Expert Systems

Expert Systems

- Expert Systems solves problems that are normally solved by human experts
- An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice.
- To solve expert-level problems, expert systems will need efficient access to a substantial domain knowledge base which must be built as efficiently as possible
- They also need to exploit one or more reasoning mechanisms to apply their knowledge to the problems they are given.
- They will also need to be able to explain, to the users who rely on them, how they have reached their decisions.

Typical tasks for expert systems

Some typical existing expert system tasks include:

1. The interpretation of data Such as sonar data or geophysical measurements

2. Diagnosis of malfunctions Such as equipment faults or human diseases

3. Structural analysis or configuration of complex objects Such as chemical compounds or computer systems

4. Planning sequences of actions Such as might be performed by robots

5. Predicting the future Such as weather, share prices, exchange rates

Characteristics of Expert Systems

1. They simulate human reasoning about the problem domain, rather than simulating the domain itself.

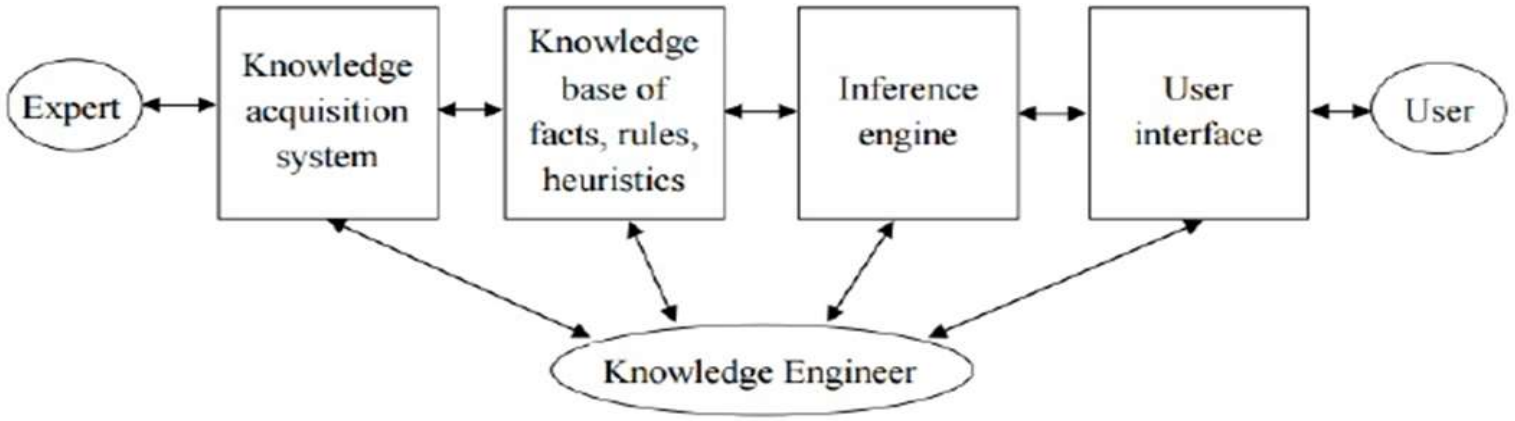
2. They perform reasoning over representations of human knowledge, in addition to doing numerical calculations or data retrieval. They have corresponding distinct modules referred to as the inference engine and the knowledge base.

3. Problems tend to be solved using heuristics (rules of thumb) or approximate methods or probabilistic methods which, unlike algorithmic solutions, are not guaranteed to result in a correct or optimal solution.

4. They usually have to provide explanations and justifications of their solutions in order to convince the user that their reasoning is correct.

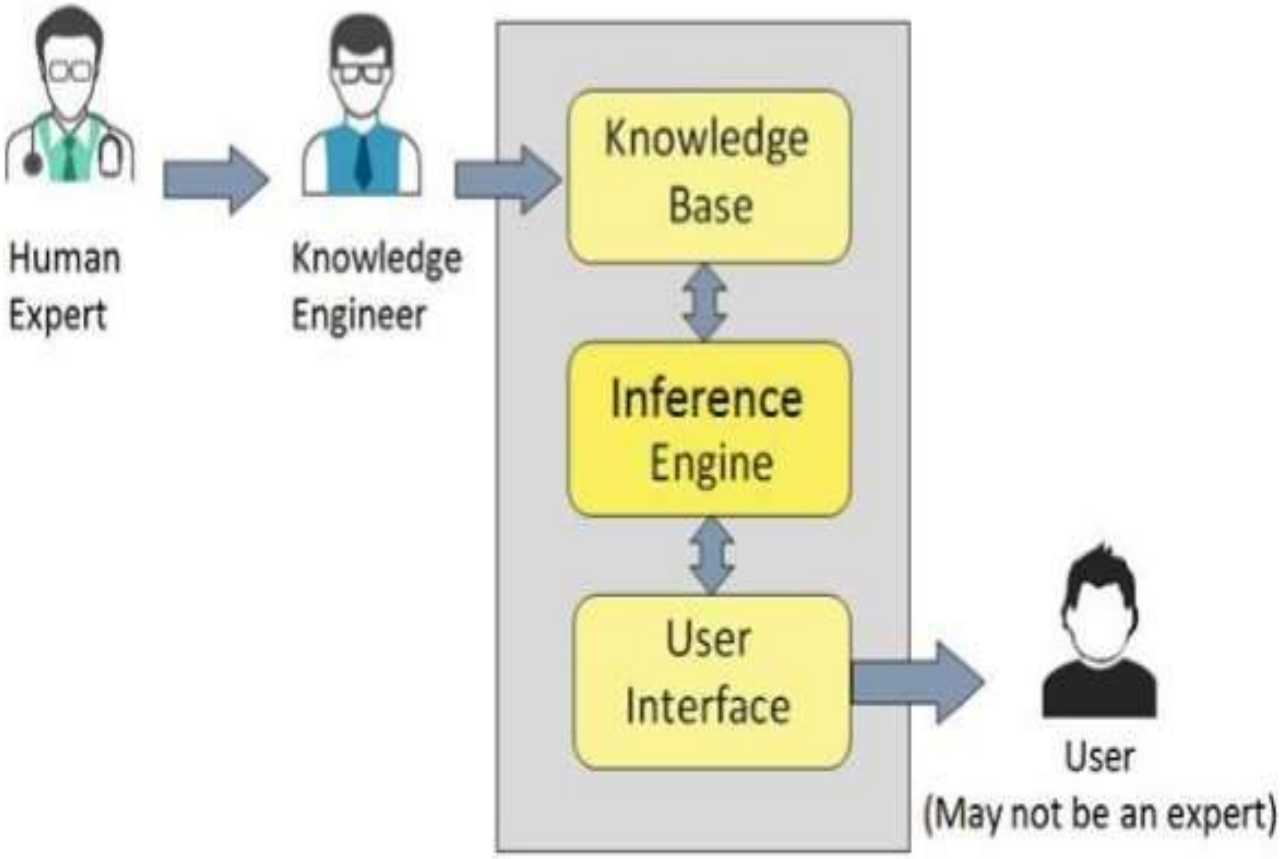
The term Intelligent Knowledge Based System (IKBS) is sometimes used as a synonym for Expert System

Architecture of Expert Systems



- The process of building expert systems is often called knowledge engineering.
- The knowledge engineer is involved with all components of an expert system:
- Building expert systems is generally an iterative process.
- The components and their interaction will be refined over the course of numerous meetings of the knowledge engineer with the experts and users.

Architecture



Knowledge acquisition

- The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base.
- The knowledge acquisition component allows the expert to enter their knowledge or expertise into the expert system, and to refine it later as and when required.
- Historically, the knowledge engineer played a major role in this process.
- The knowledge acquisition process is usually comprised of three principal stages:
  - Knowledge elicitation is the interaction between the expert and the knowledge engineer/program to elicit the expert knowledge in some systematic way.
  - The knowledge thus obtained is usually stored in some form of human friendly intermediate representation.
  - The intermediate representation of the knowledge is then compiled into an executable form (e.g. production rules) that the inference engine can process.

knowledge elicitation

It consists of several stages:

1. Find as much as possible about the problem and domain from books, manuals, etc. In particular, become familiar with any specialist terminology.
2. Try to characterize the types of reasoning and problem solving tasks that the system will be required to perform.
3. Find an expert (or set of experts) that is willing to collaborate on the project. Sometimes Interview the expert (usually many times during the course of building the system). Find out how they solve the problems your system will be expected to solve.
4. Have them check and refine your intermediate knowledge representation.

Inference Engine

It is the brain of the Expert System.

1. Match the premise patterns of the rules against elements in the working memory. Generally the rules will be domain knowledge built into the system, and the working memory will contain the case based facts entered into the system, plus any new facts that have been derived from them.

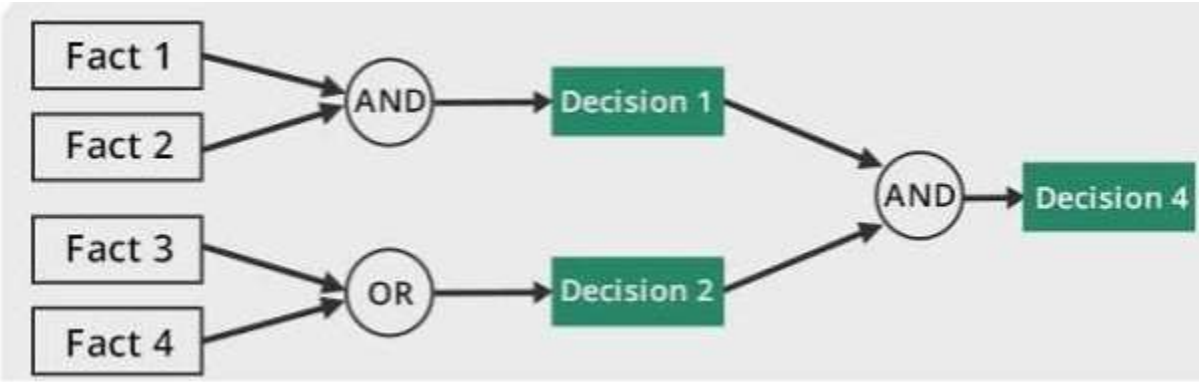
2. If there is more than one rule that can be applied, use a conflict resolution strategy to choose one to apply. Stop if no further rules are applicable.

3. Activate the chosen rule, which generally means adding/deleting an item to/from working memory. Stop if a terminating condition is reached, or return to step 2.

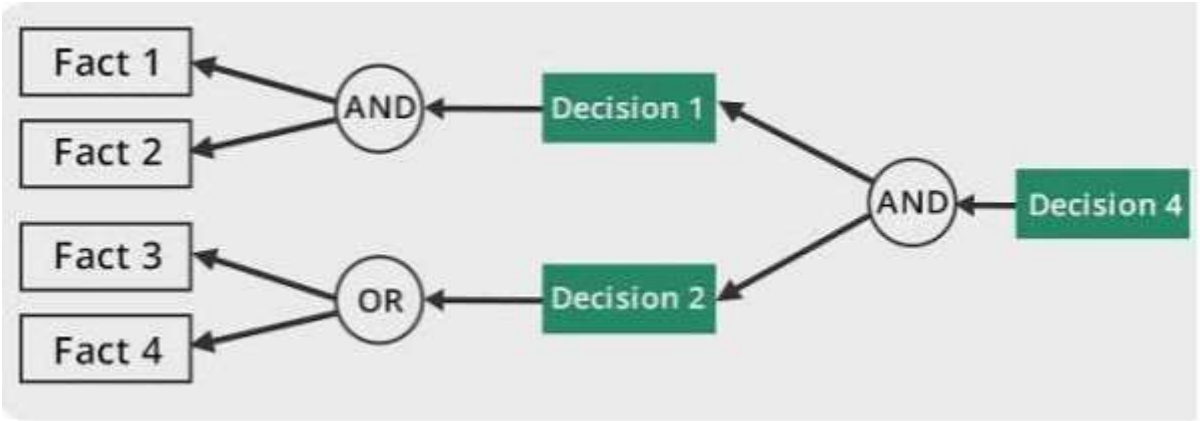
- Early production systems spent over 90% of their time doing pattern matching.

Chaining

To recommend a solution, the Inference Engine uses the following strategies –Forward Chaining, Backward Chaining



Forward



Backward

Forward Chaining

- Forward Chaining -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

- Backward Chaining - 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.

User interface

The Expert System user interface usually comprises of two basic components:

1. The Interviewer Component : This controls the dialog with the user and/or allows any measured data to be read into the system. For example, it might ask the user a series of questions, or it might read a file containing a series of test results.

2. The Explanation Component : This gives the system's solution, and also makes the system's operation transparent by providing the user with information about its reasoning process.

Meta Knowledge

- Meta knowledge can be simply defined as knowledge about knowledge.

- Meta knowledge is knowledge about the use and control of domain knowledge in an expert system.

Roles in an Expert System

Three fundamental roles in building expert systems are:

1.Expert

- Successful ES systems depend on the experience and application of knowledge that the people can bring to it during its development. Large systems generally require multiple experts.

2. Knowledge engineer –

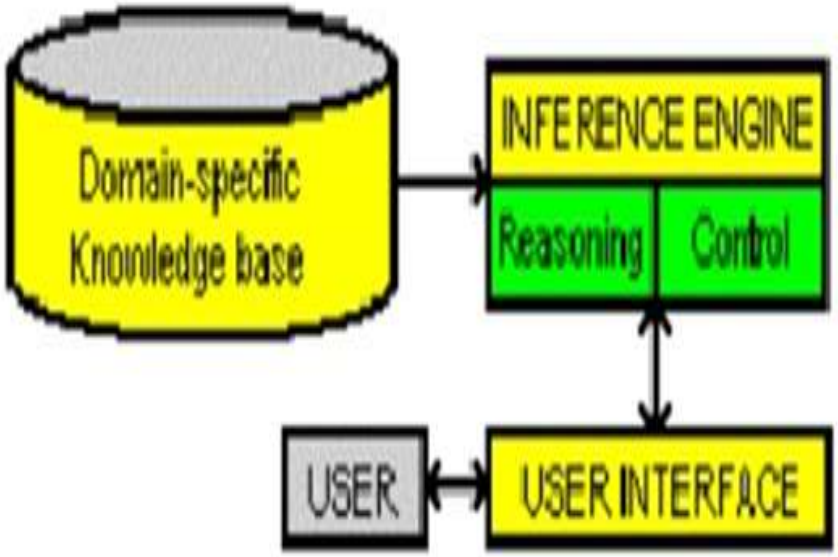
-This person should be able to elicit knowledge from the expert, gradually gaining an understanding of an area of expertise.

- Intelligence, tact, empathy, and proficiency in specific techniques of knowledge acquisition areall required of a knowledge engineer

3. User

- A system developed by an end user with a simple shell, is built rather quickly an inexpensively.

Typical Expert System

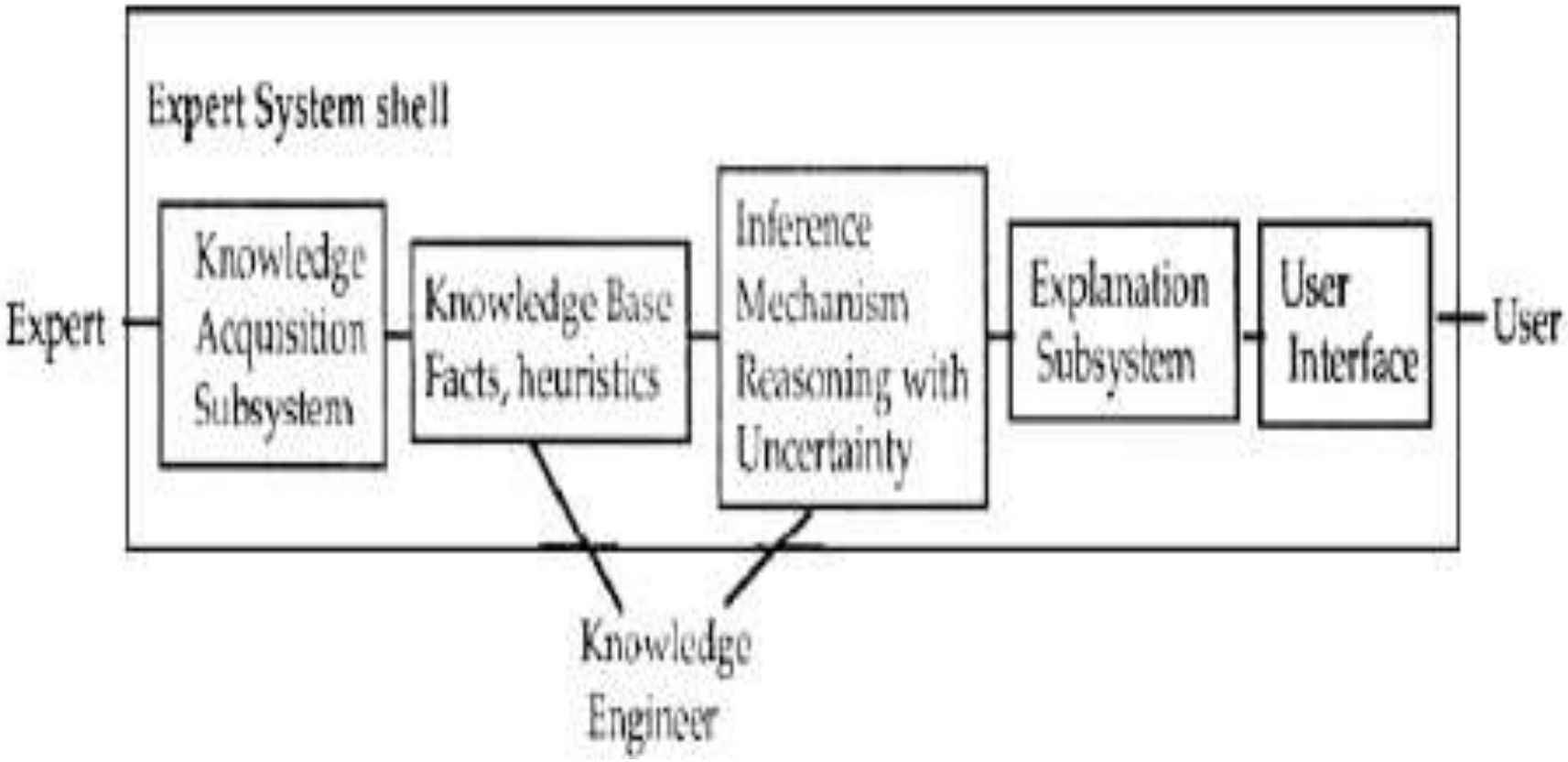


Typical Expert System

1. In a rule-based expert system, the knowledge base includes the if-then rules and additional specifications that control the course of the interview.
2. An inference engine is a set of rules for making deductions from the data and that implements the reasoning mechanism and controls the interview process.

   The inference engine might be generalized so that the same software is able to process many different knowledge bases.
3. The user interface requests information from the user and outputs intermediate and final results

Expert System Shell



Expert System Shells

- Initially each expert system is build from scratch (LISP).
- Expert System Shells are a collection of software packages & tools used to develop expert systems
- A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility.
- Example of shell is EMYCIN (for Empty MYCIN derived from MYCIN).
- Expert system shells CLIPS, JESS, DROOLS

Shells

- Shells provide greater flexibility in representing knowledge and in reasoning
- Expert system shells need to integrate with other programs easily.
- Expert systems cannot operate in vacuum.
- The shells must provide an easy-to-use interface between an expert system written with the shell and programming environment.

MYCIN

- MYCIN is used for Disease DIAGNOSIS and Therapy SELECTION
- MYCIN would attempt to diagnose patients based on reported symptoms and medical test results.
- The program could request further information concerning the patient, as well as suggest additional laboratory tests, to arrive at a probable diagonosis, after which it would recommend a course of treatment.
- If requested, MYCIN would explain the reasoning that led to its diagonosis and recommendation.
- Using about 500 production rules, MYCIN operated at roughly the same level of competence as human specialists in blood infections and rather better than general practitioners.
- MYCIN was written in LISP

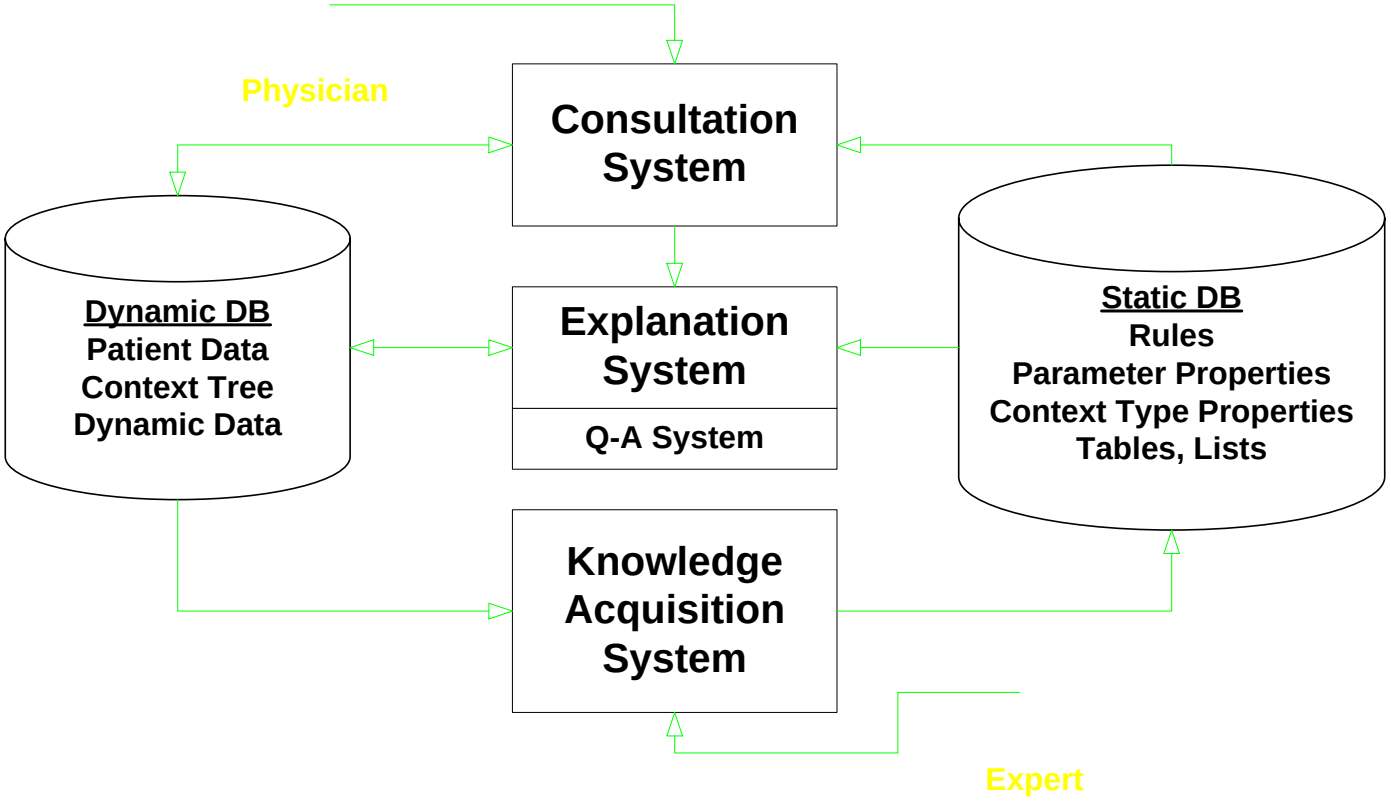
MYCIN

- Represent Domain-specific Knowledge
- Over 450 rules in MYCIN
- Premise-Action (If-Then) Form:

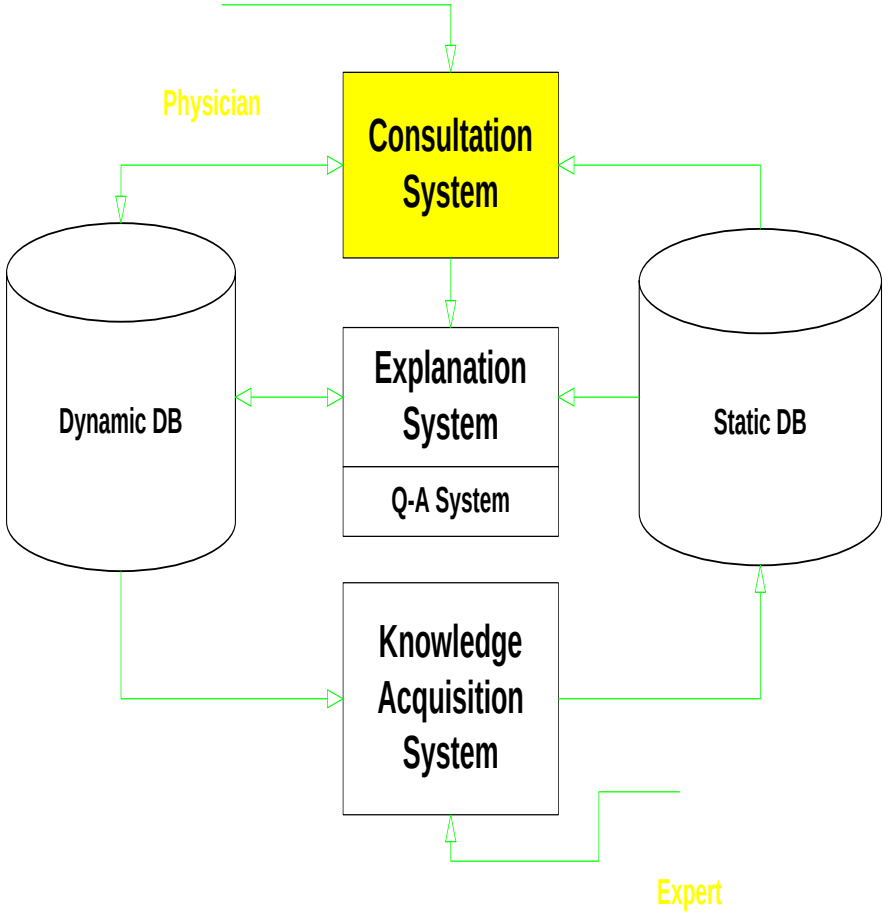
  <predicate function><object><attrib><value>

- Each rule is completely modular, all relevant context is contained in the rule with explicitly stated premises

MYCIN Architecture



Consultation System

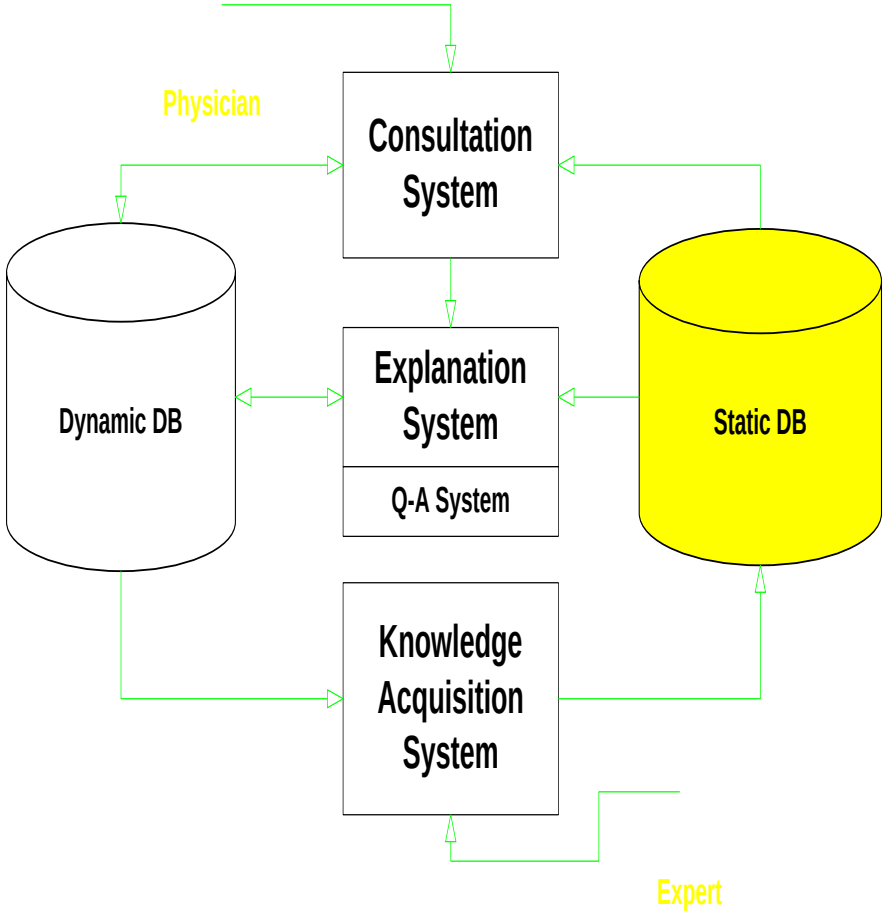


- Performs Diagnosis and Therapy Selection
- Control Structure reads Static DB (rules) and read/writes to Dynamic DB (patient, context)
- Linked to Explanations
- Terminal interface to Physician

Consultation System

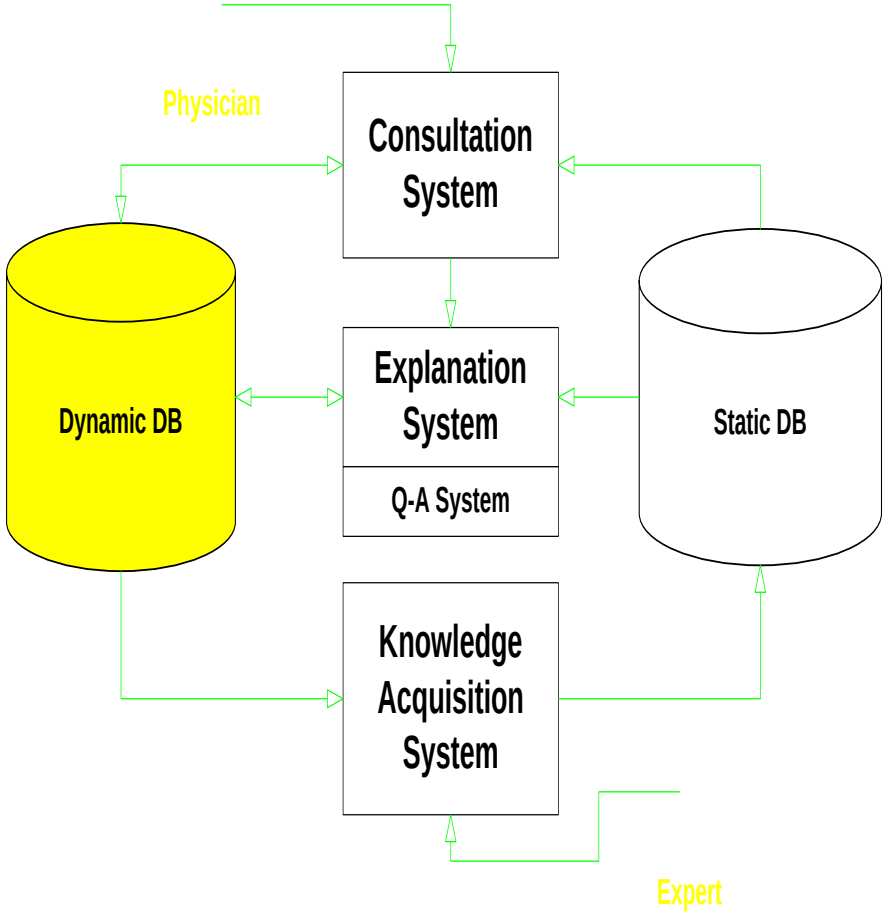
- Questions are asked when more data is needed
- Goal-directed Backward-chaining Depth-first Tree Search
- High-level Algorithm:
  1. Determine if Patient has significant infection
  2. Determine likely identity of significant organisms
  3. Decide which drugs are potentially useful
  4. Select best drug or coverage of drugs

Static Database



- Rules
- Meta-Rules
- Templates
- Rule Properties
- Context Properties
- Fed from Knowledge Acquisition System

Dynamic Database

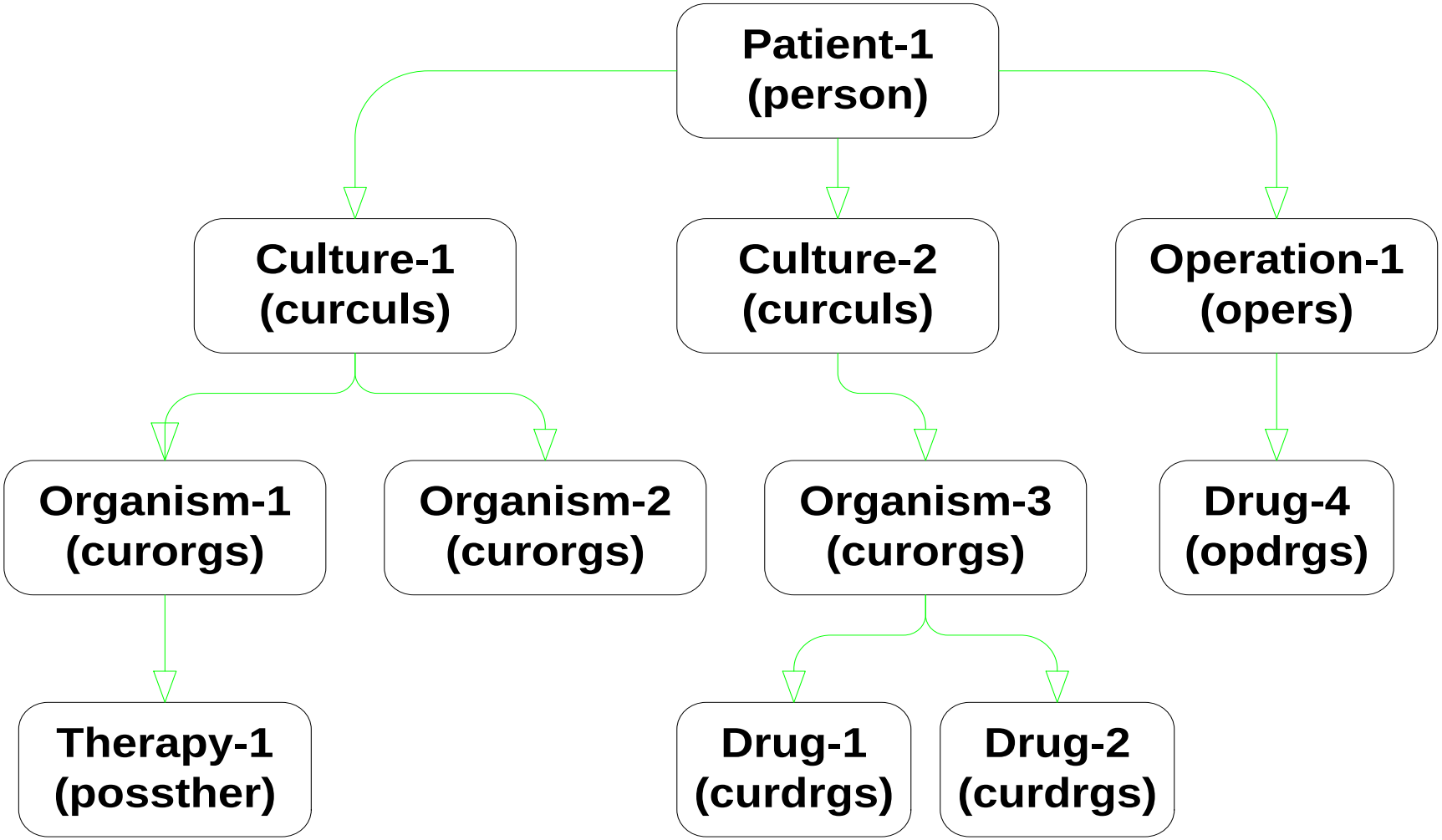


- Patient Data
- Laboratory Data
- Context Tree
- Built by Consultation System
- Used by Explanation System

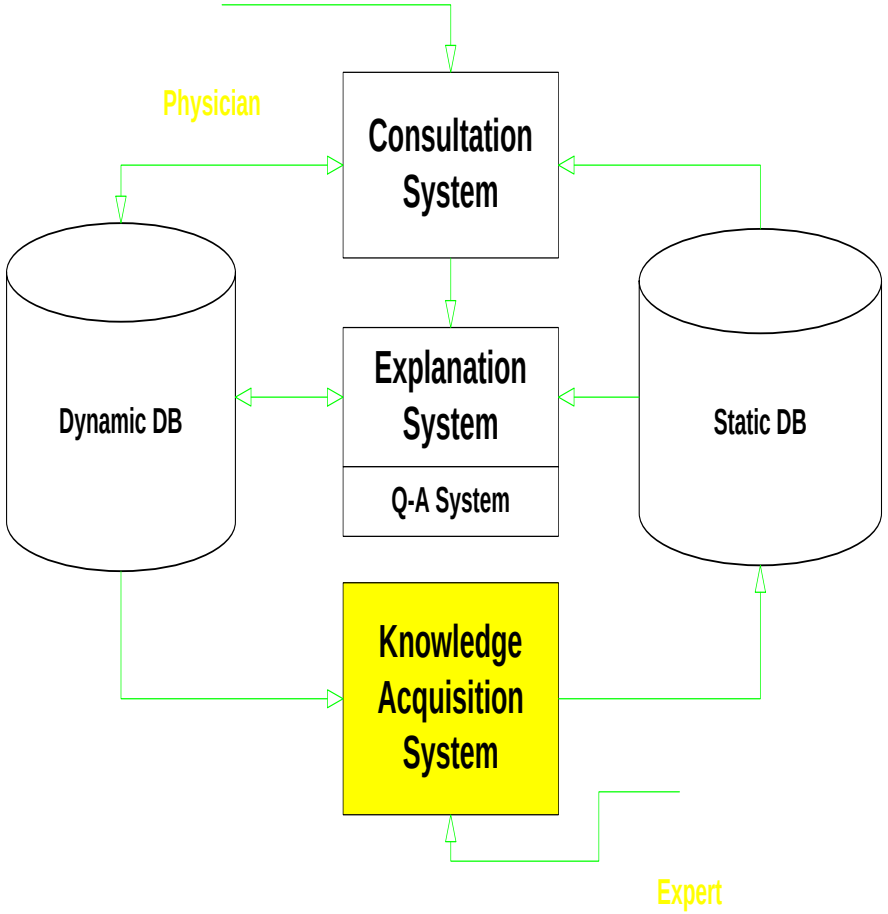
Dynamic Database

- Plan-Generate-and-Test Process
- Therapy List Creation
  - Set of specific rules recommend treatments based on the probability (not CF) of organism sensitivity
  - Probabilities based on laboratory data
  - One therapy rule for every organism

Context Tree

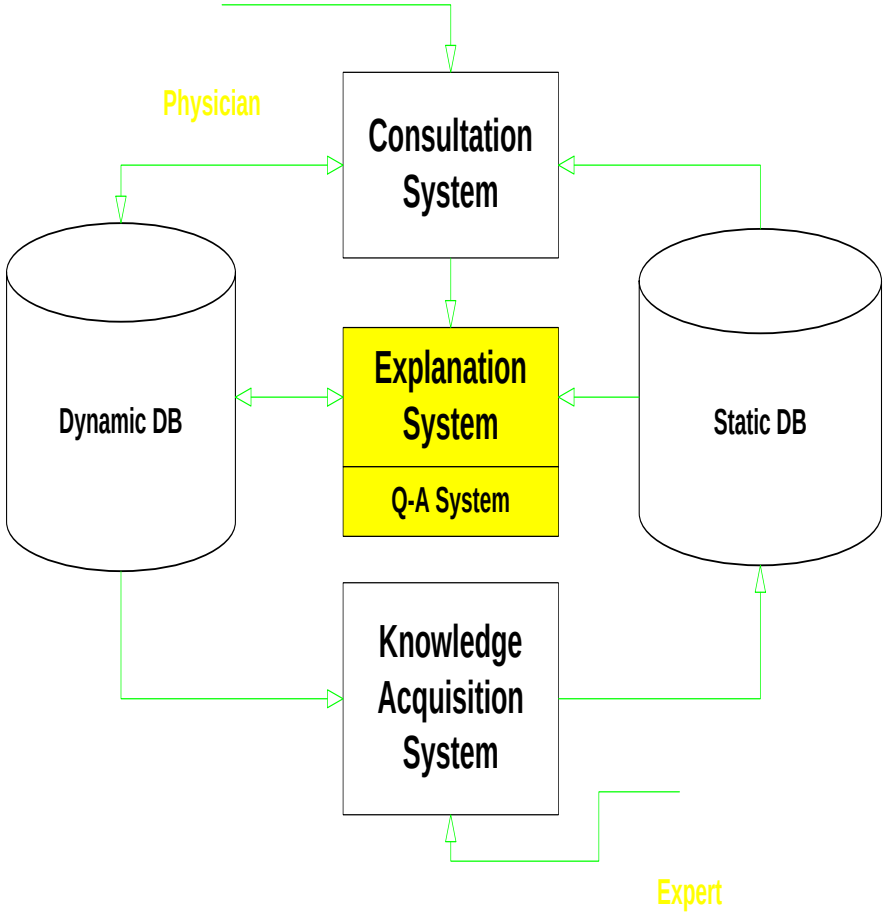


Knowledge Acquisition System



- Extends Static DB via Dialogue with Experts
- Dialogue Driven by System
- Requires minimal training for Experts

Explanation System



- Provides reasoning why a conclusion has been made, or why a question is being asked
- Q-A Module
- Reasoning Status Checker

DART (Diagnostic Assistant Reference Tool)

- An expert system for computer fault diagonosis
- DART is a joint project of stanford university and IBM
- It explores AI techniques to the diagnosis of computer faults
- DART system identifies specific system components (both hardware and software) likely to be responsible for an observed fault
- It offers a brief explanation of the major factors and evidence supporting these indictments.
- DART , was constructed using HMYCIN

DART

- It uses a device independent language for describing devices and device independent inference procedure for diagnosis
- The practical goal is the construction of an automated diagnostician capable of pinpointing the functional units responsible for observed malfunctions
- The current DART knowledge base consists of 300 EMYCIN parameters and 190 production rules and was constructed over a period of 8 months.
- During this period 5 specialists were interviewed about different aspects of the diagnostic process and the knowledge base reflects their composite expertise.
- DART has been implemented in common LISP.

XOON

- It is an Expert system that finds the hardware and software that are currently used in a particular company.
- It contains all the list of software and hardware configurations to be used and implemented accordingly.
- Its the first expert system in use.