LECTURE PLAN (EXPERT SYSTEMS COM 418 E)

Lecture No.	Contents
1.	Features of Expert System
2.	Representation & Organization of Knowledge
3.	Basic characteristics of Expert System
4.	Types of problems handled by Expert System
5.	Case study of Prospector
6.	—— do ——
7.	Techniques of Knowledge Representation – Rule based
8.	Semantic Nets, Frames
9.	Programming Languages, Knowledge Engineering
10.	System Building Aids
11.	Support Facilities – Debugging Aids, I/O Facilities
12.	Explanation Facilities, Knowledge base editors.
13.	Stages in development of Expert System – Rules, Semantic Nets, Frames, Procedure-oriented methods.
14.	Object-oriented, Logic-based, Access-oriented methods.
15.	Expert System development – Tasks in Building Expert system
16.	Stages of Expert System Development
17.	Selection of Tool for building expert

Lecture No.	Contents
18.	Acquiring Knowledge – Knowledge Acquisition Process
19.	Interviewing the Expert
20.	On-site observation, Problem discussion, Problem description, Problem analysis, System refinement, system examination, system validation.
21.	Expert System Building Process – Problem, Characterizing the Problem
22.	Concepts needed to produce a solution, Knowledge formal representation
23.	Implementing a Prototype System, Validating the rules.
24.	Difficulties in developing an Expert System – Lack of Resources, Inherent Limitations of Expert Systems
25.	Expert Systems take a long time to Build
26.	Pitfalls in Planning – Choosing an Appropriate Problem.
27.	Resources for building the system
28.	Choosing the expert system building tool
29.	Pitfalls in dealing with Domain Expert – Choosing the Domain Expert
30.	Interacting with the Expert
31.	Difficulties during Development – System Implementation
32.	System Testing and Evaluation

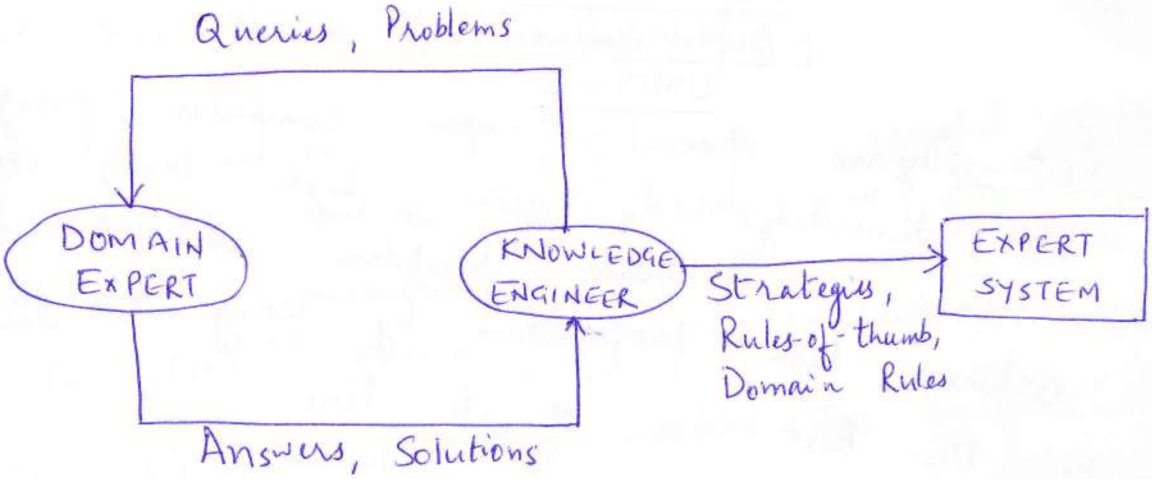
Expert Systems   CSE-446
(Dept. Elective - IV)
UNIT - I

(1) Expert Systems: They are special - purpose computer programs using expert Knowledge to attain high levels of performance in a narrow problem area. An expert system achieves high performance by using Knowledge to make the best use of its time.

(2) Knowledge Engineering: It is the process of building an expert system. It involves a special form of interaction between the expert - system builder known as Knowledge engineer and one or more human experts in some problem area.

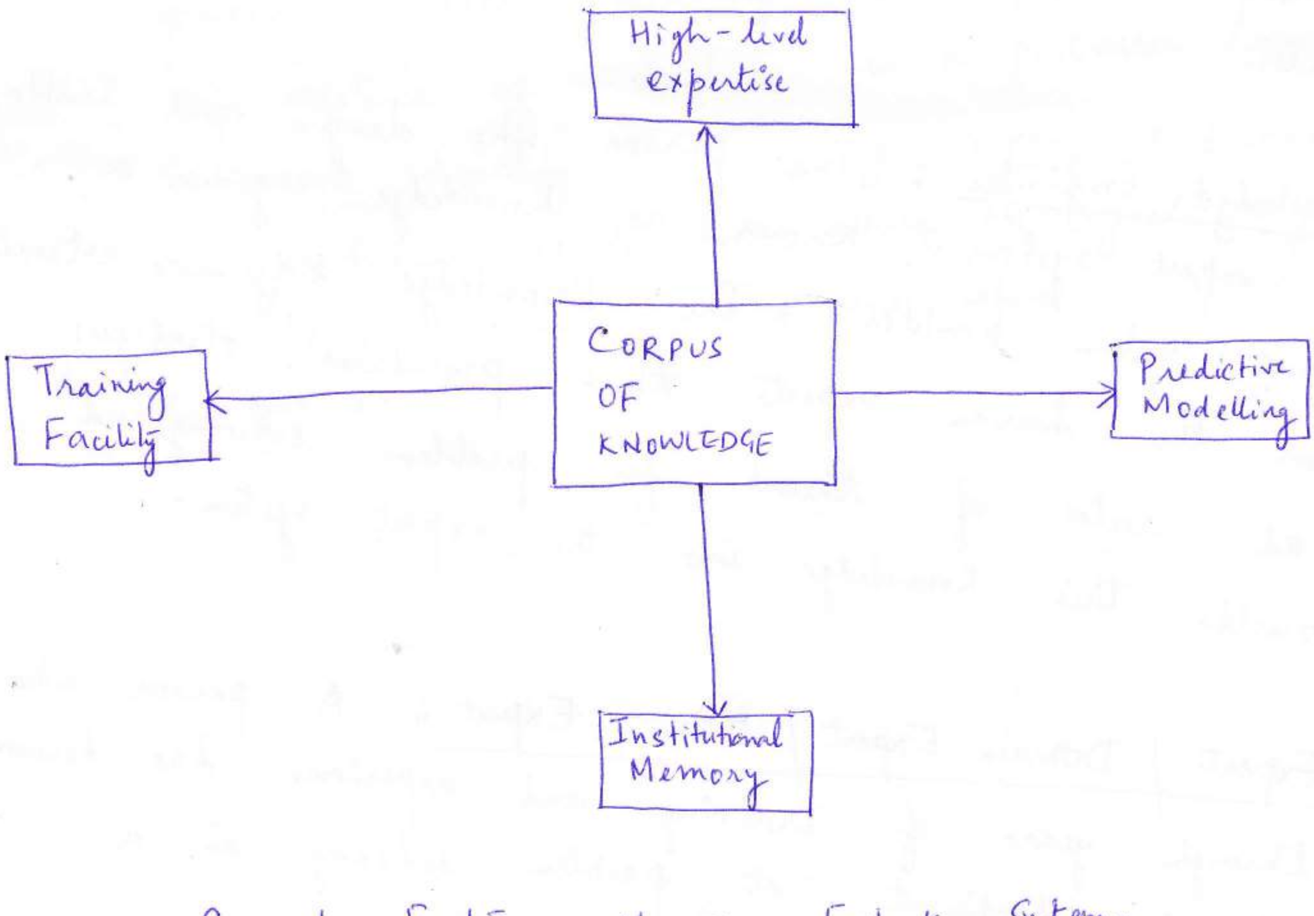
(3) Knowledge Engineer: The person who designs and builds the expert system is known as Knowledge engineer or Expert system builder. The Knowledge engineer extracts from the human experts their procedures, strategies and rules of thumb for problem solving and builds this Knowledge into the expert system.

(4) Expert / Domain Expert / Human Expert: A person who through years of training and experience has become extremely proficient at problem solving in a particular domain.



Knowledge Engg.: Transferring Knowledge from an expert to a computer program.

FEATURES OF EXPERT SYSTEM:



General Features of an Expert System.

The centre of an expert system is the corpus of Knowledge which accumulates during system building. The Knowledge is explicit and organized to simplify decision making. Moreover this Knowledge is accessible.

The features of an expert system are :

(1) <u>High-level expertise</u> :- Expert system provides high-level expertise to aid in problem solving. This expertise can represent the best thinking of the top experts in the field leading to problem solutions that are imaginative, accurate and efficient. Expert systems are cost-effective, able to earn its own way in the commercial marketplace and are flexible owing to the high-level expertise. The systems can grow incrementally as per the needs of business.

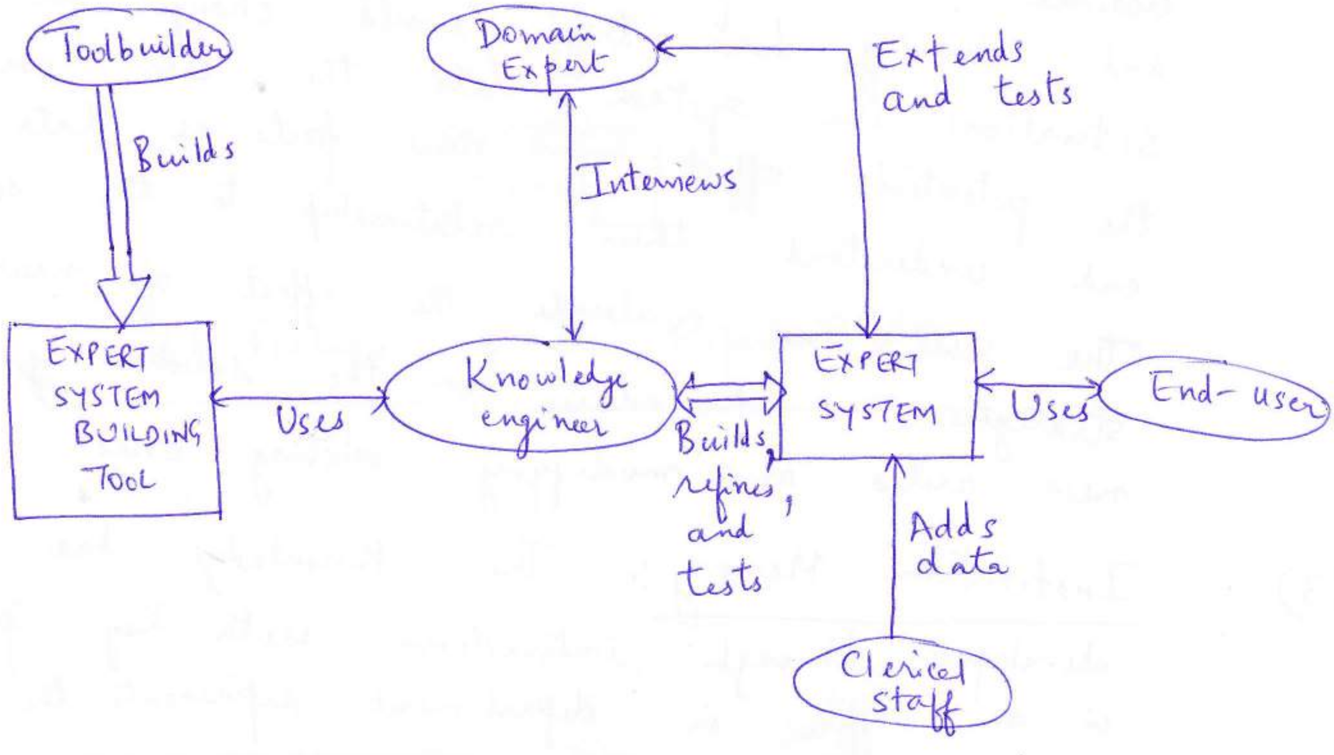
(2) <u>Predictive Modelling Power</u> :- Expert system can act as an information processing theory or model of problem solving in the given domain, providing the desired answers for a given problem situation and showing how they would change for new situations. The system lets the user evaluate the potential effect of new facts or data and understand their relationship to the solution. The user can evaluate the effect of new strategies or procedures on the solution by adding new rules or modifying existing ones.

(3) <u>Institutional Memory</u> :- The Knowledge base developed through interactions with Key personnel in an office or department represents the presen

policy of that group. This compilation of Knowledge becomes a permanent record of the best strategies and methods used by the staff. The expertise of the Key people is retained when they leave the organization.

(4) Training Facility:- Expert systems provide a training facility for Key personnel and important staff members since they contain the Knowledge and the ability to explain their reasoning processes. Software must be added to provide a smooth, friendly interface between the trainee and the expert system and Knowledge about teaching methods and user modelling must be included.

PLAYERS IN EXPERT SYSTEM BUILDING:



The main players in the expert system game are:

- Expert System
- Domain Expert
- Knowledge Engineer
- Expert - System Building Tool
- User

(1) Expert System: Expert System is a collection of programs or computer software that solves problems in the domain of interest. It contains both a problem-solving component and a support component. This support environment helps the user interact with the main program. e.g. Debugging aids which helps expert-system builder test and evaluate the programs' code, friendly editing facilities for helping experts to modify knowledge and data in the expert system.

(2) Domain Expert: Domain expert is an articulate, knowledgeable person proficient in providing good solutions to problems in a particular field. The expert uses tricks and shortcuts to make the search for a solution more efficient and the expert system models these problem solving strategies.

(3) Knowledge Engineer: Knowledge engineer is the person who builds and designs expert systems. He interviews the experts, organizes and represents the knowledge and help programmers in writing code.

(4) Expert - System Building Tool: It is the programming language used by the knowledge engineer or programmer to build the expert system.

(5) (a) End-User: The end-user is a person for whom the expert system was developed.

(b) User : The user is the person who uses the expert system once it is developed such as an end-user, a domain expert, a knowledge engineer or a clerical staff member.

ORGANIZATION OF KNOWLEDGE:

Knowledge : It is the information that a computer program must have so as to behave intelligently. This information can be in the form of facts or rules as shown :

FACTS : Tank # 23 contains sulphuric acid.
The plaintiff was injured by a portable power saw.

RULES : If the sulphate ion test is positive,
the spill material is sulphuric acid.
If the plaintiff was negligent in the use of product,
the theory of contributory negligence applies.

Facts and Rules in an expert system aren't always either true or false.

Facts: It refers to collection of data.

Rule : It is a formal way of specifying a recommendation directive or strategy, expressed as IF premise THEN conclusion or IF condition THEN action. Rules in expert systems can be either heuristics or algorithmic.

Rules

Heuristic   Algorithmic

(a) Heuristic : Heuristic is a rule of thumb or simplification that limits the search for solutions in domains that are difficult and poorly understood. e.g. Expert systems use heuristics for finding new mineral deposits. Heuristic method produces an acceptable solution most of

(b) <u>Algorithmic:</u> An algorithm is a formal procedure guaranteed to produce correct or optimal solutions. Algorithmic method is time-consuming and expensive.

Algorithmic vs Heuristic Methods

PREVENT SKYJACKING OF COMMERCIAL AIRLINERS

↓

PREVENT SKYJACKERS FROM BOARDING THE AIRLINERS

ALGORITHM vs HEURISTIC

ALGORITHM: Strip search every person who is given access to the airliner and search all luggage. This includes all passengers, flight crews and mechanics.

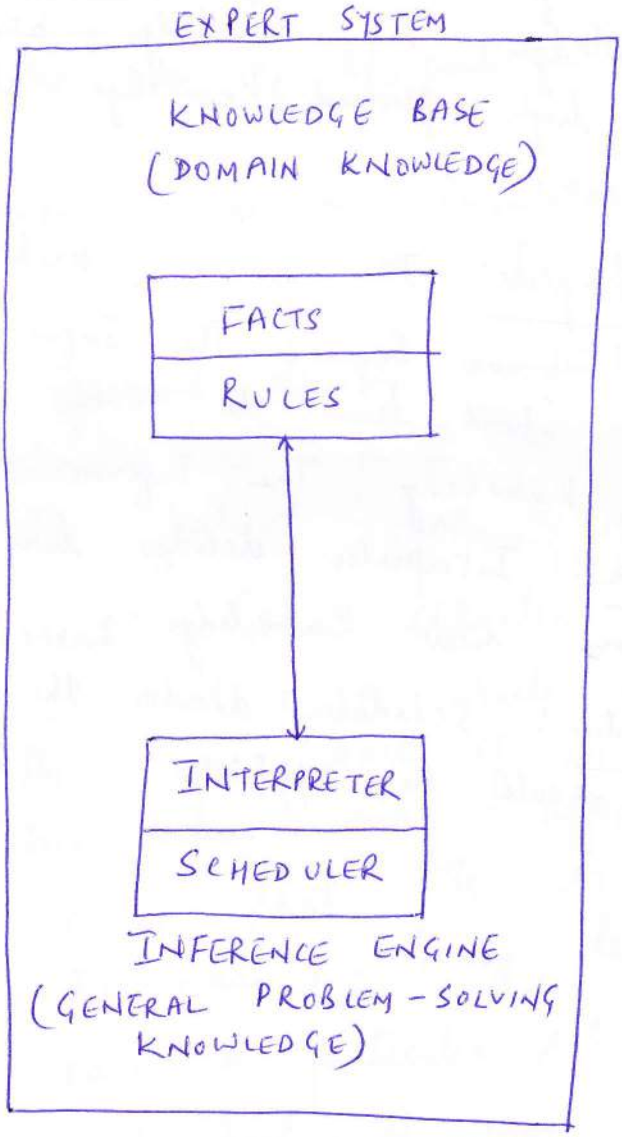
HEURISTIC: Putt all passengers and luggage thro' a metal detector. Search only those passengers that set off the detector or that match a predetermined skyjacker profile (based on certain characteristics such as age, dress, behavior etc.)

→ The algorithmic method would certainly stop skyjackings since it virtually guarantees that no one could board the plane with a weapon.

→ The heuristic method would also stop most skyjackings but could not guarantee that they wouldn't occur.

Expert Systems are Knowledge-based systems. A Knowledge-based system is a program in which the domain Knowledge is explicit and separate from the program's other Knowledge such as general Knowledge about how to solve problems or Knowledge about how to interact with the user. e.g. how to print characters at the user's terminal.

STRUCTURE / ARCHITECTURE OF EXPERT SYSTEM:



The Knowledge in an expert system is organized in a way that separates the Knowledge about the problem domain from the system's other knowledge.

Expert Systems basically consist of the following:

(1) Knowledge base: The collection of domain knowledge is called knowledge base. e.g. Knowledge about geology in an expert system for finding mineral deposits. The Knowledge-base in an expert system contains facts (data) and rules (or other representations) that use those facts as the basis for decision making. The separate domain knowledge makes it easier for the knowledge engineer to design procedures for manipulating this knowledge. The knowledge-base must contain lots of high-powered knowledge about the problem domain.

(2) Inference Engine: The general problem-solving knowledge is called Inference Engine. The Inference Engine contains knowledge about how to make effective use of the domain knowledge. The Inference Engine consists of

(a) Interpreter: Interpreter decides how to apply the rules to infere new knowledge.

(b) Scheduler: Scheduler decides the order in which the rules should be applied.

REPRESENTATION OF KNOWLEDGE :

Knowledge representation is the process of structuring knowledge about a problem in a way that makes the problem easier to solve. There are a standard set of knowledge representation techniques. Each technique provides the program with certain benefits such as making it more efficient, more easily understood or more easily modified. There are three techniques for representing knowledge :

(1) Rule - based Methods : Rule - based knowledge representation centers on the use of IF condition THEN action statement

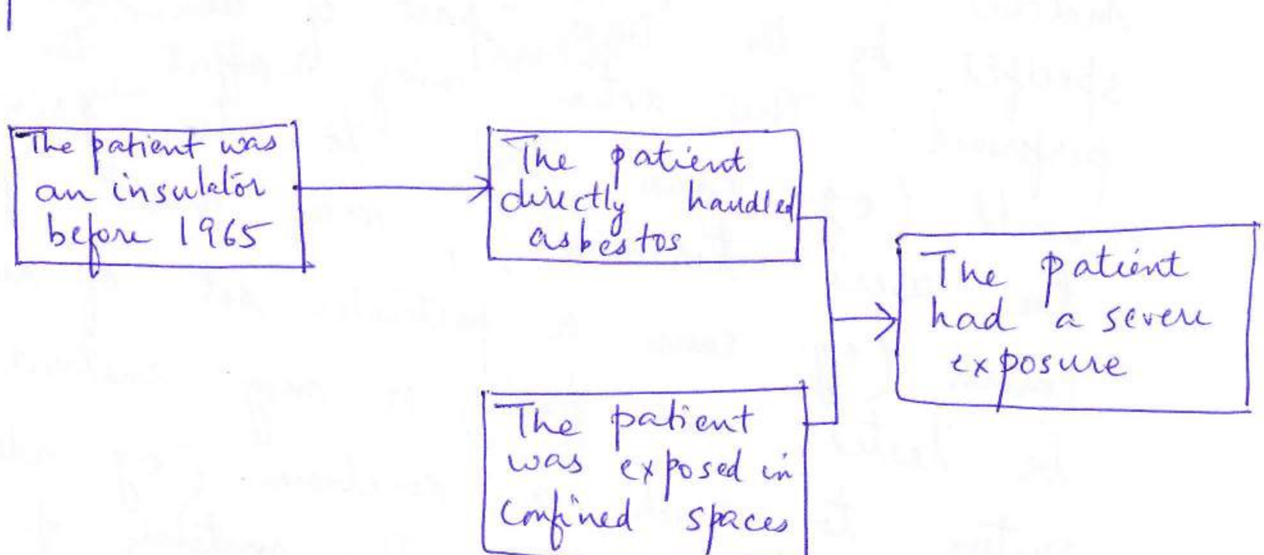
For e.g.

(a) If the patient was an insulator before 1965, then the patient directly handled asbestos.

(b) If the patient directly handled asbestos and the patient was exposed in confined spaces, then the patient had a severe exposure.

When the current problem situation satisfies or matches the IF part of a rule, the action specified by the THEN part of the rule is performed. This action may affect the outside world (e.g. cause text to be printed at the users terminal), may direct program control (e.g. cause a particular set of rules to be tested and fired) or may instruct the system to reach a conclusion (e.g. add a new fact to the database). This matching of rule IF portions to the facts produce inference chains. The inference chain formed from successive execution of rules 1 and 2 is as shown in figure below. This inference chain indicates how the system used the rules to infer the severity of the patients exposure to asbestos.

Rules provide a natural way of describing processes driven by a complex and rapidly changing environment. A set of rules can specify how the program should react to the changing data without requiring detailed advance knowledge about the flow of control. For programs where branching occurs frequently, rules provide opportunity to examine the state of the world at each step and react appropriately. The use of rules simplifies the job of explaining what the program did or how it reached a particular conclusion.

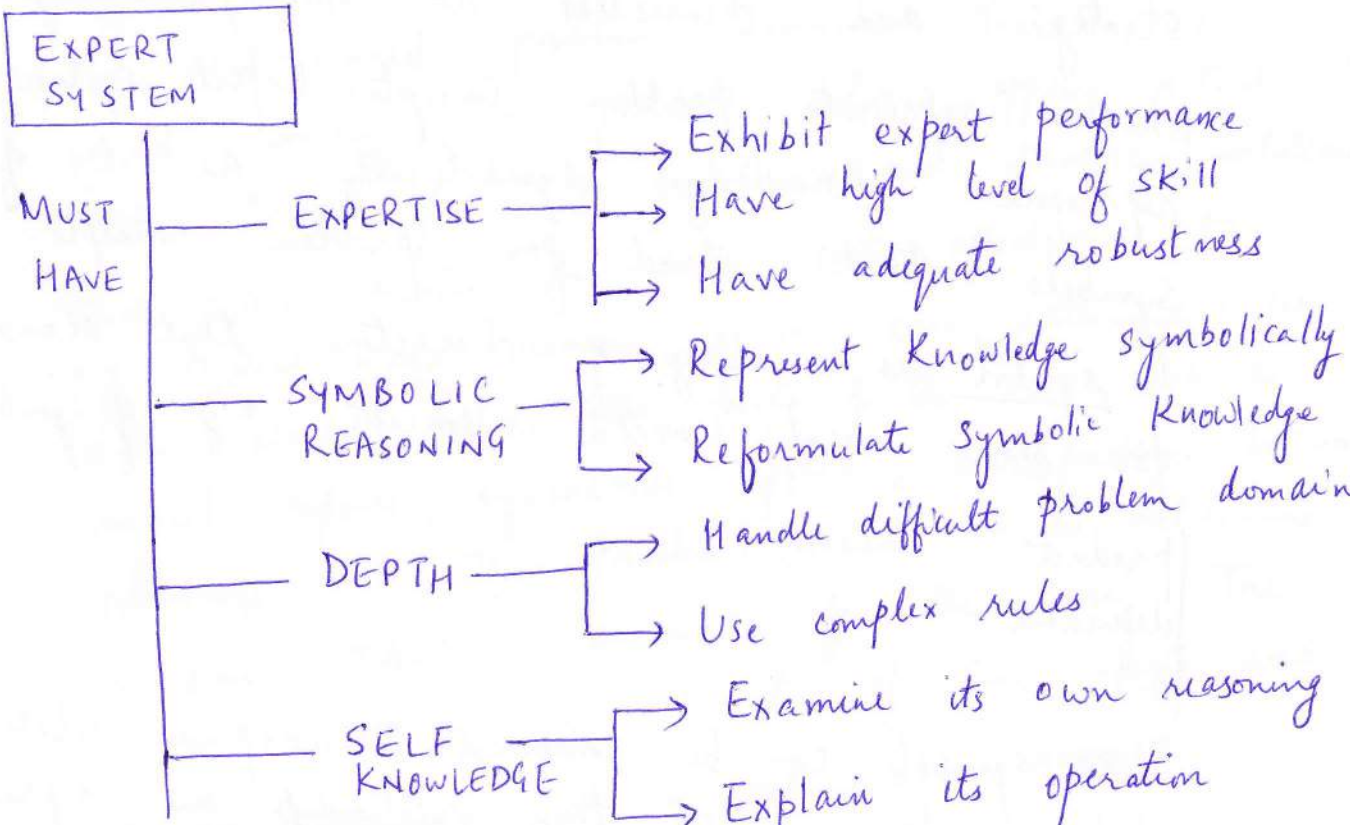


Inference chain for inferring the severity of asbestos exposure.

(2) Frames: Frame is a knowledge representation method that associates features with nodes representing concepts or objects. The features are described in terms of attributes (called slots) and their values.

(3) Semantic Nets: Semantic Net is a knowledge representation method consisting of a network of nodes, standing for concepts or objects, connected by arcs describing the relations between the nodes.

BASIC CHARACTERISTICS OF AN EXPERT SYSTEM :



The basic characteristics of an expert system are:

(1) Expertise: An expert system must perform well by achieving the same levels of performance in the domain of interest that human experts can achieve.

An expert system must be skillful. It should apply its knowledge to produce solutions both efficiently and effectively using shortcuts or tricks.

An expert system must have robustness. Robustness refers to that quality of a problem solver that permits a gradual degradation in performance when it is pushed to the limits.

(2) Symbolic Reasoning : Symbolic Reasoning refers to problem solving based on the application of strategies and heuristics to manipulate symbols which represents problem concepts. Expert systems represents Knowledge symbolically as set of symbols that stand for problem concepts.

A symbol is a string of characters that stands for some real-world concept. e.g of symbols are:

product
defendant
0.8

These symbols can be combined to express relationships between them. When these relationships are represented in an program, they are Known as Symbol structu

e.g. of Symbol structure are:

(DEFECTIVE product)
(LEASED - BY product defendant
(EQUAL (LIABILITY defendant) 0.8)

These symbol structures means "the product is defective", "the product is leased by the defendant" and "the liability of the defendant is 0.8".

An expert system performs Problem reformulation which means converting a problem stated in some arbitrary way to a form that lends itself to a fast, efficient solution.

(3) Depth: An expert system has depth i.e. it operates effectively in a narrow domain containing difficult, challenging problems. Expert systems work in real-world problem domains. In a real-world domain, the problem solver applies actual data to a practical problem and produces solutions that are useful in some cost-effective way. e.g. A real-world domain is a case settlement. The data would include the facts of the case such as medical reports, eyewitness reports, correspondence between attorneys etc. The problem would be to determine a fair settlement amount for the case. The solution might be an estimate of case worth and a recommendation of a settlement amount.

An expert system rules are sometimes necessarily complicated because of difficult challenging problems.

(4) Self-Knowledge: An expert system has knowledge that lets it reason about its own operation plus a structure that simplifies this reasoning process e.g. if an expert system is organized as sets of rules, then it can easily look at the inference chains it produces to reach a conclusion. If it's given special rules that tell it what to do with these inference chains, it can use them to check the accuracy, consistency and plausibility of its conclusions and can even devise arguments that justify or explain

its reasoning. This Knowledge the system has about how it reasons is called metaknowledge which means Knowledge about Knowledge.

An expert system have explanation facility. This is Knowledge for explaining how the system arrived at its answers. Most of these explanation involve displaying the inference chains and explaining the rationale behind each rule used in the chain.

Self-Knowledge is important in expert system because:

→ Users tend to have more faith in the results, more confidence in the system.

→ System development is faster since the system is easier to debug.

→ The assumptions underlying the system's operation are made explicit rather than being implicit.

→ Its' easier to predict and test the effect of a change on the system operation.

USES OF EXPERT SYSTEMS :

Uses

Basic Activities of Expert Systems

Types of Problems Solvable by Expert Systems

Basic Activities of Expert Systems :

Expert Systems solve different types of problems and the basic activities are as under :-

	Category	Problem Addressed
(1)	Interpretation	Inferring situation descriptions from sensor data
(2)	Prediction	Inferring likely consequences of given Situatio
(3)	Diagnosis	Inferring system malfunctions from observable behaviour such as situation descriptions, characteristics or knowledge about component design.
(4)	Design	Configuring objects under constraints.
(5)	Planning	Designing actions before acting.
(6)	Monitoring	Comparing observations or actual System behavi to expected behavior.
(7)	Debugging	Finding remedies for malfunctions.
(8)	Repair	Executing plans to administer prescribed remedies.
(9)	Instruction	Diagnosing, debugging and repairing student behaviour.
(10)	Control	Governing overall System behaviour

(1) Interpretation : Examples

→ Interpreting gauge readings in a chemical process plant to infer the status of the process.

→ Vision and speech understanding systems use natural input - visual images and audio signals to infer features and meaning.

→ Chemical interpretation systems use X-ray diffraction data or mass spectral and nuclear magnetic response data to infer the structure of compounds.

→ Medical interpretation systems use measurements from patient monitoring systems e.g. heart rate, blood pressure to diagnose and treat illnesses.

(2) Prediction : Examples

→ Predicting damage to crops from some type of insect.

→ Estimating global oil demand from current geo-political world situation.

→ Predicting next armed conflict occurrence based on intelligence reports.

(3) Diagnosis : Examples

→ Determining causes of diseases from symptoms observed in patients.

→ Locating faults in electrical circuits.

→ Finding defective components in the coolant systems of nuclear reactors.

(4) Design : Examples

→ Gene- cloning

→ Designing integrated circuit layouts

→ Creating complex organic molecules.

(5) Planning: Examples:

→ Planning for applying a series of chemical reactions to groups of atoms in order to synthesize a complex organic compound.

(6) Monitoring: Examples

→ Monitoring instrument readings in a nuclear reactor to detect accident conditions.

→ Assisting patients in an ICU by analyzing data from ICU monitoring equipment.

(7) Debugging: Examples

→ Suggest tuning a computer system to reduce a particular type of performance problem.

→ Selecting type of maintenance needed to correct faulty telephone cables

→ Choosing a repair procedure to fix a known malfunction in a locomotive.

(8) Repair: Examples:

→ Tuning a mass spectrometer i.e. setting the instruments operating controls to achieve optimum sensitivity consistent with correct peak ratios and shapes.

(9) Instruction: Examples

→ Teaching students to troubleshoot electrical circuits.

→ Instructing Navy personnel in the operation of a steam propulsion plant.

→ Educating medical students in the area of antimicrobial therapy selection.

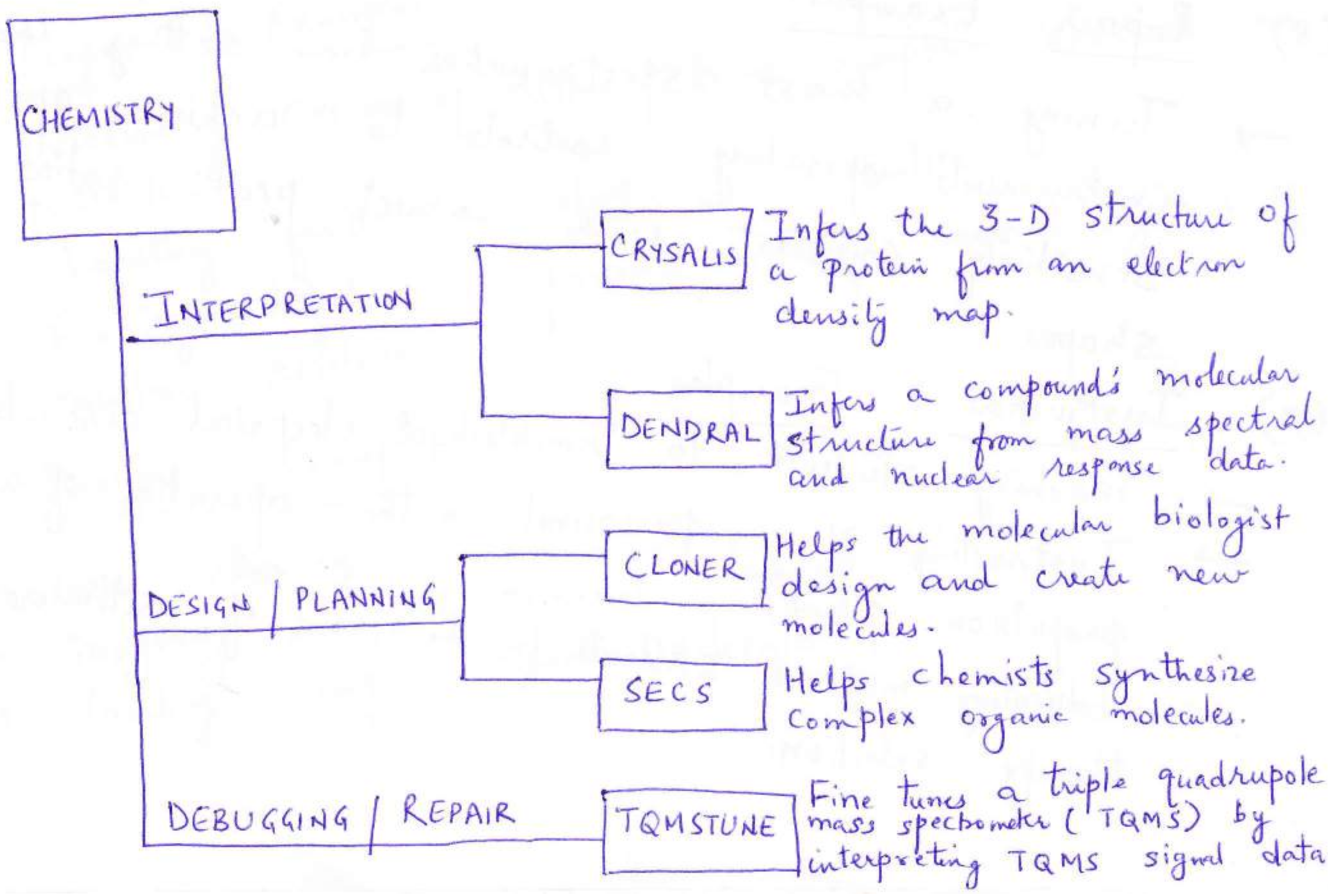
(10) Control : Examples :

→ Managing the manufacturing and distribution of computer system
→ Controlling the treatment of patients in an ICU.

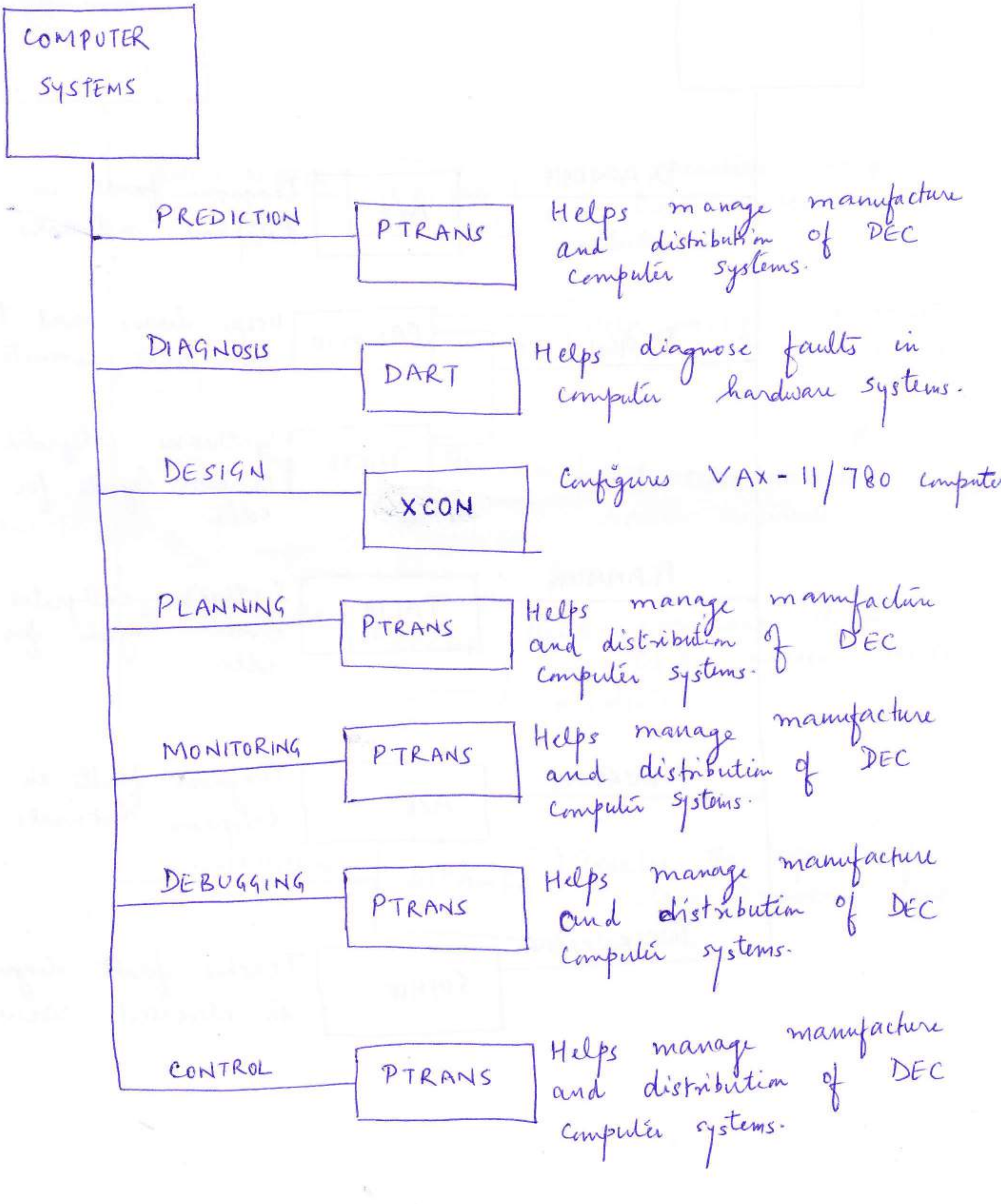
Types of Problems Solvable by Expert Systems:

Expert systems are used in the following areas:
Agriculture, Chemistry, Computer Systems, Electronics, Engineering, Geology, Information Management, Law, Manufacturing, Mathematics, Medicine, Meteorology, Military Science, Physics, Process Control, Space Technology.

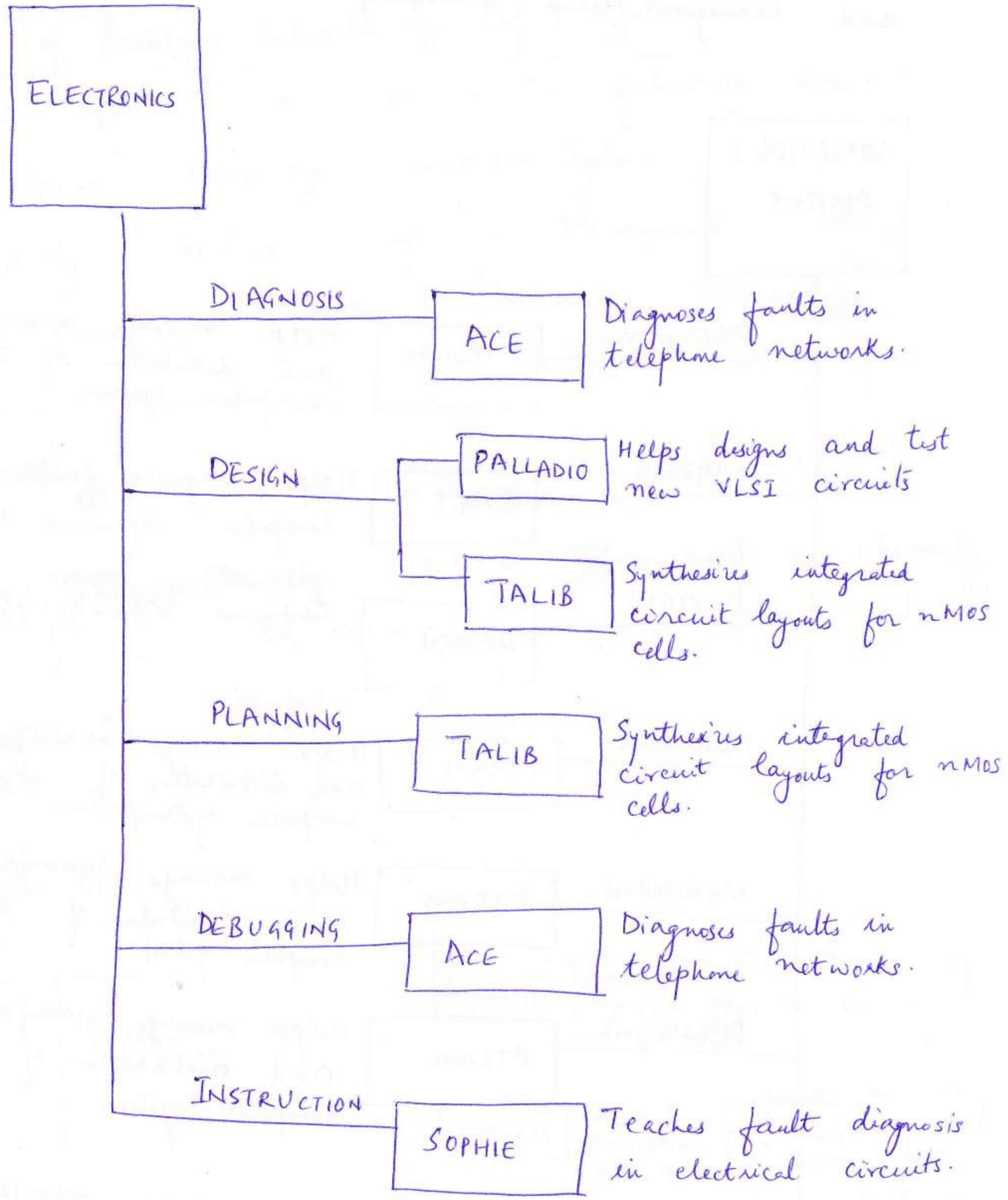
(1) Expert System in Chemistry : Expert system work in chemistry started with DENDRAL, a research project which began at Stanford University.



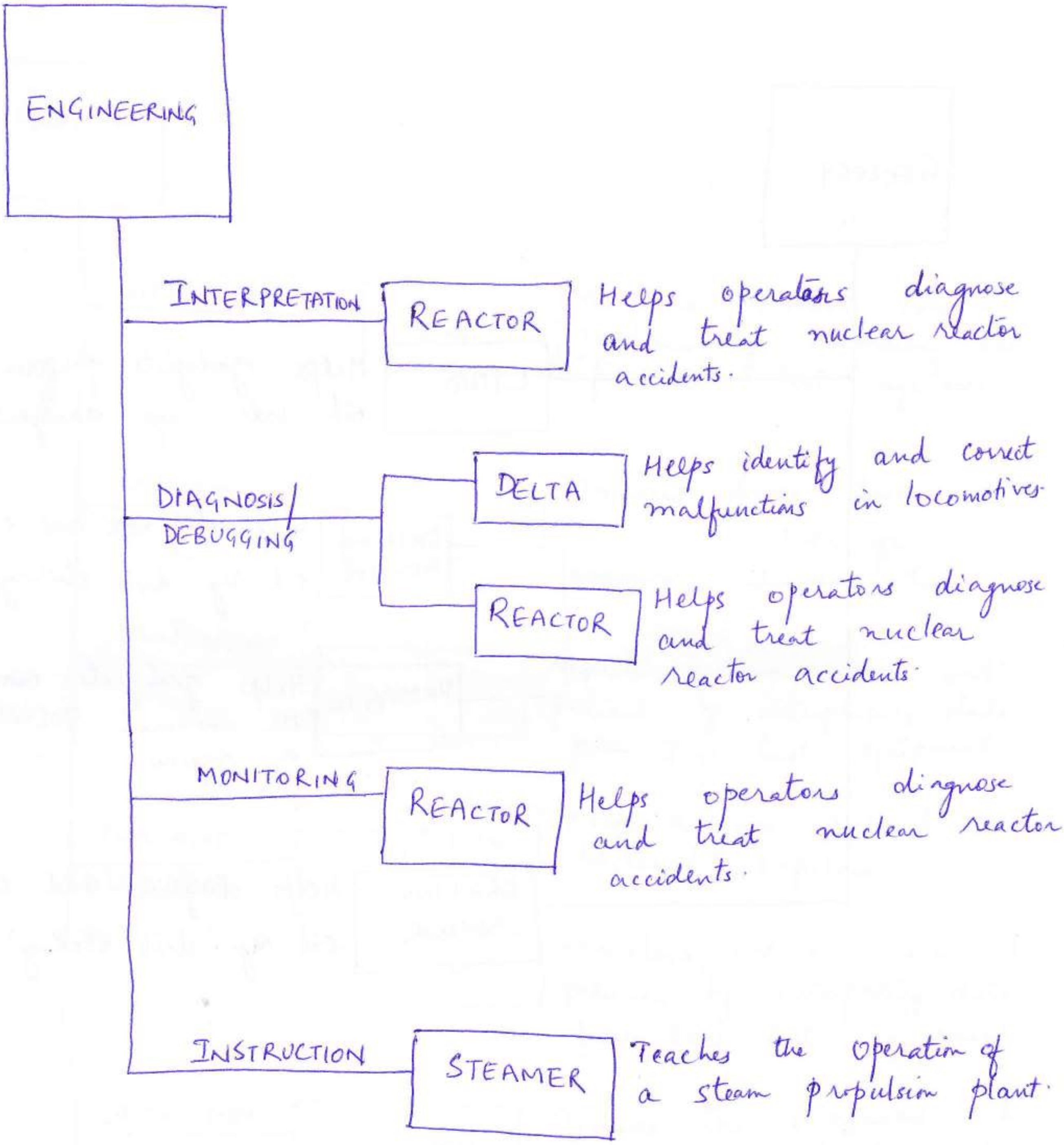
(2) Expert System in Computer Systems : Expert system work in computer systems started with XCON, a research project which began at Digital Equipment Corporation and Carnegie - Mellon University.



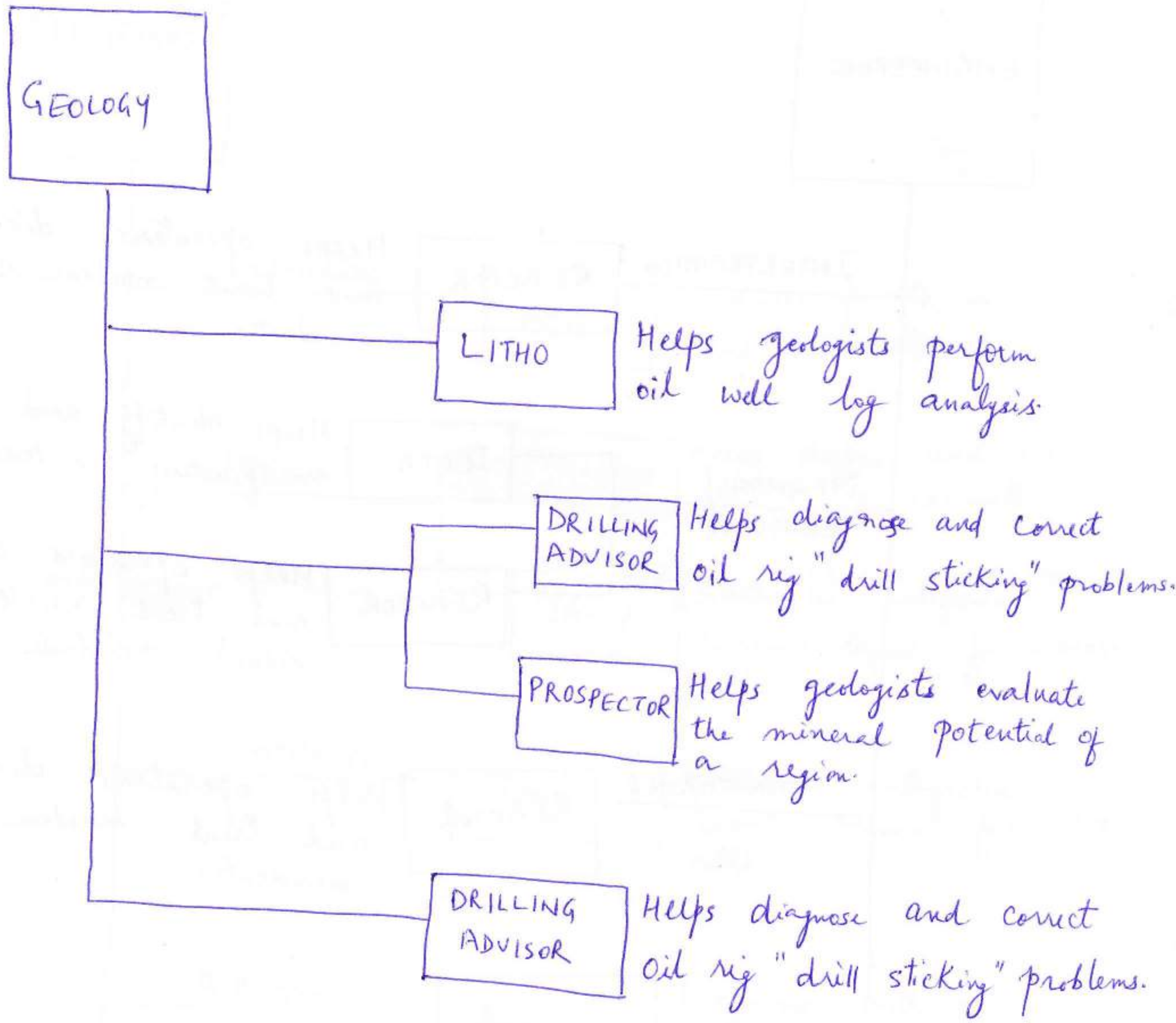
(3) Expert System in Electronics: Expert system in electronics began with ACE developed by Bell Laboratories.



4) Expert System in Engineering: Expert system in engineering started with DELTA developed by General Electric.



(5) Expert System in Geology : Expert system in geology began with PROSPECTOR developed by Stanford Research Institute.

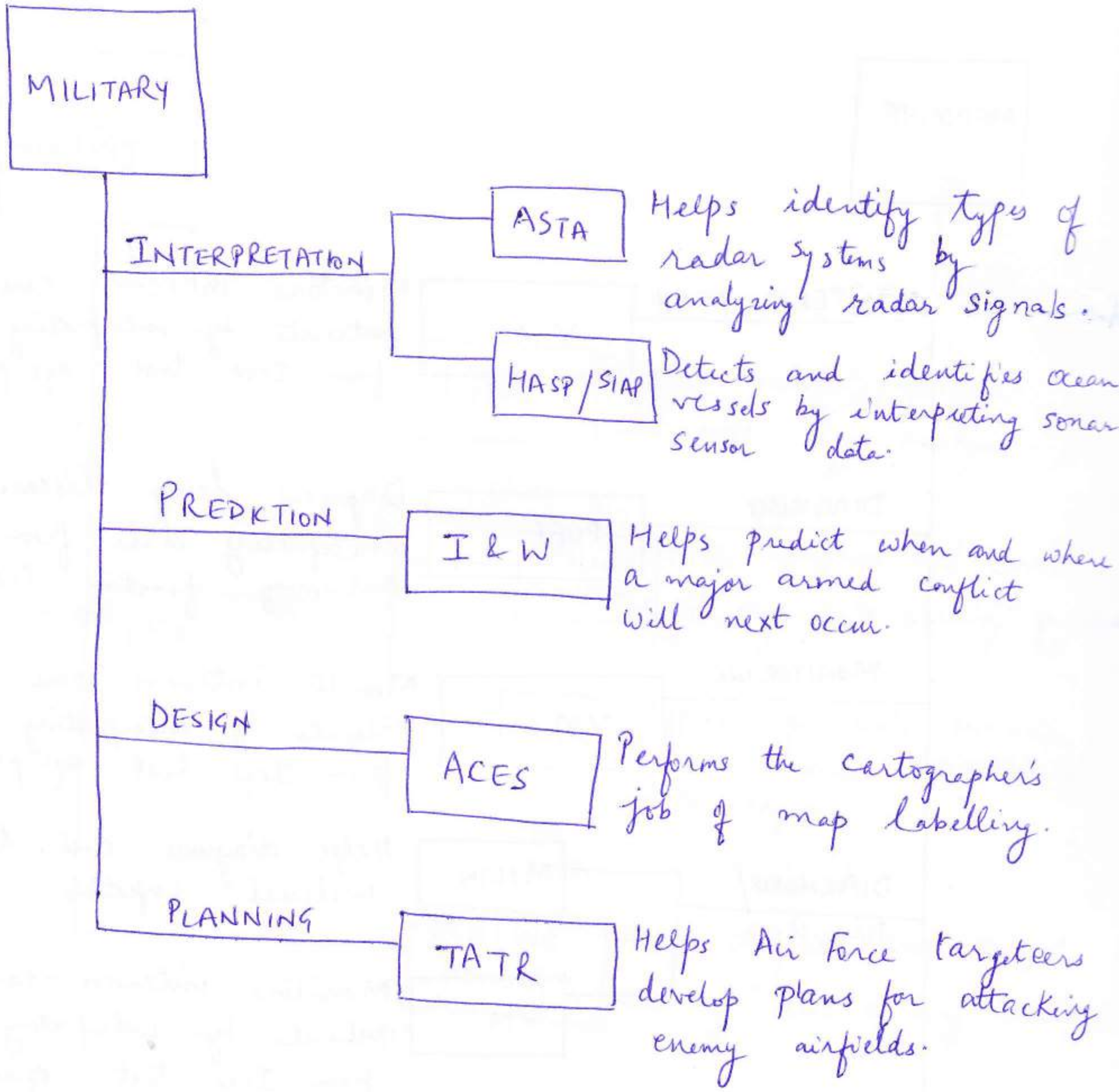


(6) Expert System in Medicine: Expert system in medicine began with MYCIN developed at Stanford University.

MEDICINE

Task	System	Description
INTERPRETATION	VM	Monitors intensive - care unit patients by interpreting data from ICU test equipment.
DIAGNOSIS	PUFF	Diagnoses lung disease by interpreting data from pulmonary function tests.
MONITORING	VM	Monitors intensive - care unit patients by interpreting data from ICU test equipment.
DIAGNOSIS/ DEBUGGING	MYCIN	Helps diagnose and treat bacterial infections.
	VM	Monitors intensive - care unit patients by interpreting data from ICU test equipment.
INSTRUCTION	GUIDON	Teaches the diagnoses and treatment of patients with bacterial infections.
CONTROL	VM	Monitors intensive - care unit patients by interpreting data from ICU test equipment.

(7) Expert System in Military Science: Expert System in military science started with HASP / SIAP developed by Stanford University and Systems Control Technology.



CASE STUDY OF PROSPECTOR :

What is PROSPECTOR :

PROSPECTOR has been the first Expert System built to assist geologists in mineral exploration. The work on PROSPECTOR started at Stanford Research University (SRI) in 1974 and continued until 1983.

Nine different mineral experts contributed their skills and expertise, working with several knowledge engineers and programmers. It took more than 30 person-years of effort to produce the current PROSPECTOR system including the field testing and evaluation. System development required extensive effort for several reasons:

- PROSPECTOR was directly implemented in INTERLISP, a powerful but relatively low-level language.
- A sophisticated support package was developed for PROSPECTOR that included both explanation and knowledge acquisition facilities.
- The system needed extensive domain knowledge. It contains 1,000 rules and uses a taxonomy of geological terms containing more than 1,000 entries.

One version of PROSPECTOR has Knowledge base with information about three different classes of ore deposits. The information is organized into three models of geological Knowledge :

- One for describing a type of sulfide deposit.
- One for a type of lead / Zinc deposit.
- One for a type of copper deposit.

Each model contains rules combined with semantic nets.

Operation of PROSPECTOR :

An exploration geologist starts by telling the program (expert system) the characteristics of a particular prospect of interest - the geologic setting, structural controls, and kinds of rocks, minerals and alteration products present or suspected. The program compares these observations with models of various Kinds of ore deposits, noting the similarities, differences and missing information. The program then engages the geologist in a dialog to obtain additional relevant information and uses that information to make an assessment of the mineral potential of the prospect. The goal here is to provide the geologist with a service comparable to giving him telephone access to authorities on many different Kinds of ore deposits.

PROSPECTOR at Work:

→ Features of the PROSPECTOR System are :-

- PROSPECTOR
  - Talks to the user (the geology expert) in terms with which the user is familiar.
  - Chooses a hypothesis about ore deposits in a region and then asks the user questions designed to either prove or disprove the hypothesis.
  - Allows the user to volunteer information about the region at any time during the consultation.
  - Rewords its questions when the user indicates they aren't understood.
  - Abandons its hypotheses and chooses new ones based on the answers given by the user to its questions.
  - Can tell the user why it needs the answer to a particular question.
  - Looks for inconsistencies in the user's answers, points them out, and allows the user to change both new and old answers to maintain consistency.
  - Presents its conclusions together with a summary of the reasons it has for reaching those conclusions.

→ A Partial taxonomy of PROSPECTOR terms is :
Forms, Ages, Materials, Rocks, Minerals etc.

How does PROSPECTOR do it:

The PROSPECTOR system contains rules linking observed evidence (E) of particular geological findings with hypotheses implied by the evidence. A rule has the form:

IF E THEN H (to degree) LS, LN.

This means that evidence 'E' suggests the hypothesis 'H' to a degree specified by the certainty factors 'LS' and 'LN'. 'LS' indicates how encouraging it is in the hypothesis to find the evidence present. 'LN' indicates how discouraging it is to find the evidence absent. Each piece of evidence and each hypothesis in the system has its own certainty factor 'P', standing for the probability that the evidence is present or the hypothesis is valid.

The evidence(E) in PROSPECTOR can be any logical combination of pieces of evidence, e.g.,

E1 and E2 and E3,

E1 or E2

E1 and (E2 or E3).

The hypothesis (H) is always a single concept, and can be used in the IF portion of a rule e.g., H2, and can be used to suggest or imply other hypotheses as shown:

$$H2 \longrightarrow H1\ (LS2, LN2)$$

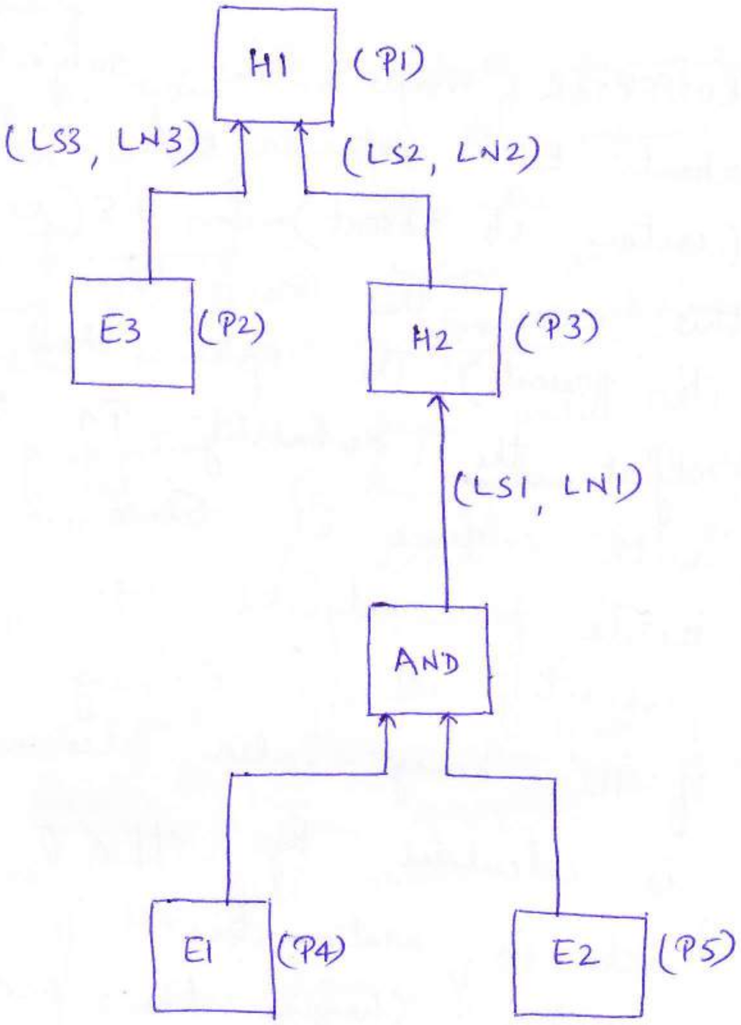
The rules in PROSPECTOR form a large inference net, which indicates all the connections between evidence and hypotheses and hence all the possible inference chains that could be generated from the rules. The inference net for three simple rules is as shown:

RULES : E1 and E2 $\longrightarrow$ H2 (LS1, LN1)

H2 $\longrightarrow$ H1 (LS2, LN2)

E3 $\longrightarrow$ H1 (LS3, LN3)

INFERENCE NET :



Inference net in PROSPECTOR.

Each of the three models in PROSPECTOR is a collection of hundreds of rules that forms an inference net. The values for the certainty factors LS and LN were defined when the model was built and remain unchanged during system operation. The values of 'P', the certainty factors for the evidence and hypotheses were also built into the model, but they change as new information is presented by the user. e.g.

Suppose the user presents the following information about evidence 'E1' in the figure shown previously on pag 33 :

E1 might be present in the region.

PROSPECTOR maps this subjective expression of certainty about 'E1' onto a scale that ranges from -5 (certain it's absent) to +5 (certain it's present). In this case, the no. chosen might be 2 (somewhat certain it's present). The system then uses the number 2 to adjust the probability P4 that was already associated with evidence 'E1'. Since '2' is greater than '0', P4 would be adjusted upward. As soon as P4, the probability of 'E1' occuring, changes, P3, the probability of 'H2' being valid also changes. The change in P3 is calculated by applying a generalization of Baye's rule in mathematics.

Changing the probability of 'E1' causes a change in the probability of 'H2' which in turn causes a change in the probability of 'H1'. This is Known as probability propagation. Probability propagation means the adjusting of probabilities at the nodes in an inference net to account for the effect of new information about the probability at a particular node. This probability propagation occurs automatically in PROSPECTOR whenever the user inputs new information. The propagation continues all the way to the top nodes, changing the probabilities of the goal hypotheses e.g., that the region contains a particular type of sulfide, lead / zinc or copper deposit. If the probability of the current active goal hypothesis falls below the probability of either of the other two goals as a result of probability propagation, PROSPECTOR switches goals, picking the new one with the highest probability.

The part of the system that actually propagates the probabilities forward (or upward) through the inference net is termed <u>PROSPECTOR's inference engine</u>. Since the rules are being processed by starting with the IF portions and proceeding to the THEN portions, this is known as <u>forward chaining</u>.

Probability propagation can't begin until the user supplies a new piece of information to the system. Therefore PROSPECTOR's inference engine must also decide what questions to ask the user. It does this by examining the rules that support the current goal hypothesis and asking the best question it can about the evidence in those rules. The best question is one that, when answered will most affect the probability of the goal node.

In the previous figure, if H1 were the current goal, the system would examine rules $E3 \rightarrow H1$ and $H2 \rightarrow H1$ to determine whether knowing E3 or H2 would most affect the probability of H1. If E3 would have more effect, the system would ask the user about E3. If H2 would have more effect, the system would use this same procedure to find the question that when answered, would most affect the probability of H2. This backward search through the rules continues until a question is finally chosen. Since the rules are being accessed from THEN to IF portions, this is a form of <u>backward chaining</u>.

Rules in PROSPECTOR tend to have a simple structure with only a few pieces of evidence in the IF portion of each rule.

e.g. It shows a set of <u>seven typical rules</u> from the PROSPECTOR system:

(1) IF : The igneous rocks in the region have a fine to medium grain size
THEN : They have a porphyritic texture (0, 5)

(2) IF : The igneous rocks in the region have a fine to medium grain size
THEN : They have a texture suggestive of a hypabyssal regional environment (2, 0.000001)

(3) IF : The igneous rocks in the region have a fine to medium grain size and they have a porphyritic texture
THEN : They have a texture suggestive of a hypabyssal regional environment (100, 0.000001)

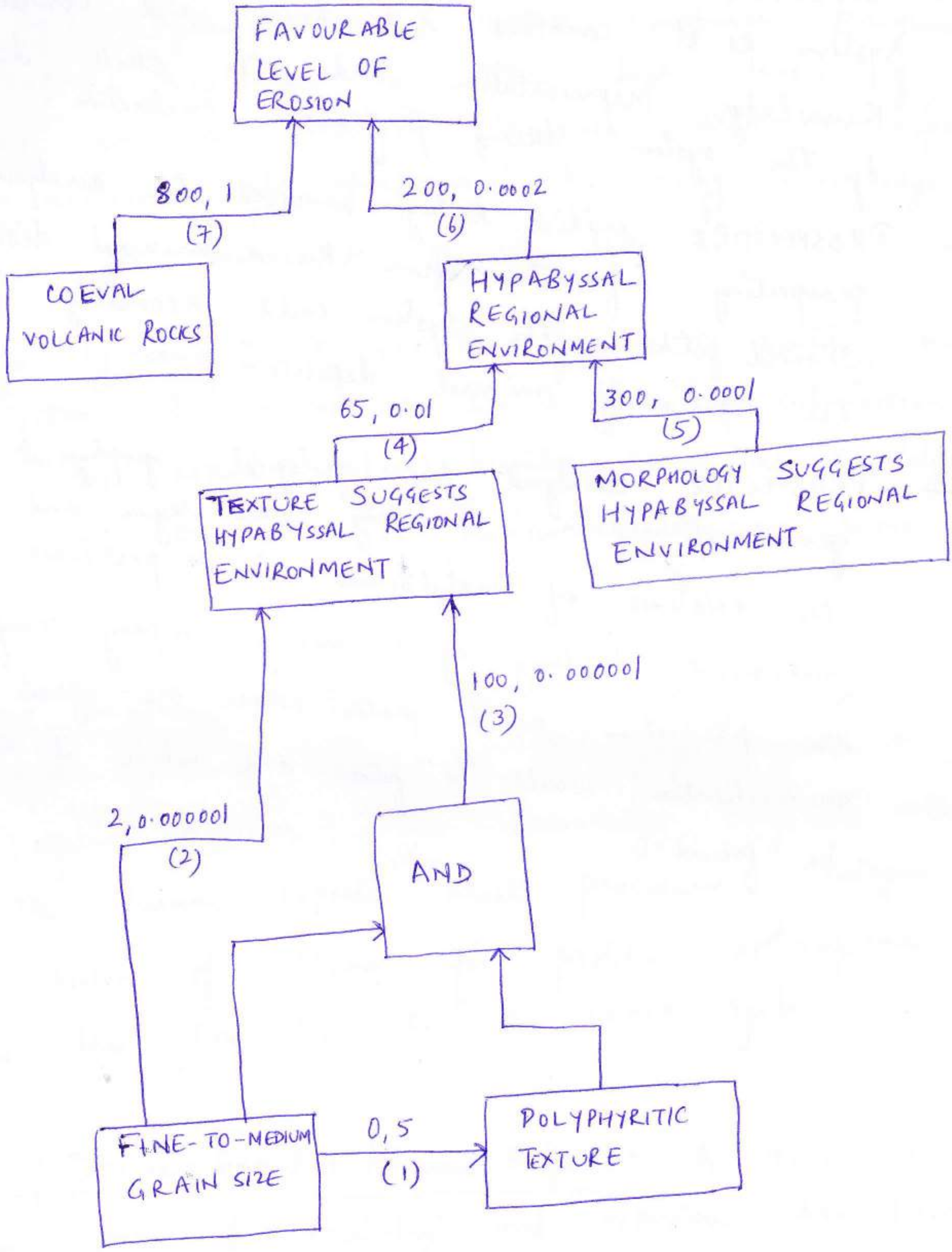
(4) IF : The igneous rocks in the region have a texture suggestive of a hypabyssal regional environment
THEN : The region is hypabyssal regional environment (65, 0.01)

(5) IF : The igneous rocks in the region have a morphology suggestive of a hypabyssal regional environment
THEN : The region is hypabyssal regional environment (300, 0.000

(6) IF : The region is a hypabyssal regional environment
THEN : The region has a favourable level of erosion (200, 0.00

(7) IF : Coeval volcanic rocks are present in the region
THEN : The region has a favourable level of erosion (800, 1

Inference net for previous rules:



What has PROSPECTOR accompolished :

- PROSPECTOR is an interesting and successful expert system as it combines rule-based and semantic net knowledge representation and the skill demonstrated by the system during performance evaluation.
- PROSPECTOR's initial testing involved the analysis of prospecting information from known mineral discoveries showed that the system could accurately predict the location of mineral deposits.
- PROSPECTOR analyzed the geological, geophysical and geochemical data describing the region and predicted the existence of molybdenum in a particular location. Subsequent drilling by a mining company confirmed the prediction as to both where ore-grade molybdenum mineralization would be found and where it would not be found.