Expert Systems

Characteristics of Expert Systems

An **expert system** is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions.

Expert systems should be able to:

1. Solve complex problems with the same (or greater) solvency as human experts
2. Perform heuristic reasoning.
3. Work with data containing errors.
4. Consider multiple hypotheses simultaneously.
5. Perform at the level of a human expert.
6. Respond in a reasonable amount of time.
7. Maintain reliability and should not crash.
8. Explain the steps of the reasoning process. It should justify how it arrived at a particular conclusion.

Components of Expert Systems

The following are the fundamental modules of an expert system:

1. Knowledge base
2. Inference engine
3. User interface
4. Explanation facility
5. Knowledge acquisition facility
6. External interface
7. Database

A **knowledge base** contains a collection of information in a given field. The knowledge base of an expert system stores both factual and heuristic knowledge.

- **Factual knowledge** – based on facts
- **Heuristic knowledge** – based on practice, evaluation, experiences, and the ability to guess

An **inference** engine is used to acquire insights from the information housed in the knowledge base. It can be divided into three (3) functional elements:

- **Control system** – determines the order of testing in the knowledge base rules
- **Rule interpreter** – defines the Boolean application rules
- **Explanation mechanism** – justifies the outcome to the user with the reasoning process

The inference engine repeatedly applies the rules to the working memory, adding new information (obtained from the rules' conclusions) until a goal state is produced or confirmed. There are two (2) strategies that an inference engine can use to reach a conclusion:

- **Forward chaining**: The inference process moves from the facts of the case to a goal (conclusion). The inference engine attempts to match each rule's condition (IF) in the knowledge base with the facts currently available in the working memory. Forward-chaining systems are used to solve the open-ended problems of a design or those that involve planning.
- **Backward chaining**: The inference engine attempts to match the assumed conclusion to the goal or sub-goal state with the conclusion (THEN) part of the rule. If such a rule is found, its premise becomes the new sub-goal. Backward chaining is best suited for applications in which the possible conclusions are limited in number and well defined.

The component of an expert system that helps its users to communicate with it is known as the **user interface**. The user interface helps in explaining how the expert system has arrived at a particular recommendation. The explanation may appear in the following forms:

- Natural language displayed on the screen
- Verbal narrations in natural language
- A listing of rule numbers displayed on the screen

The user interface should:

- Help users accomplish their goals in the shortest possible way
- Be designed to work for users' existing or desired work practices
- Adapt to the user's requirements, not the other way around
- Make efficient use of the user's input

An **explanation facility** allows the user to ask the expert system how it reached a particular conclusion and why a specific task (fact) is needed. An expert system must explain its reasoning and justify its advice, analysis, or conclusion.

Knowledge acquisition includes eliciting, collecting, analyzing, modeling, and validating knowledge for knowledge engineering and management projects. It is considered the major bottleneck in expert system development.

An **external interface** allows an expert system to work with external files using programs written in conventional programming languages like C and C++. It provides the communication link between the expert system and the external environment.

A **database** is a collection of organized information that can easily be accessed, managed, and updated. It is used by the inference engine to hold data while it is working on a problem. It holds the data about the current tasks, which include the following:

- The user's answers to questions
- Any data from outside sources
- Any intermediate results of the reasoning
- Any conclusions reached so far

Expert System Architectures

Expert systems fall into two (2) types of architectures: rule-based system architecture and non-production system architecture.

The **rule-based system architecture** (or production systems) is the most common form of architecture used in expert and other knowledge-based systems. This type of system uses knowledge in the form of production rules (if-then rules). The rule has a conditional part on the left-hand side and a conclusion or action part on the right-hand side.

Each rule represents a small chunk of knowledge related to a given domain of expertise. When the known facts support the conditions on the rule's left side, the conclusion or action part of the rule is then accepted as known. A number of related rules collectively may correspond to a chain of inferences, which lead from some initially known facts to some useful conclusions. Inference in a production system is achieved by chaining through the rules recursively, either in a forward or backward direction, until a conclusion is reached, or a failure occurs.

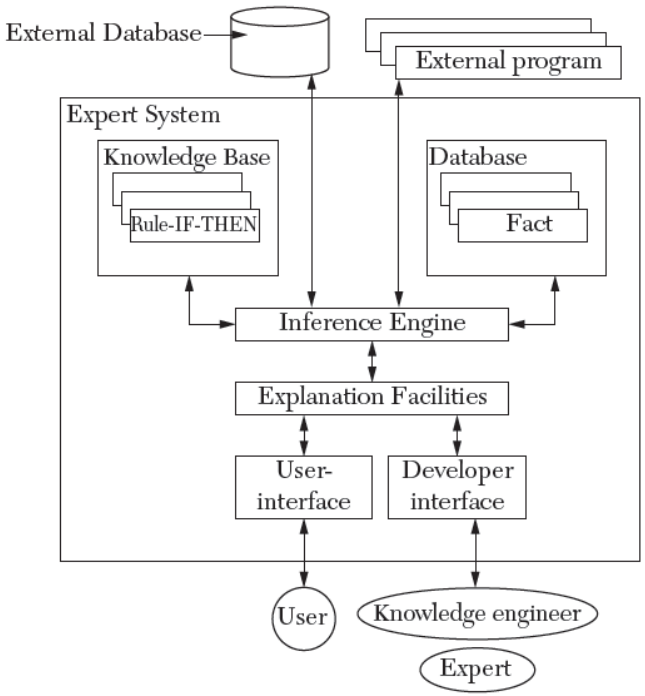

Figure 1. Structure of a rule-based expert system.

Instead of rules, **non-production systems** use more structured representation schemes, such as the semantic/associative network, frames, tree structure (decision trees), and neural networks.

When knowledge can be structured in a top-to-bottom manner, it may be stored in the form of a decision tree. For example, the identification of objects can correspond to an object's attribute, and the terminal nodes can correspond to the identities of objects. A decision tree takes input from an object given by a set of properties and outputs a Boolean value (yes/no decision). Each internal node in the tree corresponds to a test of one property. Branches are labeled with possible values of the test. For example, let's say you are waiting for a table at a restaurant. A decision tree can be used to determine whether to wait or not. The following are the possible attributes and an illustration of the decision tree:

- **Alternative**: Alternative restaurant nearby
- **Bar**: Bar area to wait
- **Fri/Sat**: Is it a Friday or Saturday
- **Hungry**: Whether you are hungry
- **Customers**: How many people are in the restaurant (none, some, or full)
- **Raining**: Raining outside
- **Reservation**: Whether you made a reservation
- **Wait Estimate**: Estimated wait time (<10, 10–30, 30–60, or >60)

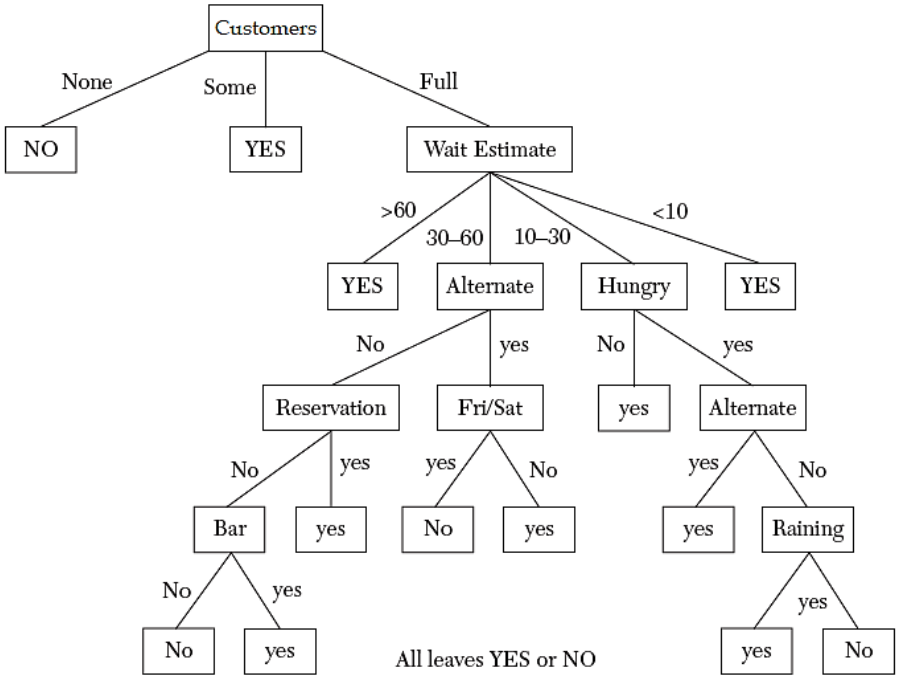

Figure 2. A sample decision tree.

The **blackboard system architecture** refers to a special type of knowledge-based system that uses different knowledge sources that communicate through a common information field. The blackboard system architecture consists of three (3) functional components: the blackboard, knowledge sources, and control unit.

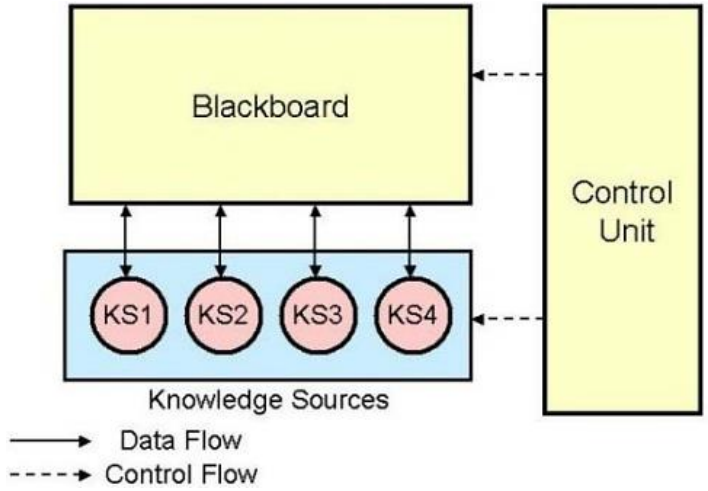

Figure 3. A blackboard system architecture.

Knowledge sources are independent modules that contain the knowledge needed for problem solving. The **blackboard** is used as a global database for sharing different information as input data, partial solutions, alternatives and final solutions. The **control unit** determines which knowledge sources to execute for an optimal problem solution.

Neural networks are computing systems modeled on the human brain's mesh-like network of interconnected processing elements called **neurons**. A neural network is an array of interconnected processing elements, each of which can accept inputs, process them, and produce a single output with the objective of imitating the human brain. Knowledge is represented in neural networks by the pattern of connections formed during the processing of elements and by adjusting the weights of these connections.

References:

Gupta, N. & Mangla, R. (2020). *Artificial intelligence basics: A self-teaching introduction*. Mercury Learning and Information LLC.

Russell, S. & Norvig, P. (2022). *Artificial intelligence: A modern approach* (4th ed.). Pearson Education Limited.