EXPERT SYSTEM

Components of Expert System

Expert System:

An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field. It acquires relevant knowledge from its knowledge base, and interprets it as per the user's problem. The data in the knowledge base is essentially added by humans who are experts in a particular domain. However, the software is used by non-experts to gain information.

Expert systems in Artificial Intelligence are a prominent domain for research in AI. It was initially introduced by researchers at Stanford University and were developed to solve complex problems in a particular domain. The concept of expert systems was first developed in the 1970s by Edward Feigenbaum, professor and founder of the Knowledge Systems Laboratory at Stanford University. He explained that the world was moving from data processing to "knowledge processing," a transition which was being enabled by new processor technology and computer architectures.

Expert systems have played a large role in many industries including in financial services, telecommunications, healthcare, customer service, transportation, video games, manufacturing, aviation and written communication.

The Three C's of ES

The three C's of Expert System includes **Characteristics**, **Capabilities** and **Components** of an Expert System.

Characteristics/Features of Expert System:

An expert system operates as an interactive system that responds to questions, asks for clarification, makes recommendations and generally aids the decision making process. Expert system provides expert advice and guidance in a wide variety of activities from computer diagnosis to delicate medical surgery. An expert system is usually designed to have the following general characteristics.

1. **High level Performance**

The system must be capable of responding at a level of competency equal to or better than an expert system in the field. The quality of the advice given by the system should be in a high level integrity and for which the performance ratio should be also very high.

2. **Domain Specificity**

Expert systems are typically very domain specific. For ex., a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert would. The developer of such a system must limit his or her scope of the system to just what is needed to solve the target problem. Special tools or programming languages are often needed to accomplish the specific objectives of the system.

3. **Good Reliability**

The expert system must be as reliable as a human expert.

4. **Understandable**

The system should be understandable i.e. be able to explain the steps of reasoning while executing. The expert system should have an explanation capability similar to the reasoning ability of human experts.

5. **Adequate Response time**

The system should be designed in such a way that it is able to perform within a small amount of time, comparable to or better than the time taken by a human expert to reach at a decision point. An expert system that takes a year to reach a decision compared to a human expert's time of one hour would not be useful.

6. **Use symbolic representations**

Expert system use symbolic representations for knowledge (rules, networks or frames) and perform their inference through symbolic computations that closely resemble manipulations of natural language.

7. **Linked with Meta-knowledge**

Expert systems often reason with meta-knowledge i.e. they reason with knowledge about themselves and their own knowledge limits and capabilities. The use of meta-knowledge is quite interactive and simple for various data representations.

8. **Expertise knowledge**

Real experts not only produce good solutions but also find them quickly. So, an expert system must be skilful in applying its knowledge to produce solutions both efficiently and effectively by using the intelligence human experts.

9. **Justified Reasoning**

This allows the users to ask the expert system to justify the solution or advice provided by it. Normally, expert systems justify their answers or advice by explaining their reasoning. If a system is a rule based system, it provides to the user all the rules and facts it has used to achieve its answer.

10. **Special Programming Languages**

Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process. The major advantage of these languages, as compared to conventional programming languages is the simplicity of the addition, elimination or substitution of new rules and memory management capabilities. Some of the distinguishing characteristics of programming languages needed for expert system work are as follows:

- Efficient mix of integer and real variables.
- Good memory management procedures.
- Extensive data manipulation routines.
- Incremental compilation.
- Tagged memory architecture.
- Efficient search procedures.
- Optimization of the systems environment.

11. Explaining capability

Expert systems are capable of explaining how a particular conclusion was reached and why requested information is needed during a consultation. This is very important as it gives the user a chance to access and understand the system's reasoning ability, thereby improving the user's confidence in the system.

Capabilities of Expert System:

The expert systems are capable of a number of actions including:

- Advising
- Assistance in human decision making
- Demonstrations and instructions
- Deriving solutions
- Diagnosis
- Interpreting inputs and providing relevant outputs
- Predicting results
- Justification of conclusions
- Suggestions for alternative solutions to a problem

Components/ Architecture of Expert Systems

An expert system is typically composed of at least three primary components. These are the **knowledge base**, the **inference engine**, and the **User interface**. Other components may include **Knowledge Acquisition and learning module**, and **Explanation module**.

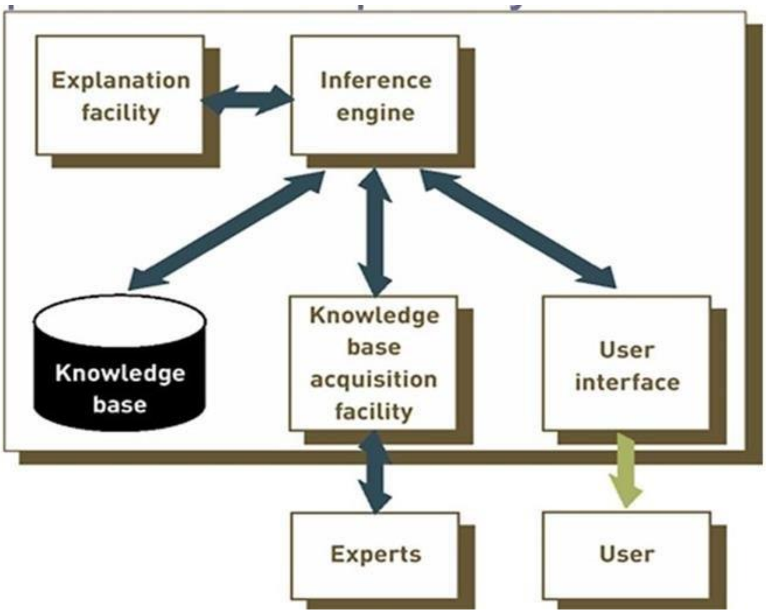


Fig: Components/ Architecture of Expert Systems

1. Knowledge Base

This is the most important element of an expert system since it holds the expert's problem solving knowledge. It is where the knowledge elicited from the expert is stored. It contains rules, facts and descriptions of objects etc.

With newer expert system products, the knowledge base is always stored in data. The information in knowledge base is everything that is necessary for understanding & formulating the problem & then solving it.

The key to knowledge base is how the knowledge is represented. The knowledge acquired from the expert has to be represented formally. Such knowledge representation deals with the structuring of the information, manipulation of information, and knowledge acquisition. The power of a system tends to be related from all sides of the knowledge in the knowledge base.

2. Inference Engine

It is that part of the program which regains & determines how to apply the knowledge in the knowledge base to the facts & premises presented at the user interface. It performs this task in order to deduce new facts which are subsequently used to draw further conclusions. The interference engine is the active component of an expert system. It is the Brain of the expert system.

It is also known as the control structure or the rule interpreter. This component is essentially a computer program that processes the knowledge base to achieve the goal stipulated by the user, who is communicating with the system via the user interface. It provides a methodology for reasoning about information in the knowledge base & for formulating conclusions.

Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct, flawless solution.

In case of **knowledge-based ES**, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

In case of **rule based ES**, it:

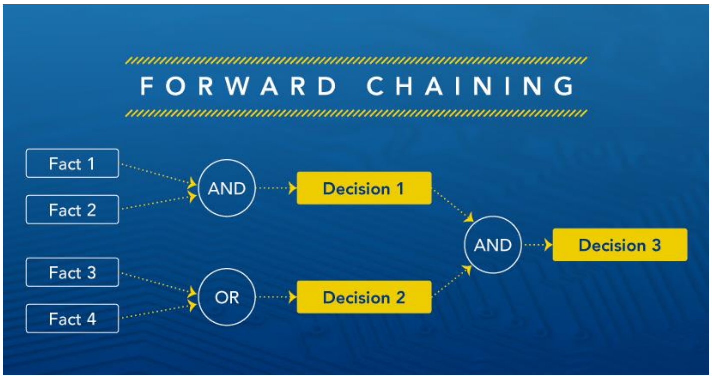
- Applies rules repeatedly to the facts, which are obtained from earlier rule application.
- Adds new knowledge into the knowledge base if required.
- Resolves rules conflict when multiple rules are applicable to a particular case.

To recommend a solution, the inference engine uses the following strategies:

- Forward Chaining
- Backward Chaining

Forward Chaining

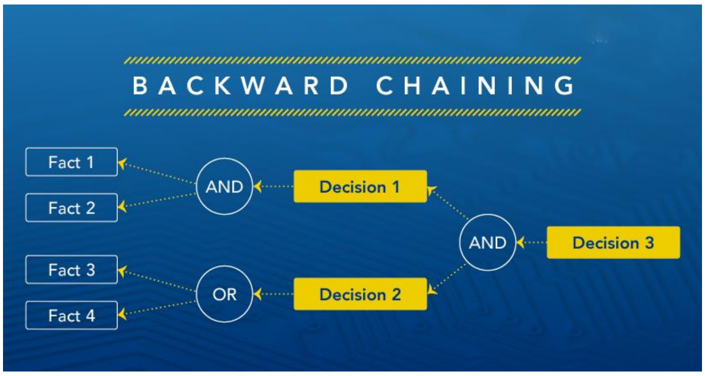
It is a strategy of an expert system to answer the question, **"What can happen next?"**



Here, the inference engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution. This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.

Backward Chaining

With this strategy, an expert system finds out the answer to the question, **"Why this happened?"**



On the basis of what has already happened, the inference engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.

3. User Interface

The human computer interface or user interface technology allows users to interact with the system. The user presents the problem and has the conclusions presented to him. A significant feature of some expert systems is that they can justify the conclusions reached as well as explain why certain options were used or discarded.

There are different ways in which the initiative can be shared between the system and the user. The most straight forward method is that in which the system determines the flow of the interactive session by prompting the user with questions and asking for data, to be inputted. In this instance, the user cannot volunteer information.

In systems, where the initiative is shared, the whole decision making process is shared between user and system. In a diagnostic system, the users can select a hypothesis and at each stage comment as to whether to continue along the same route or change it. So, this type of system is much more complex to design.

The ultimate interface would be one allowing the user to take all of the initiative; to be able to input any number of suggestions in a natural language form. This is highly complex and is under constant development.

4. Knowledge acquisition and learning module

This is the job of the knowledge engineer and this component functions to allow the expert systems to acquire more data from various sources and store it in the knowledge base. Experts make decisions based on qualitative & quantitative information. The system engineer has to translate the standard procedures into the form suitable for the expert system. As this is a skilled & time consuming operation, it is often this which limits the designing and functioning of expert system in a commercial environment.

Knowledge Acquisition program is used by an individual, who has expertise in the problem to, creates, add to or change the knowledge base. Potential sources of knowledge include human expert, research reports, textbooks, databases and the user’s own experience.

5. **Explanation module**

As the name suggests, this module helps in providing the user with an explanation of the achieved conclusion.

Advantages of Expert System

An ES is no substitute for a knowledge worker's overall performance of the problem-solving task. But these systems can dramatically reduce the amount of work the individual must do to solve a problem, and they do leave people with the creative and innovative aspects of problem solving.

Some of the possible organizational benefits of expert systems are:

1. They can complete its part of the tasks much faster than a human expert.
2. The error rate of successful systems is low, sometimes much lower than the human error rate for the same task.
3. They make consistent recommendations
4. These are a convenient vehicle for bringing to the point of application difficult-to-use sources of knowledge.
5. They can capture the scarce (rare) expertise of a uniquely qualified expert.
6. They can become a vehicle for building up organizational knowledge, as opposed to the knowledge of individuals in the organization.
7. When use as training vehicles, they results in a faster learning curve for novices (beginners).
8. The company can operate them in environments hazardous for humans.

Limitations of Expert System

No technology offers an easy and total solution. Large systems are costly and require significant development time and computer resources. Expert Systems also have their limitations which includes:

1. Limitations of the technology
2. Problems with knowledge acquisition
3. Operational domains as the principal area of ES application
4. Maintaining human expertise in organizations

Expert System Life Cycle

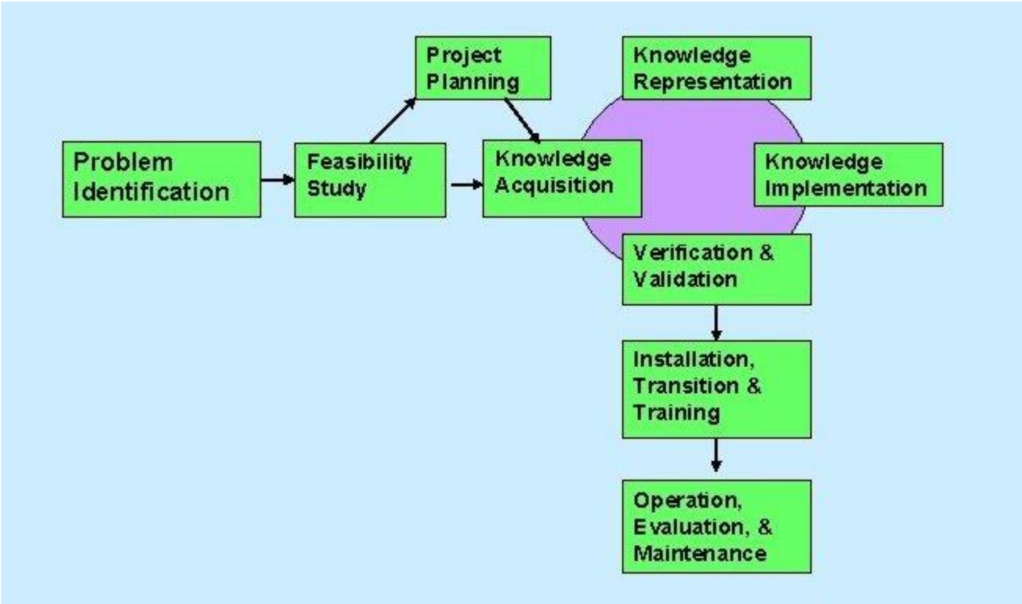


Fig: Expert System Development Life Cycle Phases (ESDLC)

1. **Problem Identification Phase**

   Identifying the problem and opportunity where the organization can obtain benefits from an ES, establishing the ES general goals.

2. **Feasibility Study Phase**

   Assessing the feasibility of the ES development in terms of its technical, economical, and operational feasibility.

3. **Project Planning Phase**

   Planning for the ES project, including development team workers, working environment, project schedule, and budget.

4. **Knowledge Acquisition Phase**

   Extracting domain knowledge from domain experts and determining the system's requirements.

5. **Knowledge Representation Phase**

   Representing key concepts from the domain, and inter relationships between these concepts, using formal representation methods.

6. **Knowledge Implementation Phase**

   Coding the formalized knowledge into a working prototype.

7. **Verification and Validation Phase**

   Verifying and validating a working prototype against the system's requirement, and revising it as necessary according to domain experts' feedback.

8. **Installation, Transition & Training Phase**

   Installing the final prototype in the operating environment, training the users, and developing documentation/user's manual.

9. **Operation, Evaluation & Maintenance**

   Running the system in an operating environment, evaluating its performance and benefits, and maintaining the system.

Comparison of Different Types of Expert System

Parameters	Rule based expert system	Fuzzy expert system	Frame based expert system	Neural expert system	Neuro-fuzzy expert system
Knowledge Representation	IF-THEN rules in the knowledge base	Based on degree of membership using fuzzy logic	Knowledge in frames using hierarchical structure	IF-THEN rules in the neural knowledge base	In linguistic variable and using IF-THEN rules in fuzzy structure
Learning ability	Cannot learn on its own, and update existing knowledge base	Lacks ability to learn from the experience	Cannot learn and adjust to the new environment	Neural network can learn but the learning is black box process for the user	Has learning ability because of neural network as one of the component
Uncertainty tolerance	Difficult to measure uncertainty	Probabilistic reasoning can deal with uncertainty	Not possible because of knowledge structure	Approximate reasoning	Using script value and probabilistic reasoning
Imprecision tolerance	Very low, required precise information	High, as fuzzy logic can deal with imprecision	Very low, imprecise data can lead to faulty output	Neural network component can deal with imprecise data	Very high because of combination of neural network and fuzzy logic
Explanation facility	Yes	Yes, using linguistic variable	Yes, good explanation for the output	Yes, because of rule based component	Yes, very effective
Inference Engine	Process rules and derive conclusion	Process rules using fuzzification and defuzzification	Search for goals using methods and demons	Three layer neural network combined with rule extraction	Fuzzy inference process using fuzzification and defuzzification
Knowledge update	Difficult to add new rules	Difficult to introduce new linguistic variables in existing structure	Not possible	Yes, With new knowledge, neural network component can learn	Yes, new linguistic variables can be added into existing knowledge structure
Maintainability	Moderately difficult	Very difficult	Easy	Easy	Difficult because of fuzzy component
Adaptability	No	No	No	Yes	Yes
Processing time	Very high due to each rule is processed	Processing time is reduced compare to rule base expert system	The knowledge stored in the frames can be processed very rapidly.	Learning for neural network takes time, but once knowledge is stored in neurons, then processing time for rule is fast.	Learning for neural network takes time, but then processing time is significantly reduced.
Knowledge Structure	Adhoc, cannot understand logical dependence of rules	Quite unstructured	Highly structured	Structured, but stored in neurons	Moderately structured

Categories of Expert System

Category	Problem addressed
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observations, a type of interpretation
Design	Configuring objects under constraints, such as med orders
Planning	Developing plans to achieve goals (care plans)
Monitoring	Comparing observations to plans, flagging exceptions
Debugging	Prescribing remedies for malfunctions (treatment)
Repair	Administer a prescribed remedy
Instruction	Diagnosing, debugging, and correcting student performance
Control	Interpreting, predicting, repairing, and monitoring system behavior

Developing an Expert System

The following points highlight the five main stages to develop an expert system. The stages are:

1. **Identification**
2. **Conceptualisation**
3. **Formalisation (Designing)**
4. **Implementation**
5. **Testing (Validation, Verification and Maintenance).**

A **knowledge engineer** is an AI specialist, perhaps a computer scientist or programmer, who is skilled in the 'Art' of developing expert systems. A **domain expert** is an individual who has significant expertise in the domain of the expert system being developed. The **knowledge engineer and the domain expert** usually work very closely together for long periods of time throughout the several stages of the development process.
An expert system is developed and refined over a period of several years since it is typically computer-based software. It divides the process of expert system development into five distinct stages since an expert system is typically a computer based system.

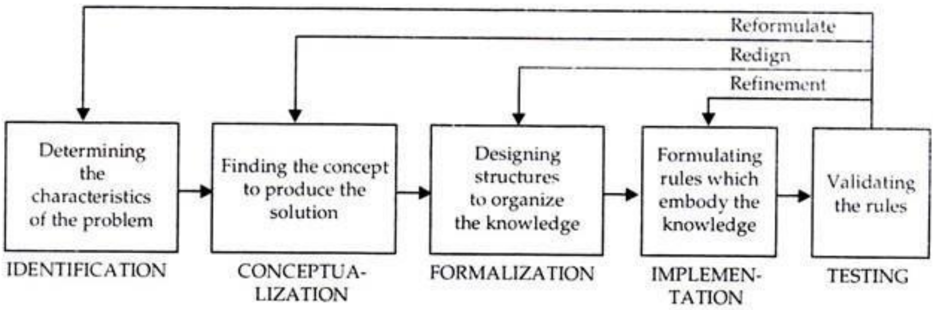


Fig: Stages in Developing an Expert System

Stage # 1— Identification:

Before beginning to develop an expert system, it is important to describe, with as much precision as possible, the problem which the system is intended to solve. We must determine the exact nature of the problem and state the precise goals which indicate exactly how the expert system is expected to contribute to the solution.

I**nitially the knowledge engineer** consults manuals and training guides to gain some familiarity with the subject **then the domain expert** describes several typical problem states. The knowledge engineer attempts to extract fundamental concepts from the similar cases in order to develop a more general idea of the purpose of the expert system.

After the domain expert describes several cases, the knowledge engineer develops a 'first-pass' problem description. When the domain expert feel that the description does not entirely represent the problem then he suggest changes to the description and provides the knowledge engineer with additional examples to illustrate further the problem's fine points. **Next, the knowledge engineer revises the description, and the domain expert suggests further changes.** This process is repeated until the domain expert is satisfied that the knowledge engineer understands the problems and until both are satisfied that the description adequately portrays the problem which the expert system is expected to solve.

This 'iterative' procedure is typical part of the entire expert-system development process. The results are evaluated at each stage of the process and compared to the expectations. If the results do not meet the expectations, adjustments are made to that stage of the process, and the new results are evaluated. The process continues until satisfactory results are achieved.

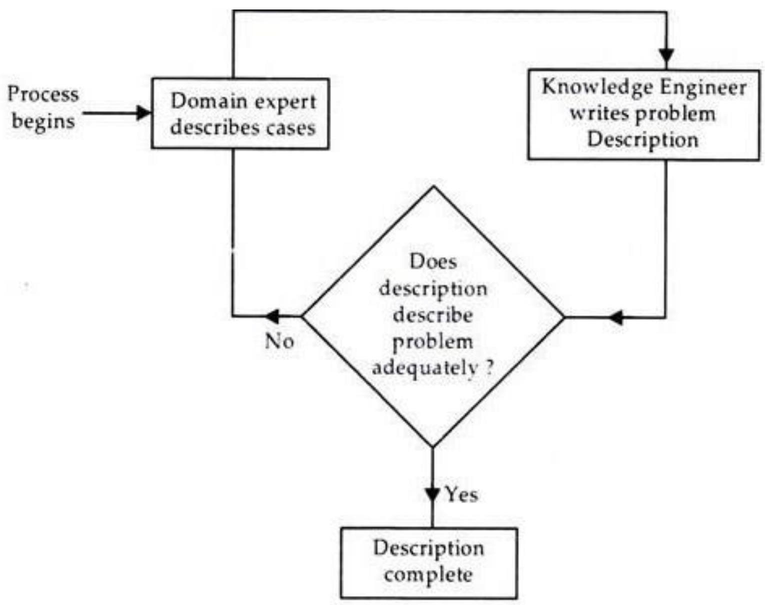


Fig: Iterative process for identifying the problem

It is also important to identify our resources like —

- Who is to participate in the development process?
- Does a single domain expert possess all the necessary expertise, or is the domain knowledge distributed over several people in an organisation?
- Can a single knowledge engineer develop the system in a timely fashion, or is it necessary to provide additional technical assistance?

- ***Domain experts are not the only resources which must be identified. It is unusual for all domain knowledge to be embodied in human experts; therefore, more tangible sources of information, such as reference books and manuals, are usually identical and located.***

Stage # 2— Conceptualisation:

Once it has been identified for the problem an expert system is to solve, the next stage involves analysing the problem further to ensure that its specifics, as well as generalities, are understood. In the conceptualisation stage, **the knowledge engineer frequently creates a diagram of the problem to depict graphically the relationships between the objects and processes in the problem domain. It is often helpful at this stage to divide the problem into a series of sub-problems and to diagram both the relationships among the pieces of each sub-problem and the relationships among the various sub-problems.**

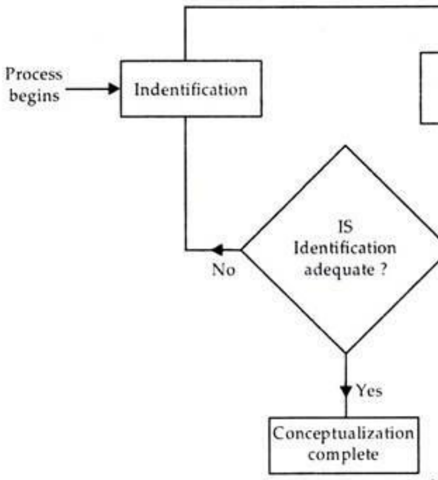


Fig: Iterative process between identification and conceptualization stages

The conceptualisation stage involves a circular procedure of iteration and reiteration between the knowledge engineer and the domain expert. When both agree that the key concepts and the relationships among them-have been adequately conceptualised, this stage is complete.

- ***Not only each stage in the expert system development process circular, the relationships among the stages may be circular as well. Since each stage of the development process adds a level of detail to the previous stage, any stage may expose a weakness in a previous stage.***

Stage # 3— Formalisation (Designing):

In the preceding stages, no effort has been made to relate the domain problem to the artificial intelligence technology which may solve it. **During the identification and formalization stages, the focus is entirely on understanding the problem**. Now, **during the formalization stage, the problem is connected to its proposed solution**, an expert system is supplied by analysing the relationships depicted in the conceptualization stage. The knowledge engineer begins to select the techniques which are appropriate for developing this particular expert system.

During formalization, it is important that the knowledge engineer be familiar with the following:

1. The various techniques of knowledge representation and intelligent search techniques used in expert systems.
2. The expert system tools which can greatly expedite the development process.
3. Other expert systems which may solve similar problems and thus may be adaptable to problem at hand.

It is desirable to select a single development technique or tool which can be used throughout all segments of the expert system. However, the knowledge engineer may determine that no particular technique is appropriate for the entire expert system, making it necessary to use different techniques for different sub-problems. Once it has been determined which technique(s) will be used the knowledge engineer starts to develop a formal specification which can be used to develop a prototype expert system.

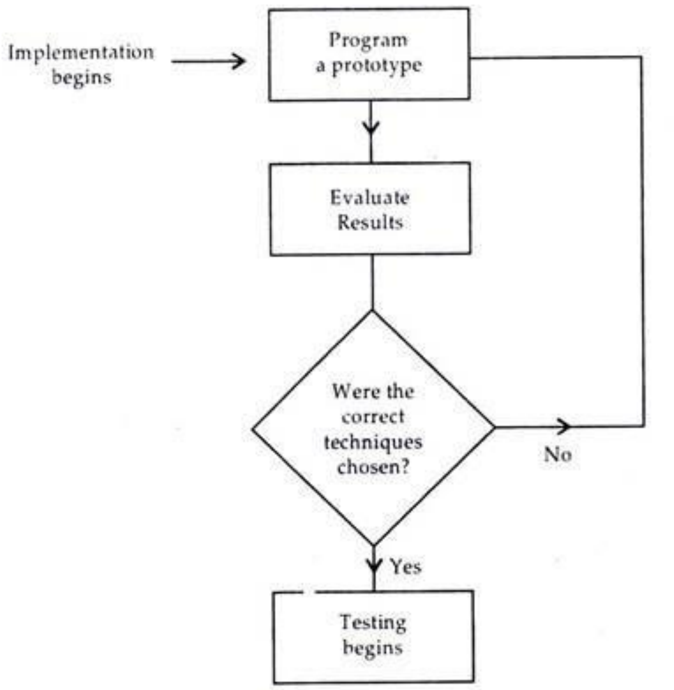
In the case of a rule-based system, for example, the knowledge engineer develops a set of rules designed to represent the knowledge communicated by the domain expert. This is a critical part of the development process requiring great skill on the part-of the knowledge engineer. Many domain experts can explain what they do but not why; therefore, one of the knowledge engineer's primary responsibilities is to analyse example situations and filter in from those examples a set of rules which describe the domain expert's knowledge.

The formalisation process is often the most interactive stage of expert system development, as well as the most time consuming. The knowledge engineer must develop a set of rules and ask the domain expert if those rules adequately represent the expert's knowledge. The domain expert reviews the rules proposed by the knowledge engineer and suggests changes, which are then incorporated into the knowledge base by the knowledge engineer. As in the other development stages, this process also is iterative: the rule review is repeated and the rules are refined continually until the results are satisfactory.

Stage # 4— Implementation:

During the implementation stage the formalised concepts are programmed into the computer which has been chosen for system development, using the predetermined techniques and tools to implement a 'first-pass' (prototype) of the expert system.

Theoretically, if the methods of the previous stages have been followed with diligence and care, the implementation of the prototype should proceed smoothly. In practice, the development of an expert system may be as much an art as it is a science, because following all the rules does not guarantee that the system will work the first time it is implemented. In fact, experience suggests the opposite. Many scientists actually consider the prototype to be a 'throw-away' system, useful for evaluating progress but hardly a usable expert system.



<u>Fig: Implementation stage of an Expert System Development</u>

If the prototype works at all, the knowledge engineer may be able to determine if the techniques chosen to implement the expert system were the appropriate ones. On the other hand, the knowledge engineer may discover that the chosen techniques simply cannot be implemented. It may not be possible, for example, to integrate the knowledge representation techniques selected for different sub-problems. At that point, the concepts may have to be re-formalised, or it even may be necessary to create new development tools to implement the system efficiently. Once the prototype system has been refined sufficiently to allow it to be executed, the expert system is ready to be tested thoroughly to ensure that it expertise's correctly.

<u>Stage # 5— Testing (Validation, Verification and Maintenance)</u>:

The chance of prototype expert system executing flawlessly the first time it is tested are so slim as to be virtually non-existent. A knowledge engineer does not expect the testing process to verify that the system has been constructed entirely correctly. Rather, testing provides an opportunity to identify the weaknesses in the structure and implementation of the system and to make the appropriate corrections.

Depending on the types of problems encountered, the testing procedure may indicate that the system was implemented incorrectly or perhaps that the rules were implemented correctly but were poorly or incompletely formulated. **Results from the tests are used as 'feedback' to return to a previous stage and adjust the performance of the system.**

Once the system has proven to be capable of correctly solving straight-forward problems, the domain expert suggests complex problems which typically would require a great deal of human expertise. These more demanding tests should uncover more serious flaws and provide ample opportunity to 'fine tune' the system even further. Ultimately, **an expert system is judged to be entirely successful only when it operates at the level of a human expert. The testing process is not complete until it indicates that the solutions suggested by the expert system are consistently as valid as those provided by a human domain expert.**