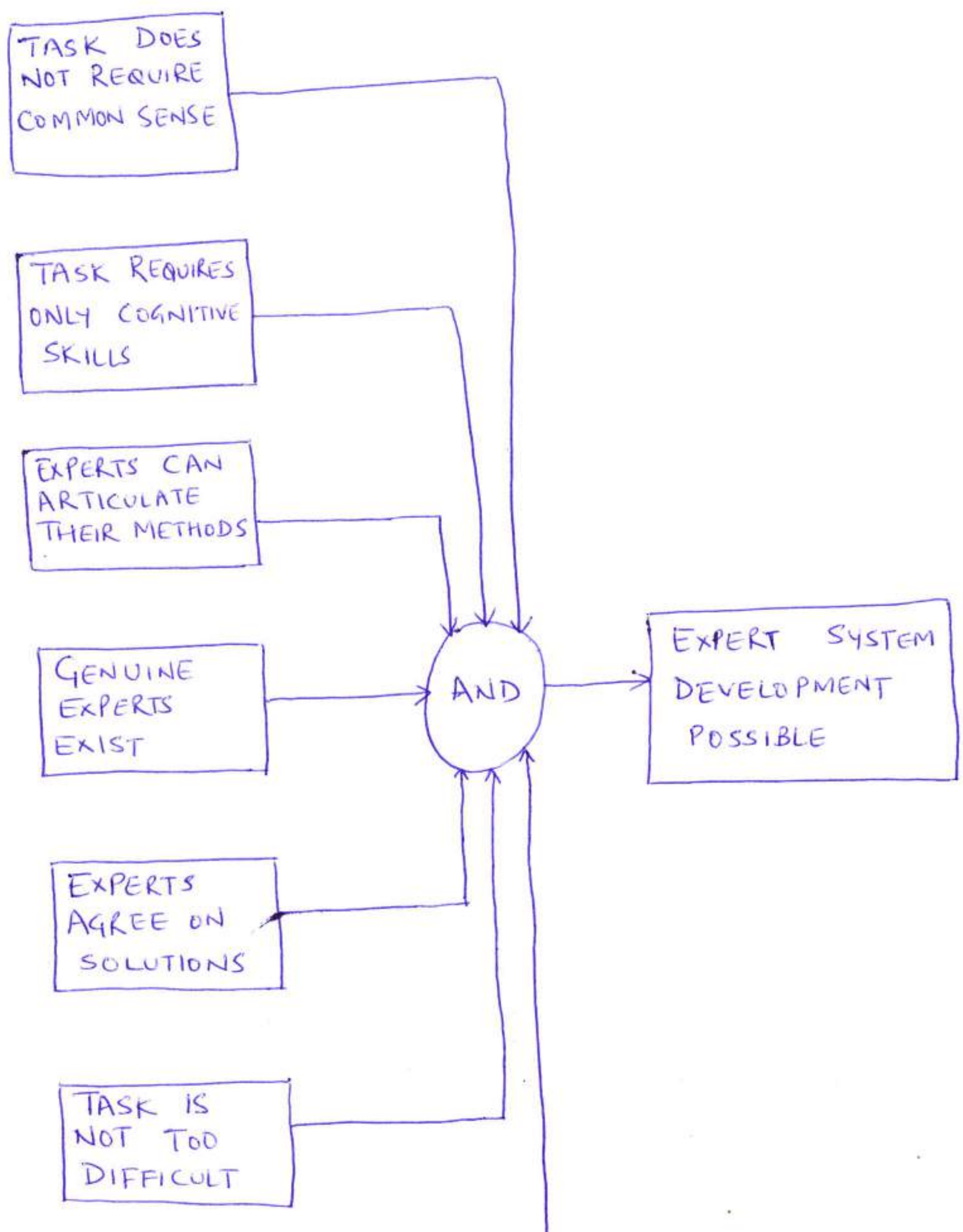
UNIT - III

EXPERT SYSTEM DEVELOPMENT :

- NECESSARY REQUIREMENTS FOR EXPERT SYSTEM DEVELOPMENT
- JUSTIFICATION FOR EXPERT SYSTEM DEVELOPMENT
- CHARACTERISTICS THAT MAKE THE USE OF EXPERT SYSTEMS APPROPRIATE

NECESSARY REQUIREMENTS FOR EXPERT SYSTEM DEVELOPMENT :



(1) Task does not require common sense : Knowledge engineering will not work if the task requires a significant amount of common sense.

(2) Task requires only cognitive skills : Task requiring cognitive skills can be handled by knowledge engineering techniques.

(3) Experts can articulate their methods : The experts must be able to articulate and explain the methods that they use to solve domain problems. If they can't do this, the knowledge engineers will have little success in extracting the knowledge from them and embedding it in a program.

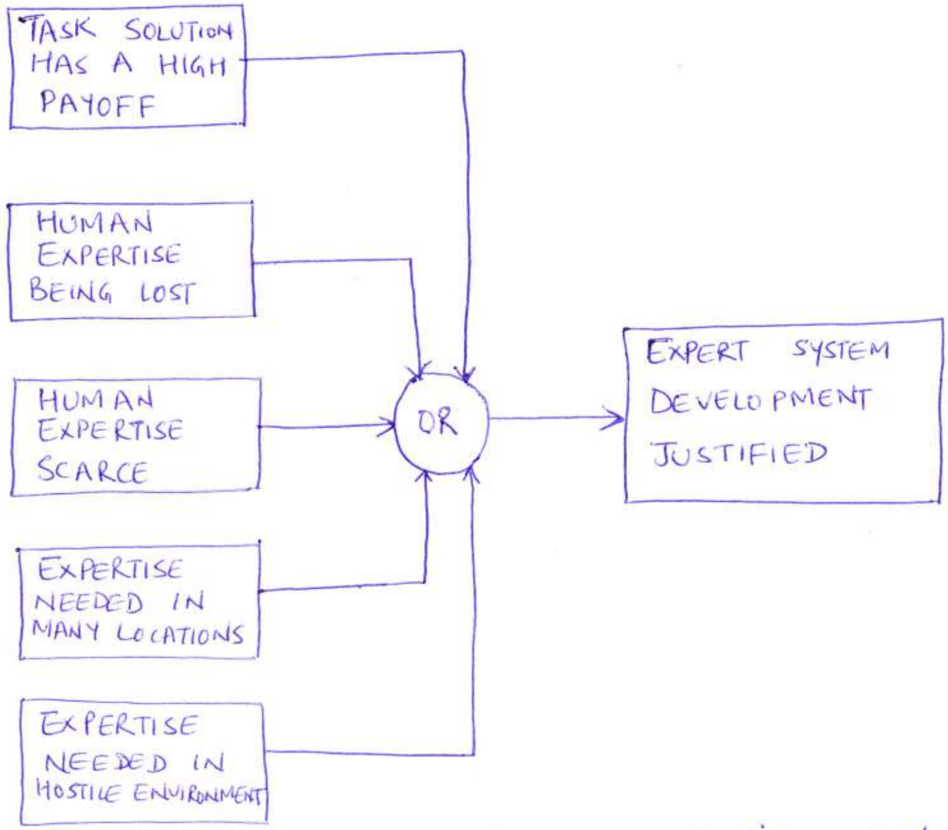
(4) Genuine experts exist : Genuine experts are people acknowledged to have an extremely high level of expertise in the problem area and are better at solving problems in a domain. The development effort will fail to produce a skillful program without a source of extensive and powerful knowledge.

(5) Experts agree on solutions : The experts must agree about the choice and accuracy of solutions in the problem domain.

(6) Task is not too difficult : The process may be too difficult to capture in an expert system as expertise can only be developed through on-the-job experience.

(7) Task is not properly understood : If the task is poorly understood that it requires basic research to find solutions, knowledge engineering will not work.

JUSTIFICATION FOR EXPERT SYSTEM DEVELOPMENT:



(1) Task solution has a high payoff: A company can justify expert system development when the task solution has a very high payoff. e.g. An expert system for mineral exploration could uncover a rich ore deposit worth millions of dollars. Development is a good idea if there is a reasonable possibility of a high payoff.

(2) Human expertise being lost: Expert systems are justified when significant expertise is being lost to an organization through personnel changes i.e. Retirement, job transfer. The institutional memory aspect of an expert system can minimize or eliminate this problem.

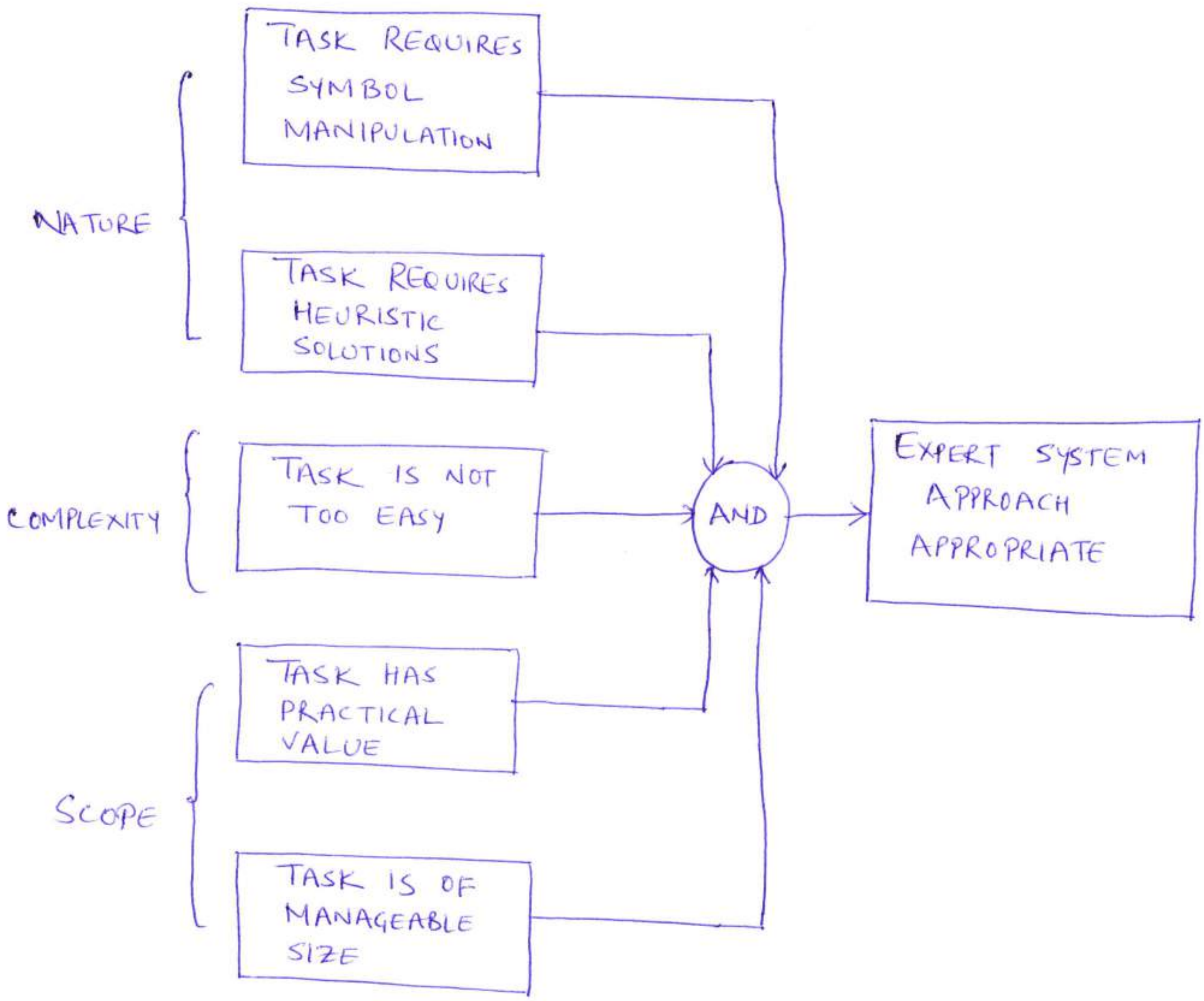
(3) Human expertise scarce: Human experts are scarce, very much in demand and expensive. An expert system is a cheap and effective way to handle this situation.

(4) Expertise needed in many locations: When a company requires similar expertise at many different locations, then an expert system is a cheap and effective way to handle such situation.

(5) Expertise needed in hostile environment: Expert system development is

t take place in an unfriendly or

CHARACTERISTICS THAT MAKE USE OF EXPERT SYSTEMS APPROPRIATE:



(1) Nature:

(a) Task requires symbol manipulation: A problem must be solvable by manipulating symbols and symbol structures. e.g. Mathematical Problems such as algebraic simplification are suitable for expert system development.

(b) Task requires heuristic solutions: Expert system development is suitable for problems that are heuristic in nature i.e. they require use of rules of thumb to achieve acceptable solutions.

(2) Complexity:

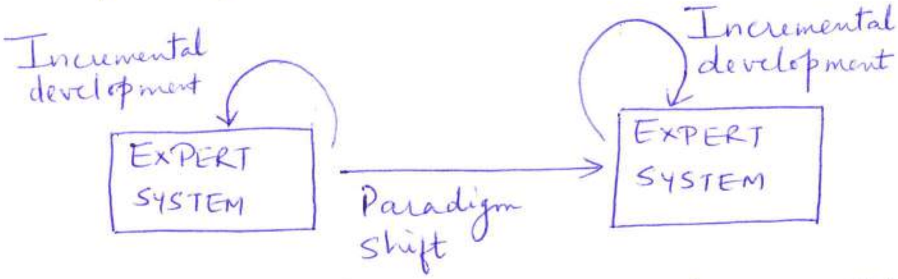
Expert System development is appropriate if the problem is not too easy. The problem is a serious problem, in a domain in which it takes a human years of study or pratice to achieve expert status.

(3) Scope:

(a) Task has practical value: The problem should be of practical

BUILDING AN EXPERT SYSTEM :

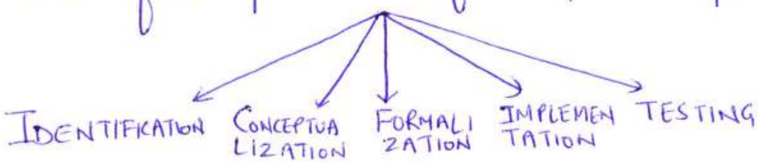
- The evolution of an expert system proceeds from simple to hard tasks by incrementally improving the organization and representation of the system's knowledge.
- The incremental approach to development means that system itself can assist in the development effort.
- After acquiring enough knowledge, the builders construct a very simple system and use feedback from the running model to direct and focus the effort.
- The incremental approach lets system builders earn profit.
- The result of incremental development is known as paradigm shift as shown :

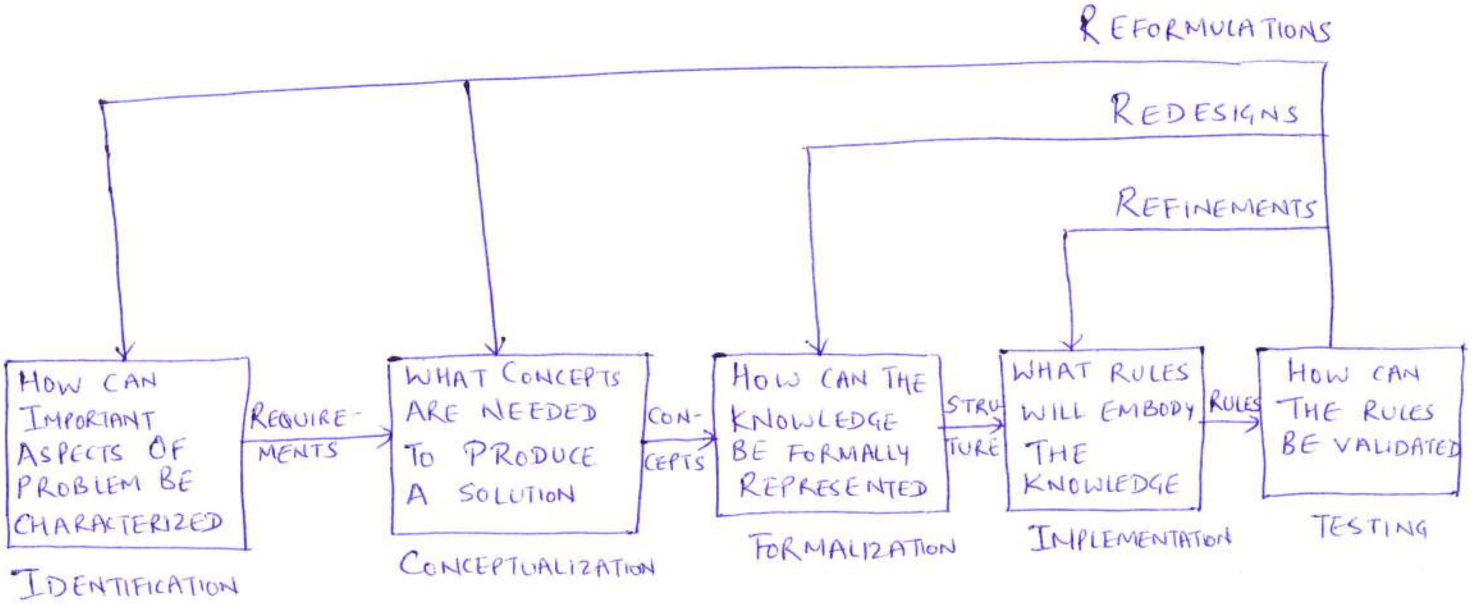


- The knowledge base may reach an unmanageable size at some point during development, the control becomes unwieldy and slow and the system may resemble a patchwork of routines and unintegrated constructs.
- At this time, redesigning and reimplementing the system may be considered.
- The knowledge engineer and domain expert should reexamine the problem and reassess their initial representation scheme.
- This leads to a more suitable architecture for the problem.

TASKS IN BUILDING EXPERT SYSTEMS:

There are five phases of expert system development:





During system development, the Knowledge engineer may engage in any of the five phases. Identification happens first and testing last.

(1) Identification: The Knowledge engineer and expert determine the important features of the problem which includes identifying the problem itself (e.g. type and scope), the participants in the development process (e.g. additional experts), the required resources (e.g. time and computing facilities) and the goals or objectives of building the expert system (e.g. improve performance or distribute scarce expertise).

(2) Conceptualization: The Knowledge engineer and expert decide what concepts, relations and control mechanisms are needed to describe problem solving in the domain. Subtasks, strategies and constraints related to the problem-solving ... also explored. The issue of granularity is

pick the abstract level of detail (coarsest grain) that provides adequate discrimination between Key concepts.

(3) Formalization: It involves expressing the Key concepts and relations within a framework suggested by an expert system building language. The Knowledge engineer should have ideas about appropriate tools for the problem. e.g. If the problem seems appropriate to a rule-based approach, the Knowledge engineer might select ROSIE as the system-building language and gather expertise in the form of IF-THEN rules.

(4) Implementation: In implementation phase, the Knowledge engineer turns the formalized Knowledge into a working computer program. Constructing a program requires content, form and integration. The content comes from the domain Knowledge made explicit during formalization i.e. the data structures, inference rules and control strategies necessary for problem solving. The form is specified by the language chosen for system development. Integration involves combining and reorganizing various pieces of Knowledge to eliminate global mismatches between data structures and rule or control specifications. Implementation of the initial prototype should proceed rapidly in order to check the effectiveness of the design decisions made during the earlier phases of development.

(5) Testing: It involves evaluating the performance and utility of the prototype program and revising it as necessary. The domain expert evaluates the prototype and helps the Knowledge engineer to revise it. As soon as the prototype runs on a few examples, it should be tested on many problems to evaluate its performance and utility. This evaluation may uncover problems with the representational scheme such as missing concepts and relations, Knowledge represented

The expert system must be refined and tested in a laboratory environment before it can be released for field testing. Users in the field expect the expert system to be of high quality, fast, reliable, easy to use, easy to understand. Thus the expert systems needs extensive field testing before it will be ready for commercial use.

STAGES OF EXPERT SYSTEM DEVELOPMENT:

There are five stages of expert system development:

- DEMONSTRATION PROTOTYPE
- RESEARCH PROTOTYPE
- FIELD PROTOTYPE
- PRODUCTION PROTOTYPE
- COMMERCIAL SYSTEM

(1) Demonstration Prototype: Most expert systems begin as a demonstration prototype. It is a small, demonstration program that handles a portion of the problem that will eventually be addressed. It is used in two ways: First, that the expert systems technology can effectively be applied to the problem in question and second, to test ideas about problem definition, scoping and representation for the domain. A rule-based demonstration prototype might contain 50 to 100 rules, perform adequately on one or two test cases and take one to three months to develop.

(2) Research Prototype: Most expert systems have evolved to the stage of research prototype, a medium-sized program capable of displaying credible performance on a number of test cases.

rule- based research prototype might contain 200 to 500 rules, perform well on a large number of test cases and take one to two years to develop.

(3) Field Prototype : Some expert systems have evolved past the research prototype to the stage of field prototype. These systems are medium - to large - sized programs that have been revised through testing on real problems in the user community. They are moderately reliable, contain smooth, friendly interfaces and address the needs of the end - user. A typical rule- based field prototype might contain 500 to 1000 rules, perform very well on many test cases and take two to three years to develop.

(4) Production Prototype : A few expert systems have reached the stage of production prototype. These systems are large programs that have been extensively field - tested and are likely to have been reimplemented in a more efficient language to increase speed and to reduce computer storage requirements. A typical rule- based production prototype might contain 500 to 1500 rules, provide accurate, fast and efficient decision making and take two to four years to develop.

(5) Commercial System : Only a very few expert systems have reached the stage of commercial system. These systems are production prototypes used on a regular commercial basis. e.g. XCON, a commercial system contains over 3000 rules, reaches correct conclusions 90 to 95 percent of the time and took six years to develop.

SELECTION OF EXPERT - SYSTEM - BUILDING TOOL :

The decision about what tool is appropriate for a specific problem task consists of three parts :

- SELECTING A TOOL
- EVALUATING A TOOL
- IS BEST DEVELOPMENT TOOL THE BEST PRODUCTION TOOL

ISSUES INVOLVED IN SELECTING A TOOL :

(1) Development Constraints : Expert system development requires time, money, personnel and hardware all of which influence the choice of a tool. These factors influence the decision about what type of tool to select i.e. a programming language (e.g. LISP) or a Knowledge engineering language (e.g. KEE).

- Programming languages offer more flexibility but they require the developer to design the Knowledge base and implement the inference engine that accesses the Knowledge. Development takes longer but the result fit the needs of the problem domain.
- Knowledge engineering languages offer less flexibility but more guidelines and mechanisms for how to represent and access the system's Knowledge. Development is easier, faster and cheaper.

The guideline is to pick a tool that complements the strength of the Knowledge engineering team.

(2) Support Facilities : Support facilities speed development and same time and money. The more extensive the support facilities, the more attractive the tool becomes. The Kinds of support facilities available in expert - system - building tools includes

(3) Reliability : An experimental or research tool can cause problems because of incomplete testing, obsolete documentation and fluctuating language specifications. The tool is reliable if it has a large user community and a reputation for being robust and well-debugged. The guideline is dont build an expert system with a tool still under development.

(4) Maintainability : The tool's developer should be available to maintain the tool's performance and clarify its use. A very old tool can be a problem since the developer may no longer have an interest in maintaining it or providing adequate documentation. The guideline is to pick a tool you will not have to maintain yourself during expert system development.

(5) Task Characteristics : The task characteristics influence the choice of an expert-system-building tool. Features of the problem will suggest types of solutions which in turn suggest particular tool features. Features of the application will suggest features needed by the expert system itself which will suggest particular tool features. The guideline is to pick a tool with features suggested by the problem and its application.

EVALUATING THE SYSTEM-BUILDING TOOL :

After selecting the tool, implement a prototype system with it in four to six weeks to test its effectiveness. This involves using the tool to solve a small, representative problem in the domain of interest. During prototype testing, particular attention is paid to the execution speed of the system. The degradation-of-performance (i.e. as the problem becomes harder, the performance of the tool degrades) criterion

IS BEST DEVELOPMENT TOOL THE BEST PRODUCTION TOOL :

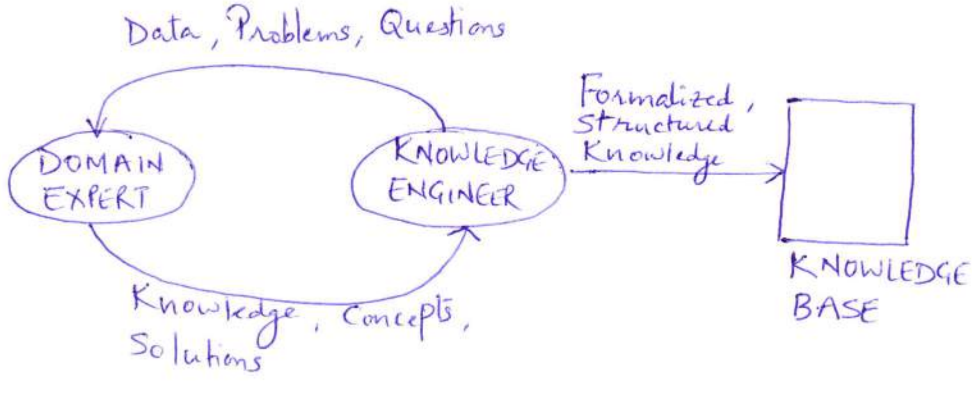
The tool best for developing the expert system may not be best for producing the system's final version. The development environment should have an abundance of support tools (e.g. debugging, editing) must provide the computing power to sustain a long-term development effort and must be flexible enough to allow experimentation with different types of representation and control techniques. The delivery environment must have good interfacing to the user (e.g. natural language or graphics) and must be both fast and efficient.

ACQUIRING KNOWLEDGE FROM THE EXPERTS:

Acquiring the Knowledge needed to power an expert system and structuring that Knowledge into a usable form is one of the primary bottlenecks in expert system development.

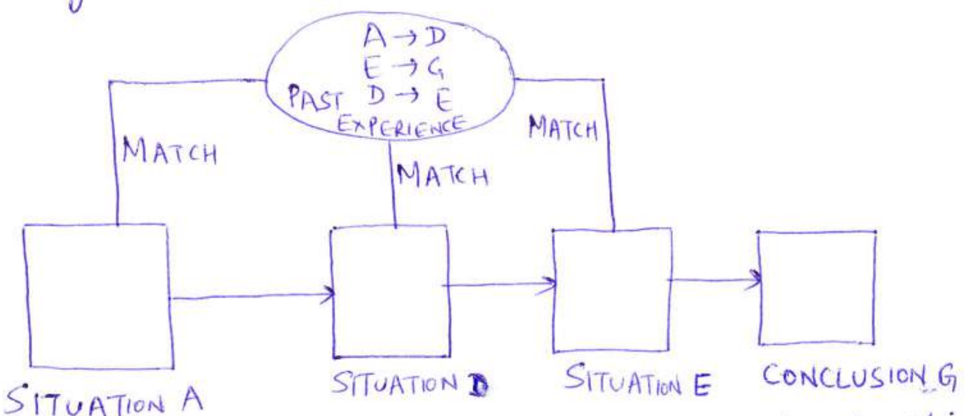
KNOWLEDGE ACQUISITION PROCESS:

A Knowledge engineer obtains the Knowledge through direct interaction with the expert as shown:



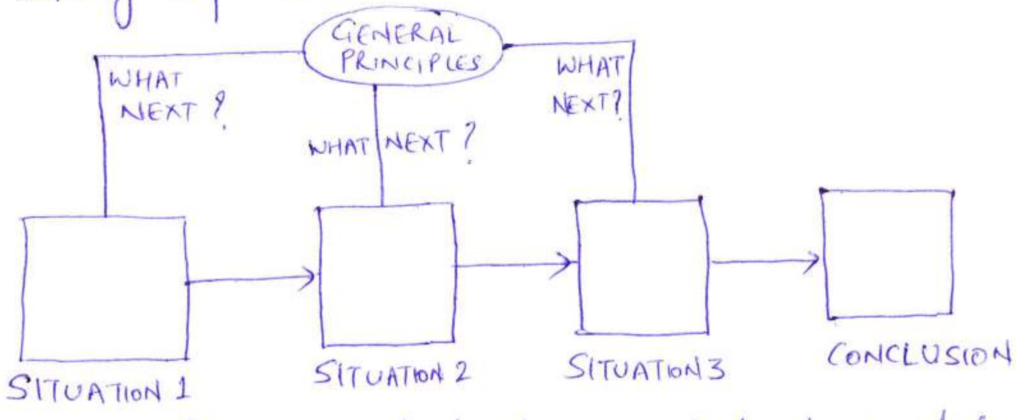
- The interaction consists of a prolonged series of interviews extending over a period of many months. The Knowledge engineer presents the expert with realistic problems to solve that are the type of problems the expert System is being designed to handle.
- The Knowledge engineer must work with the expert in the context of solving particular problems. It is effective to ask the expert directly about his or her rules or methods for solving a particular type of problem in the domain.
- The more competent domain experts become, the less able they are to describe the Knowledge they use to solve problems.
- If you are building an expert system and are a domain expert, having a Knowledge engineer help you understand and formalize your problem - solving methods. If you are a Knowledge engineer who has studied the domain extensively, still work with real expert. If you are a bonafide domain expert and an experienced Knowledge engineer, play the role of Knowledge engineer and find someone to act as domain expert. Good domain experts are plentiful than experienced

- Experienced Knowledge engineers will develop working hypotheses based on what the experts say and will test the hypotheses for validity and consistency by having the experts solve new problems requiring the hypothesized Knowledge.
- When experts solve problems in their area of expertise, they recognize new situations as instances of things with which they are already familiar as shown:



Problem-solving by an expert in familiar situation.

- When experts are faced with new or novel situations, they tend to apply general principles and deductive steps that provide casual links between various stages of a problem-solving sequence as shown:



Problem-solving by an expert in novel (new) situation.

INTERVIEWING THE EXPERT :

Psychologists and management scientists have studied experts and their problem solving techniques using observational and intuitive methods to measure performance and uncover expertise.

OBSERVATIONAL METHODS

INTUITIVE METHODS

(1) Observational Methods : This method relies on watching the expert solve the realistic problems in the domain being careful not to say or do anything that might influence the expert's problem-solving approach. Instead, investigators analyze a transcript of the session after the fact, possibly with the expert's help. This method is also known as protocol analysis where protocols provide information about the organization of an expert's Knowledge base, the actual Knowledge it contains and the control structures used to selectively apply that Knowledge. These methods are sometimes followed by a refinement phase in which experts comment on preliminary models developed to describe their behavior.

Difficulties :

The problem arises from huge gaps which often occur in the description of the process, when the expert accesses compiled Knowledge. The investigator has great difficulty filling in these gaps, even if he or she interviews the expert after the session has ended. If the expert is pushed to be more explicit, either during or after the problem-solving session, the expert may construct a line of plausible reasoning to explain his or her behavior. This line may or may not reflect the actual problem-solving techniques used.

(2) Intuitive Methods : This method relies on (a) introspection by the expert or (b) someone Knowledgeable about the subject area.
In (a), first method of assessment, an authentic or true expert acts as a builder of theories about his own behavior. The individual through introspection attempts to identify the basis for his or her own Knowledge and skill and then incorporates this directly into a computer system.
In (b), second method of assessment, an investigator of expertise studies and interacts with both experts and the literature of a field in order to become familiar with its major problem solving methods. The investigator develops a representation of expertise which is then checked against the opinion of other experts and eventually incorporated into a computer program.

Difficulties :

Since the expert uses introspection, the expert has trouble putting a finger on the actual techniques used to solve the problems. The Knowledge is well compiled through experience and overpractice that the expert accesses and manipulates it without thinking.

The Knowledge engineer relies on an interview technique that combines the observational and intuitive methods. The expert will introspect while solving a problem for the Knowledge engineer and talk about how he or she is solving it. The Knowledge engineer whenever appropriate asks relevant questions to stimulate and probe the expert. The Knowledge engineer asks questions, suggests possible rationales and hypothesizes concepts and rules.

Techniques for extracting Knowledge from domain expert :

(a) On-site observation : The Knowledge engineer observes the expert solving real problems on the job. The Knowledge engineer acts as a passive observer and it gives the Knowledge engineer insight into the complexity of the problem and the type of interface facility needed by the expert to use the finished system in the field. This technique will not be practical for some problem domains.

(b) Problem discussion : The Knowledge engineer picks a set of representative problems and discusses them with the expert. The Knowledge engineer determines how the expert organizes Knowledge about each problem, represents concepts and hypotheses and handles inconsistent, inaccurate or imprecise Knowledge and data relating to the problem.

(c) Problem description : The Knowledge engineer has the expert describe a typical problem for each main category of answer that could arise. This helps the Knowledge engineer define a prototypical problem for each category of answer - a construct that the expert system can use to help it select a strategy or basic approach to solving a given problem. This approach is used for diagnostic-type problems such as medical diagnosis.

(d) Problem analysis : The Knowledge engineer asks the expert to solve a series of problems. As the expert solves each problem, the Knowledge engineer provides any additional information or data requested by the expert. The expert must solve realistic problems, describing the solution process and giving as many [illegible] steps as possible. The Knowledge engineer questions hypotheses being entertained, strategies being used to generate hypotheses and goals being pursued which guide strategy selection.

(e) System refinement: The expert gives the Knowledge engineer problems to solve ranging from very easy to fairly difficult. Before the expert system is operational, the Knowledge engineer solves them on paper using the concepts, formalisms and rules acquired to date from the expert. As soon as the expert system can operate, the Knowledge engineer should use it to solve the problems supplied by the expert.

(f) System examination: The expert examines and critiques each rule in the prototype system and evaluates the control strategies used to select the rules. This includes verifying the accuracy of each rule and establishing a justification for each that the system can later use to explain its operation. The expert should compare the control strategies in the prototype with his or her method of handling problems in the domain.

(g) System validation: The Knowledge engineer presents the cases solved by the expert and the prototype system to other experts. This provides a way to compare strategies of different experts and find essential points of disagreement.

EXPERT - SYSTEM - BUILDING PROCESS :

In expert-system-building process, the Knowledge engineer follows a series of steps to develop an expert system.
The following steps are followed :

(1) Defining the Problem : The Knowledge engineer takes on the job of building an expert system in the problem domain.

(2) Characterizing the Problem : The Knowledge engineer begins the first phase of expert system building i.e. identification, where the task is to identify the participants, resources, goals and problem characteristics.

(3) Concepts needed to produce Solution : The Knowledge engineer enters the next phase of expert system building i.e. conceptualization, where he makes explicit the Key concepts and relations needed to solve problems in the domain.

(4) Formal representation of Knowledge : The Knowledge engineer enters the next phase of expert system building i.e. formalization, where the Key concepts, subproblems and control features are mapped into a more formal representation suggested by an expert-system-building tool.

(5) Implementing a Prototype system : The Knowledge engineer enters the implementation phase in which he turns the concepts and relations between them into a working computer program.

(6) Validating the Rules : The Knowledge engineer enters the testing phase, where he attempts to validate the rules that have been formulated.