study	1 week
III. Legal requirements of contracts	2 weeks
IV. Economic, political, and technological constraints	4 weeks
V. Knowledge data base requirements	4 weeks
VI. Hardware, software, and manpower requirements	4 weeks
VII. Report and recommendations	3 weeks

V. Project inputs.

A. IIE Inputs

As mentioned before, IIE has substantial expertise in technology transfer negotiations. It has a separate legal department to handle these negotiations, whose expertise is invaluable for this feasibility study. IIE has also expressed interest in participating in this study.

B. UNIDO input

1. An expert at the cost of US$ 20,000
2. Headquarters staff will provide information to the expert regarding UNIDO's role in technology transfer negotiations.

VI. Envisaged follow-up.

Any follow-up will depend upon the recommendations of this study, which could take the form of designing a small prototype expert system or simply designing a simulation game for negotiating a contract.

BIBLIOGRAPHY

[ADDI85] T. R. Addis, Designing Knowledge-Based Systems, Prentice-Hall Inc., New Jersey, 1985.

[ANTO86] A. A. Arroyo, Artificial Intelligence and Expert Systems, IEEE Potentials, Oct. 1986, pp. 16-18.

[BARR82] B. Barr, A. and Feigenbaum, E. A., The Handbook of Artificial Intelligence, vol. 2, Los Altos, CA, W. Kaufmann, 1982.

[BENA85] R. Benares-Alcantara et al., Knowledge-Based Expert System for Computer Aided Design,C. E. P., September 1985, pp. 25-30.

[BERN80] R. Bernhard, An Electronic Advisor/Companion, IEEE spectrum, Sept. 1980, pp. 39-43.

[BONN86] A. Bonnet, Artificial Intelligence, Prentice-Hall Inc., New Jersey, 1985, pp. 141-186.

[BROW86] D. C. Brown, B. Chandrasekaran, Knowledge and Control for a Mechanical Design Expert System, IEEE Computer, July 1986, pp. 92-100.

[BRUN86] G. Bruno, A. Elia, and P. Laface, A Rule-Based System to Schedule Production, IEEE Computer, July 1986, pp. 32-40.

[BUCH82] B. Buchanan, and R. Duda, Principles of Rule-Based Expert Systems, Heuristic Programming Project Report HPP 82-14, Dept. of Computer Science, Stanford Univ., Stanford, CA, Aug. 1982.

[CAST86] L. H. Castillo et al., A Rule-to-Network Based Expert System Shell for Power System Knowledge Engineering, Proceedings of IASTEP (Power High Tech 1986), Montana State University.

[CHIE86] Y. Chien and J. Liebowitz, Expert Systems in the SDI Environment, IEEE Computer, July 1986, pp. 115-122.

[CROS86] G. R. Cross, Tools for Constructing Knowledge-Based Systems, Optical Engineering, March 1986, pp. 436-444.

[DALY85] J. Daly, Population Program Applications of "Expert Systems", Poplac News, 1985, pp. 3-14.

[DENN86] P. J. Denning, The Science of Computing: Expert Systems, American Scientist, Vol. 74, Jan./Feb. 1986, pp. 18-20.

[DEWI85] P. Elmer-Dewitt, How to Clone an Expert, Time, Sept. 2, 1985, page 44.

[DUDA81] R. Duda, Knowledge-Based Expert Systems Come of Age, BYTE, vol. 6, No. 9, Sept. 1981, pp. 238-281.

[ERIC84] W. K. Erickson, W. C. Likens, An Application of Expert Systems Technology to Remotely Sensed Image Analysis, Pecora Conference, 1984, pp. 258-276.

[FAUG86] W. S. Faught, Applications of AI in Engineering, IEEE Computer, July 1986, pp. 17-27.

[FREI85] M. Freiling, J. Alexander, S. Messick, S. Rehfuss, and S. Shulman, Starting a Knowledge Engineering Project: A Step-by-Step Approach, AI Magazine, Fall 1985, pp. 150-164.

[GEVA82] W. B. Gevarter, An Overview of Expert Systems, U.S. Department of Commerce, PB82-227547, National Bureau of Standards, May 1982.

[GEVA83] W. B. Gevarter, Expert Systems: Limited but Powerful, IEEE spectrum, August 1983, pp. 39-45.

[GEVA85] W. B. Gevarter, Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics, Prentice-Hall Inc, New Jersey, 1985, pp. 46-66.

[GIB86a] H. G. Gibson, D. M. Knudson, and L. D. Gaultney, Using Expert Systems in Forestry: Application in the Dominican Republic, In Proc. Intern. Union of Forestry Res. Organizations, World Congress, Spetember 1986, Ljubljana, Yugoslavia.

[GIB86b] H. G. Gibson, D. D. Jones, and C-H Shih, Timber Harvesting Equipment Selection: An Expert System, ASAE Paper no. 86-1604, December 1986.

[HAYE84] F. Hayes-Roth, Knowledge-Based Expert Systems, IEEE Computer, October, 1984, pp. 263-273.

[HAYE85] F. Hayes-Roth, Knowledge-Based Expert Systems, Communications of the ACM, September, 1985, pp. 921-932.

[HARM85] P. Harmon and D. King, The Engineers Behind Expert Systems, Expert Systems: Artificial Intelligence in Business, John Wiley & Sons, Inc. New York, 1985.

[HUGG86] L. F. Huggins, J. R. Barrett, and D. D. Jones, Expert Systems: Concepts and Opportunities, Agriculture Engineering, Jan./Feb. 1986, pp. 21-23.

[LAFF86] T. J. Laffey, Fault Diagnosis, IEEE Potentials, Oct. 1986, pp. 8-11.

[LEMM86] H. Lemmon, Comax: An Expert System for Cotton Crop Management, Science, vol. 233, July 1986, pp. 29-33.

[LENA86] D. Lenat, M. Prakash, and M. Shepherd, CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks, The AI Magazine, Winter 1986, pp. 65-85.

[MCCA85] J. C. McCallum, Maintenance of Expert Systems: Life-Cycle Validity, SIGART Newsletter, October 1985, pp. 26 and 27.

[MICH85] R. H. Michaelsen, D. Michie, and A. Boulanger, The Technology of Expert Systems, BYTE, April 1985, pp. 303-312.

[MILL86] A. Miller, Expert Systems, IEEE Potentials, Oct. 1986, pp. 12-15.

[MILS86] W. L. Mills, Jr. Expert Systems: Applications in the Forest Products Industry, To appear in Forestry Products Journal.

[MITT85] S. Mittal and C. L. Dym, Knowledge Acquisition from Multiple Experts, The AI Magazine, Summer, 1985, pp. 32-36.

[MITT86] S. Mittal, C. L. Dym, and M. Morjaria, PRIDE: An Expert System for the Design of Paper Handling Systems, IEEE Computer, July 1986, pp. 102-114.

[MOON83] D. W. Mooneyhan, The Potential of Expert Systems for Remote Sensing, Seventeenth International Symposium on Remote Sensing of Environment, May 9-13, 1983, pp. 51-64.

[MORI85] K. Morik and C. Rollinger, The Real Estate Agent - Modeling Users By Uncertain Reasoning, The AI Magazine, Summer, 1985, pp. 44-52.

[NAU83] D. S. Nau, Expert Computer Systems, IEEE Computer, Feb. 1983, pp. 63-85.

[NAZI84] A. M. Nazif and M. D. Levine, Low Level Image Segmentation: An Expert System, IEEE PAMI, Sept. 1984, pp. 555-575.

[PIPI86] F. Pipitone, The FIS Electronics Troubleshooting System, IEEE Computer, July 1986, pp. 68-76.

[PRER85] D. S. Prerau, Selection of an Appropriate Domain for an Expert System, The AI Magazine, Summer, 1985, pp. 26-30.

[RAUC86] W. Rauch-Hindin, AI Shapes Up for Mainstream Use, Mini-Micro Systems, August, 1986.

[RICH83] E. Rich, Artificial Intelligence, McGraw-Hill, New York, 1983, pp. 284-291.

[ROD86a] G. Rodriguez and P. Rivera, A Practical Approach to Expert Systems for Safety and Diagnostics, INTECH, July 1986, pp. 53-57.

[ROD86b] G. Rodriguez and P. Rivera, An Expert System Shell to Evaluate Geothermal Reservoirs for Power Generation, Technical Report, Instituto de Investigaciones Electricas, Cuernavaca, Mexico, 1986.

[RYCH84] M. D. Rychener et al., A Rule-based Blackboard Kernel System: Some Principles in Design, IEEE Workshop on Principles of Knowledge-based Systems, 1984.

[SCHU85] J. K. Schueller, R. M. Slusher, and S. M. Morgan, A Speech Synthesis Grain Combine Expert System, ASAE Paper no. 85-1580.

[SRID85] N. S. Sridharan, Evolving Systems of Knowledge, AI Magazine, Fall 1985, pp. 108-120.

[STAN86] S. A. Stansfield, ANGY: A Rule-Based Expert System for Automatic Segmentation of Coronary Vessels From Digital Subtracted Angiograms, IEEE PAMI, March 1986, pp 188-199.

[STEF82] M. Stefik, J. Alkins, R. Balzer, J. Benoit, L. Birnbaum, R. Hayes-Roth, and E. Sacerdoti, The organization of Expert Systems, A Tutorial, Artificial Intelligence, vol. 18, 1982, pp. 135-173.

[STEF86] M. Stefik, and D. G. Bobrow, Object-Oriented Programming: Themes and Variations, The AI Magazine, Winter 1986, pp. 40-62.

[STRO86] G. J. Stroebel, R. D. Baxter, and M. J. Denney, A Capacity Planning Expert System for IBM System/38, IEEE Computer, July 1986, pp. 42-50.

[SUBR86] P. A. Subrahmanyam, Synapse: An Expert System for VLSI Design, IEEE Computer, July 1986, pp. 78-89.

[TALU86] S. N. Talukdar, E. Cardozo, and L. V. Leao, Toast: The Power System Operator's Assistant, IEEE Computer, July 1986, pp. 53-60.

[TAYL85] E. C. Taylor, Developing A Knowledge Engineering Capability in the TRW Defense Systems Group, The AI Magazine, Summer, 1985, pp. 58-63.

[THIE86] R. H. Thieme, D. D. Jones and H. G. Gibson, A Knowledge-based Structure for Forest Road Layout, ASAE Paper no. 86-5042, July 1986.

[THOM85] B. A. Thompson and W. A. Thompson, Inside an Expert System, BYTE, April 1985, pp. 315-330.

[UNID79] UNIDO, Guidelines for Evaluation of Transfer of Technology Agreements, Vienna.

[WALK86] T. C. Walker and R. K. Miller, Expert Systems 1986 - An Assessment of Technology and Applications, SEAI Technical Publication, ISBN: 0-89671-074-2.

[WALL86] P. Wallich, Assoc. Editor, Software 'doctor' prescribes remedies, IEEE Spectrum, October, 1986, pp. 43-48.

[WHI86b] A. D. Whittaker, et al., An AI Structure for Soil Erosion Modeling, ASAE Paper no. 86-5040, July 1986.

[WHI86a] A. D. Whittaker, et al., Getting Started with Expert Systems, ASAE Paper no. 86-5037, July 1986.

[WINS84a] P. H. Winston, Artificial Intelligence, Addison-Wesley, Massachusetts, 1984.

[WINS84b] P. H. Winston, LISP, 2nd Edition, Addison-Wesley, Massachusetts, 1984.

[WOOD86] D. D. Woods, Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems, AI Magazine, Winter 1986, pp. 86-92.