Unit-5

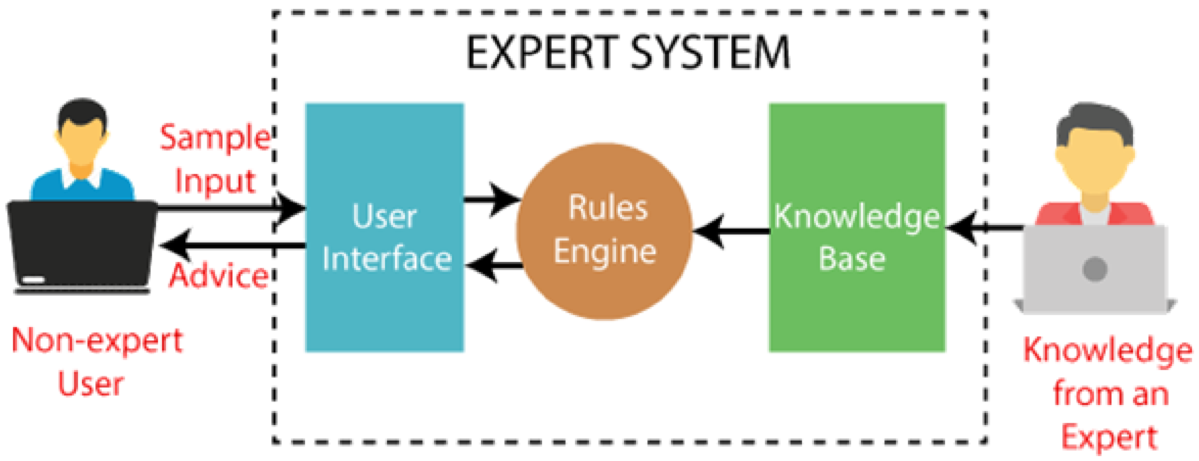
Expert Systems:

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for complex problems using both facts and heuristics like a human expert. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as medicine, science, etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:



Architecture of expert systems

Typical expert system architecture is shown in Figure.

The knowledge base contains the specific domain knowledge that is used by an expert to derive conclusions from facts.

In the case of a rule-based expert system, this domain knowledge is expressed in the form of a series of rules.

The explanation system provides information to the user about how the inference engine arrived at its conclusions. This can often be essential, particularly if the advice being given is of a critical nature, such as with a medical diagnosis system.

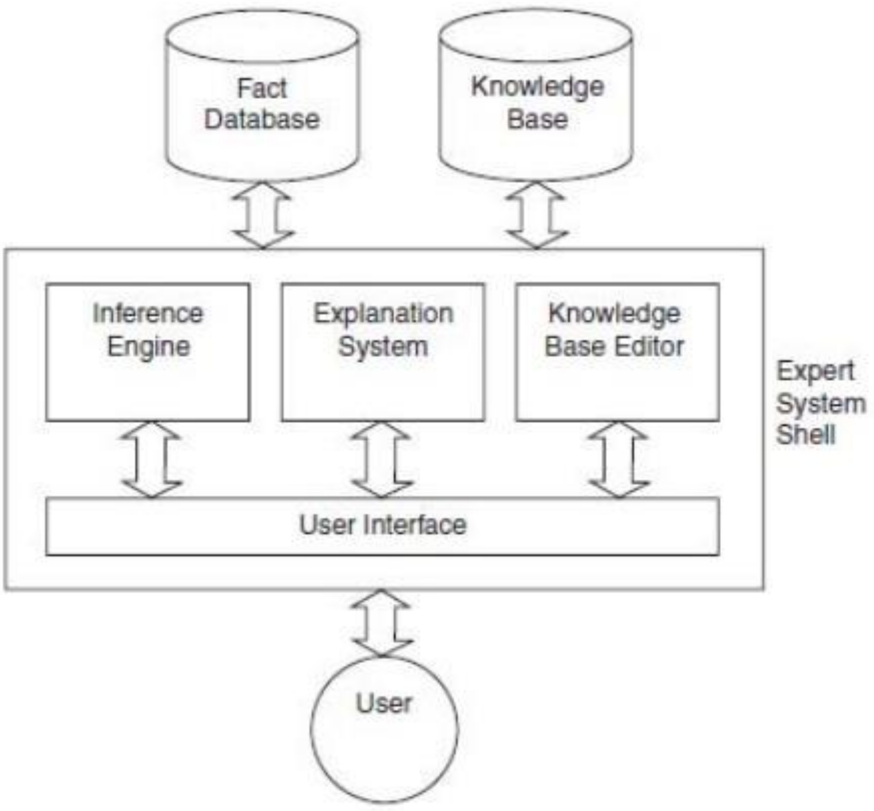


Fig Expert System Architecture

If the system has used faulty reasoning to arrive at its conclusions, then the user may be able to see this by examining the data given by the explanation system.

The fact database contains the case-specific data that are to be used in a particular case to derive a conclusion.

In the case of a medical expert system, this would contain information that had been obtained about the patient’s condition.

The user of the expert system interfaces with it through a user interface, which provides access to the inference engine, the explanation system, and the knowledge-based editor.

The inference engine is the part of the system that uses the rules and facts to derive conclusions. The inference engine will use forward chaining, backward chaining, or a combination of the two to make inferences from the data that are available to it.

The knowledge-based editor allows the user to edit the information that is contained in the knowledge base.

The knowledge-based editor is not usually made available to the end user of the system but is used by the knowledge engineer or the expert to provide and update the knowledge that is contained within the system.

Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics

Knowledge Acquisition

Knowledge acquisition refers to the process of extracting, structuring, and organizing knowledge from various sources, such as human experts, books, documents, sensors, or computer files, so that it can be used in software applications, particularly knowledge-based systems. This process is crucial for the development of expert systems, which are AI systems that emulate the decision-making abilities of a human expert in a specific domain.

The knowledge acquired can be in the form of facts, relationships, heuristics, and rules that are essential for competent performance within a given field. This knowledge is then encoded into a knowledge base, which is a component of a knowledge-based system that uses the encoded expertise to solve complex problems by reasoning through the knowledge.

Various methods are employed for knowledge acquisition, including rule-based systems, decision trees, artificial neural networks, and fuzzy logic systems. The process can be challenging due to the complexity of human knowledge and the dynamic nature of data, which can change over time, making it difficult to keep AI models up-to-date.

Knowledge acquisition is not only a technical process but also an area of intense research, as it involves understanding how to effectively capture and represent knowledge in a way that AI systems can utilize. It is a key component of machine learning and is essential for the continuous improvement of AI systems' accuracy and effectiveness.

Understanding Knowledge Acquisition in AI

Knowledge acquisition in artificial intelligence (AI) involves collecting, selecting, and interpreting data to build and update knowledge in a specific domain. It's integral to machine learning and knowledge-based systems, where it enables AI to solve problems more efficiently and effectively. The process is not without challenges, such as determining relevant information, managing the time and cost of data collection, and ensuring data quality.

Various methods exist for knowledge acquisition, each suited to different problems and data types. Rule-based systems apply heuristics to make decisions, while decision trees follow a structured path of if-then-else statements. Artificial neural networks, inspired by the human brain, learn from data to make predictions, and fuzzy logic systems use fuzzy set theory to handle uncertainty in decision-making.

Expert systems rely on domain-specific knowledge provided by human experts. Machine learning models generalize from training data to make predictions on new data. Natural language processing extracts knowledge from textual data, and the semantic web uses standards like RDF and OWL to represent knowledge online. Knowledge representation and reasoning formalize knowledge for automated reasoning.

The interpretation of information is crucial, requiring human expertise to select and process data correctly. AI can assist by automating knowledge acquisition, enhancing speed and accuracy. However, the data used to train AI models must be accurate, representative, and up-to-date, which can be costly and complex. Despite these challenges, advancements in AI continually improve knowledge acquisition techniques, aiding in the development of more effective AI systems.

In practice, knowledge acquisition is about selecting high-quality data sources and processing them into a usable format through feature engineering. The choice of AI model—such as a convolutional neural network for image recognition—depends on the task. Once trained, AI systems can be deployed in real-world environments where they continue to learn and adapt through transfer learning, ensuring ongoing improvement and relevance.

Meta Knowledge In AI

Meta-knowledge in AI refers to knowledge about knowledge itself. In the context of artificial intelligence, meta-knowledge can include understanding how knowledge is acquired, how it can be represented, and how it can be manipulated or refined.

In AI systems, this can involve using higher-level reasoning to guide the learning process, determine the relevance of information, or even reflect on the system’s own reasoning process. For example, a machine learning model could use meta-knowledge to determine which learning strategy to use, given the particular task and data at hand, or to decide when to seek additional information from external sources.

In essence, meta-knowledge provides a way for an AI system to be more intelligent and adaptable by using knowledge about its own reasoning processes and the world around it to make more informed decisions. This is an ongoing area of research in AI and has significant implications for the development of more robust and autonomous AI systems.

Heuristics

Heuristics is a problem-solving or decision-making technique that uses minimum relevant information, past results, and experiences to produce a workable and practical solution for a problem in a reasonable time. These strategies focus on providing quick results with an acceptable accuracy range rather than offering near-perfect solutions.

Heuristics comprises vital ingredients of the machine learning (ML) and artificial intelligence (AI) disciplines. It is a go-to approach when it is highly impractical to derive a solution for a problem by following a step-by-step algorithm. Moreover, as heuristic strategies look to provide speedy solutions rather than accurate ones, they are generally blended with optimization algorithms to improve the results.

Technically, all iterations are interdependent, implying that each level of a deep neural network is crucial in deciding which solution path to choose and which to discard based on their closeness to the desired result. Thus, the term ‘heuristics’ is synonymous with ‘short-cut’, since it does not employ resources to explore solution paths that do not yield acceptable results.

Heuristic methods in AI are based on cognitive science principles that revolve around ‘how humans think’. Moreover, heuristic algorithms in AI enable systems to produce approximate solutions rather than exact ones. Heuristics do not necessarily provide a cheaper solution. Instead, the ones that do not overestimate the cost of achieving the result are termed ‘admissible heuristics’. This is a crucial characteristic of heuristics that ensures the solution’s optimality. At the fundamental level, an admissible heuristic simplifies the original problem by reducing its constraints.

Although heuristic processes tend to find solutions or results that often work or are correct, they may only sometimes be right, provable, optimal, or accurate. However, decisions based on heuristics are usually good enough to solve small-scale problems and provide solutions in situations of uncertainty where complete information is unavailable.

Heuristics rely on shortcuts to provide immediate, efficient, and short-term solutions that facilitate timely decisions for businesses. Analysts across industries use specific thumb rules that allow companies to address a problem and make decisions and judgments rapidly and efficiently. These include the process of trial and error, elimination, intelligent guesswork, past results or formulas, and even the analysis of historical data. However, in the computing world, a heuristic model acts as a rule of thumb to speed up and simplify decision-making processes in situations where there’s not enough time for careful consideration of all the aspects of the problem.

Typical expert systems

MYCIN

Overview: MYCIN is one of the earliest and most influential expert systems developed in the 1970s. It was specifically designed for medical diagnosis.

Functionality: MYCIN uses backward chaining to diagnose bacterial infections, such as meningitis and bacteremia. It identifies the bacteria causing the infection by asking the doctor a series of questions about the patient’s symptoms and test results.

Significance: Although not used clinically, MYCIN greatly influenced the development of medical expert systems.

MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections and to recommend antibiotics, with the dosage adjusted for patient’s body weight.

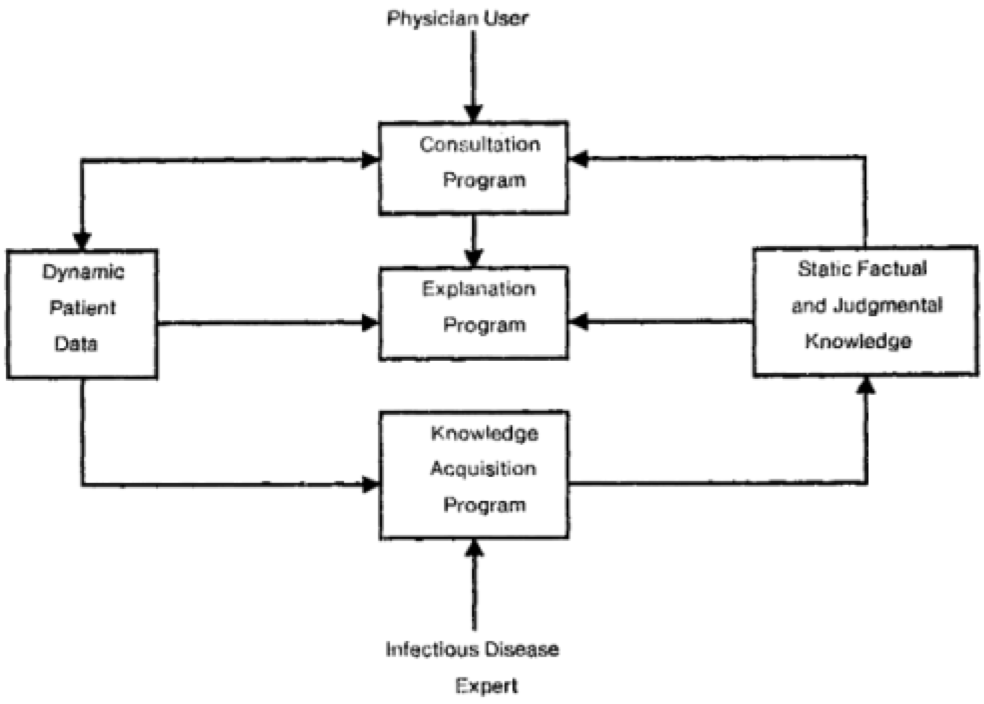
MYCIN was developed over five or six years in the early 1970s at Stanford University.

It was written in Lisp

MYCIN operated using fairly simple inference engine, and a knowledge base of approximately 600 rules.

MYCIN is a computer program designed to provide attending physicians with advice comparable to that which they would otherwise get from a consulting physician specializing in bacteremia and meningitis infections. To use MYCIN, the attending physician must sit in front of a computer terminal that is connected to a DEC-20 (one of Digital Equipment Corporation's mainframe computers) where the MYCIN program is stored. When the MYCIN program is evoked, it initiates a dialogue. The physician types answers in response to various questions. Eventually MYCIN provides a diagnosis and a detailed drug therapy recommendation.

Structure of Mycin Program:


Physician User
Consultation Program
Dynamic Patient Data
Explanation Program
Static Factual and Judgmental Knowledge
Knowledge Acquisition Program
Infectious Disease Expert

The MYCIN system comprises three major subprograms, as depicted in Figure above.

- The Consultation Program
- Explanation Program
- Knowledge Acquisition Program

The Consultation Program is the core of the system; it interacts with the physician to obtain information about the patient, generating diagnoses and therapy recommendations.

The Explanation Program provides explanations and justifications for the program's actions.

The Knowledge-Acquisition Program is used by experts to update the system's knowledge base.

DART

DART an application of artificial intelligence techniques that is use for computer system fault diagnosis. It is an automated consultant that advises IBM field service personnel on the diagnosis of faults occurring in computer installations.

Scope of DART

- A typical, large-scale computer installation is composed of numerous subsystems including CPUs, primary and secondary storage, peripherals and supervisory software.
- Each of these subsystems, in turn, consists of a richly connected set of both hardware and software components such as disk drives, controllers, CPUs, memory modules, and access methods.
- Generally, each individual component has an associated set of diagnostic aids designed to test its own specific integrity.
- However, very few maintenance tools and established diagnostic strategies are aimed at identifying faults on the system or subsystem level.
- As a result, identification of single or multiple faults from systemic manifestations remains a difficult task.
- The non-specialist field service engineer is trained to use the existing component-specific tools as a result, is often unable to attack the the failure at the systemic level.
- Expert assistance is then required, increasing both the time and cost required to determine and repair the fault.
- The design of DART reflects the expert's ability to take a systemic viewpoint on problems and to see that viewpoint to indicate a specific components, thus making more effective use of the existing maintenance capabilities.

For our initial design, we choose to concentrate on problems occurring within the teleprocessing (TP) subsystems for the IBM 370-class computers.This subsystem includes various network controllers, terminals, remote,. entry facilities, modems, and several software access methods. In addition to these well-defined components there are numerous available test points the program can use during diagnosis. Here we have focused our effort on handling two of the most frequent TP problems,

1. When a user is unable to log on to the system from a remote terminal.
2. When the system operator is unable to initialize the TP network itself.

In a new system configuration, these two problems constitute a significant percentage of service calls received. Interviews with field-service experts made it apparent that much of their problem-solving expertise is derived from their knowledge of several.

DART Consultation:

- During a DART consultation session, the field engineer focuses on a particular computer system that is experiencing a problem.
- Within each such system, the user describes one or more problems by indicating a failure symptom, currently using a list of keywords.
- Using this description, the consultant makes an initial guess about which of the major subsystems might be involved in the fault.
- The user is then given the opportunity to select which of these implicated subsystems are to be pursued and in which order.
- Each subsystem serves as a focal point for tests and findings.
- These subsystems currently correspond to various input/output facilities or the CPU-complex itself.
- For each selected subsystem, the user is asked to identify one or more logical pathways which might be involved in the situation.
- Logical pathways correspond to a line of communication between a peripheral and an application program.
- On the basis of this information and details of the basic composition of the network, the appropriate communications protocol can be selected.
- Once the logical pathway and protocol have been determined, descriptions are gathered about the multiple physical pathways that actually implement the logical pathway.
- It is on this level that diagnostic test results are presented and actual component indictments occur.

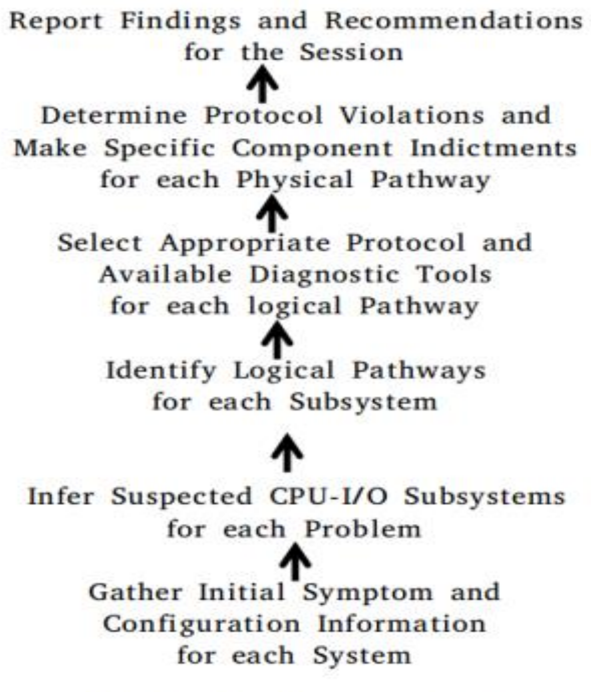

Report Findings and Recommendations for the Session
Determine Protocol Violations and Make Specific Component Indictments for each Physical Pathway
Select Appropriate Protocol and Available Diagnostic Tools for each logical Pathway
Identify Logical Pathways for each Subsystem
Infer Suspected CPU-I/O Subsystems for each Problem
Gather Initial Symptom and Configuration Information for each System

Fig: The DART Diagnostic Inference Process

For DART to be useful, the field engineer must be familiar with the diagnostic equipment and software testing and tracing facilities which can be requested, and of course, must also have access to information about the specific system hardware and software configuration of the installation. Finally, at the end of the consultation session, DART summarizes its findings and recommends additional tests and procedures to follow. Figure below depicts the major steps of the diagnostic process outlined above.

R1/XCON

Overview: R1, also known as XCON, was developed in the late 1970s by Digital Equipment Corporation (DEC) and is one of the most commercially successful expert systems.

Functionality: R1/XCON was used to configure orders for new computer systems. It would select the appropriate hardware and software components based on the customer’s requirements.

Significance: R1/XCON streamlined system configuration, saving DEC millions by reducing errors and improving efficiency.

XCON is a Rule-based system written in OPS5 in 1978 to assist in the ordering of DEC’s VAX computer systems by automatically selecting the computer system components based on the customer’s requirements.

In 1975 DEC offered 50 types of central processors with 400 core operations. The estimated possible number of configurations at that time was already in millions. So system configuration was the key process in DEC`s flexibility strategy for it converted a customer order into fully configured system that was designed, checked and ready for delivery. This process involves three separate reviews of each order.

- The first two steps relied upon highly skilled and talented technical editors who learned their craft through a long apprenticeship.
- The final review was FA & T (Fast Assembly and Test) which is nothing but actual assembly of the system prior to delivery.

Elapsed time from signed order to delivery was ten to fifteen weeks, extending at times even up to six months. Computers too were growing more complex, increasing even further the number of configuration options. DEC had to find a new way to configure its order and so XCON DEC`s configuration system was born.

Functions of XCON System:

XCON is used to configure customer orders and to guide the assembly of these orders at the customer site. Using the customer order as input, it provides the following functionality:

- Configures CPUs, memory, cabinets, power supplies, disk, tapes and so on.
- Determines and list cabling information.
- List components ordered with configuration related comments.

- Generate warning messages on issues affecting technical validity.

Scope of XCON

The initial purpose of XCON was to assist manufacturing plant personnel in validating the technical correctness of system orders about to be filled. It is now used by a broad set of users across the company's major functions like sales and marketing, manufacturing and production, field service and engineering. The users of these systems perform functions that span Digital`s complete order flow and manufacturing cycle. Thus they are involved with many different business processes. Each has different needs and takes a different perspective on the configuration information provided.

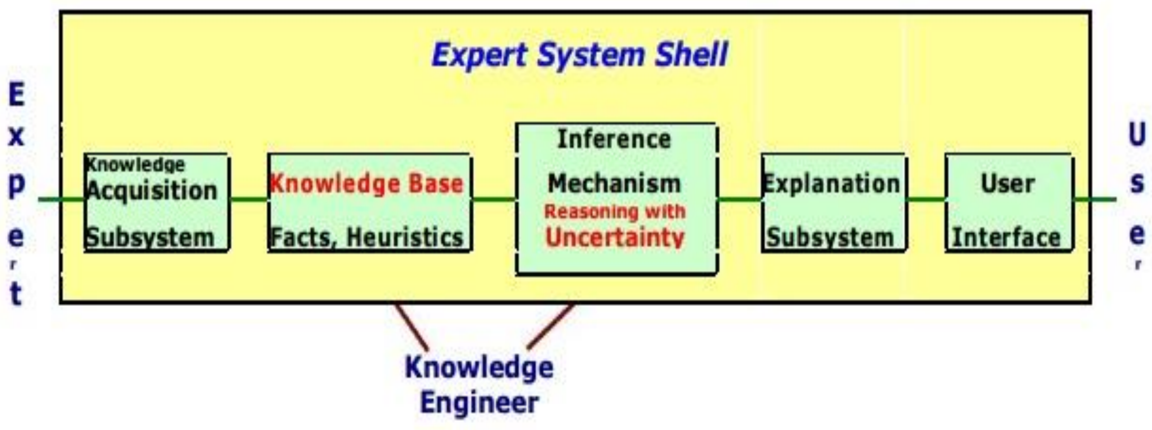
- Sales uses the configuration systems as an integral part of the automated process to generate quotations for customer
- Manufacturing uses the information to verify buildability of all incoming orders.
- Field service has the perspective of assembling the order in the customer's unique environment.
- Manufacturing and engineering benefits from the configuration system`s focus on system integration and can identify potential problems in system-level design and manufacturability.

Expert systems shells

An Expert system shell is a software development environment. It contains the basic components of expert systems. A shell is associated with a prescribed method for building applications by configuring and instantiating these components.

Shell components and description

The generic components of a shell: the knowledge acquisition, the knowledge Base, the reasoning, the explanation and the user interface are shown below. The knowledge base and reasoning engine are the core components.


Expert System Shell
Expert
Knowledge Acquisition Subsystem
Knowledge Base Facts, Heuristics
Inference Mechanism Reasoning with Uncertainty
Explanation Subsystem
User Interface
User
Knowledge Engineer

All these components are described below.

Knowledge Base

A store of factual and heuristic knowledge. Expert system tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both Frames (objects) and IF-THEN rules. In PROLOG the knowledge is represented as logical statements.

Reasoning Engine

Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base form a line of reasoning in solving a problem. The inference mechanism can range from simple modus ponens backward chaining of IF-THEN rules to Case-Based reasoning.

Knowledge Acquisition subsystem

A subsystem to help experts in build knowledge bases. However, collecting knowledge, needed to solve problems and build the knowledge base, is the biggest bottleneck in building expert systems.

Explanation subsystem

A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at justifying the need for additional data.

User Interface

A means of communication with the user. The user interface is generally not a part of the expert system technology. It was not given much attention in the past. However, the user interface can make a critical difference in the perceived utility of an Expert system.