Expert Systems

S. Sheerazuddin

March 17, 2018

1 Introduction

In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if then rules rather than through conventional procedural code.

The first expert systems were created in the 1970s and then proliferated in the 1980s. Expert systems were among the first truly successful forms of artificial intelligence (AI) software. The languages that were used to develop early expert systems were LISP and PROLOG.

Expert systems were introduced by the Stanford Heuristic Programming Project led by Edward Feigenbaum, who is sometimes termed the "father of expert systems". Other key early contributors were Bruce Buchanan and Randall Davis.

The Stanford researchers tried to identify domains where expertise was highly valued and complex, such as diagnosing infectious diseases and identifying unknown organic molecules. **Dendral** was the first expert system built by Stanford Heuristic Programming Project. It was built to identify and analyse organic molecules. **Mycin** was a medical diagnosis expert system build by the same research group which used certainty factor model.

The first successful commercial expert system was **R1**, developed at Digital Equipment Corporation in the early 1980's. It was later renamed as XCON. R1 had rules for designing configuration of computer components for DEC consumers.

2 Architecture

An expert system is divided into two subsystems: the **inference engine** and the **knowledge base**.

The knowledge base represents facts about the world and if-then-else rules to reason with them. In early expert systems such as Mycin and Dendral, these facts were represented mainly as flat assertions about variables. In later expert systems developed with commercial shells, the knowledge base took on more structure and used concepts from object-oriented programming. The world was represented as classes, subclasses, and instances and assertions were replaced by values of object instances. The rules worked by querying and asserting values of the objects.

The inference engine is an automated reasoning system that evaluates the current state of the knowledge-base, applies relevant rules, and then asserts new knowledge into the knowledge base. The inference engine may also include abilities for explanation, so that it can explain to a user the chain of reasoning used to arrive at a particular conclusion by tracing back over the firing of rules that resulted in the assertion. The inference engine uses either **forward chaining** or **backward chaining** to do the reasoning. It also has some **working memory** at its disposal.

Apart from these modules, there is a **user interface** to obtain facts and queries from the user and supply advise and/or explanation to him/her whenever asked or needed.

3 Discussion

An expert system should have the following characteristics:

- **High Performance:** Should perform at the level of a human expert.
- **Low response time:** Should have the ability to respond in a reasonable amount of time, especially in the case of real time systems.
- **Reliable:** Must be reliable and not crash.
- **Understandable:** Should not be a black box, but explain the steps in its reasoning process.

Expert Systems have many advantages:

1. **Availability:** Expert systems are available easily due to mass production software.

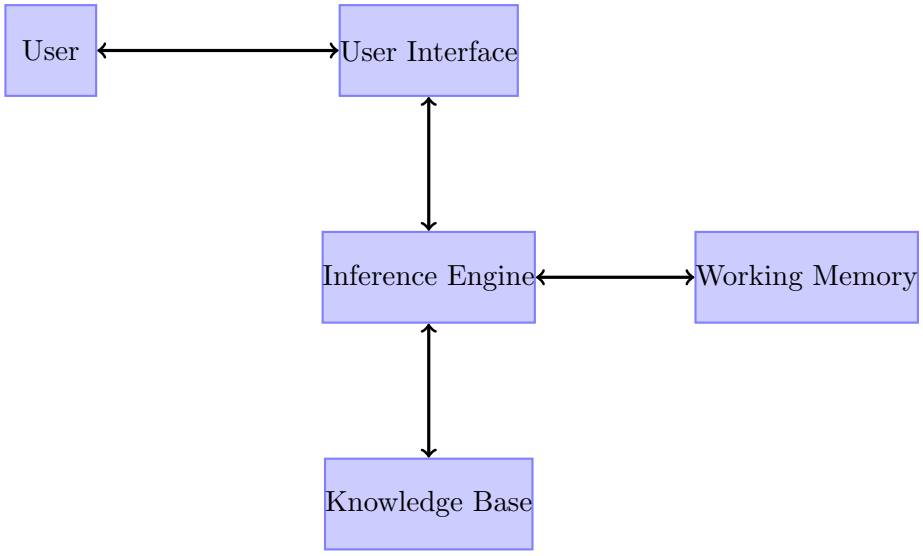

Figure 1: Architecture of Expert Systems

2. **Cheaper:** The cost of providing expertise is not expensive.
3. **Reduced danger:** They can be used in any risky environments where humans cannot work with.
4. **Permanence:** The knowledge will last long indefinitely.
5. **Multiple expertise:** It can be designed to have knowledge of many experts.
6. **Explanation:** They are capable of explaining in detail the reasoning that led to a conclusion.
7. **Fast response:** They can respond at great speed due to the inherent advantages of computers over humans.
8. Unemotional and response at all times: Unlike humans, they do not get tense, fatigue or panic and work steadily during emergency situations.

There are two major disadvantages with expert systems:

1. **knowledge acquisition**: Obtaining the time of domain experts for any software application is always difficult, but for expert systems it

was especially difficult because the experts were by definition highly valued and in constant demand by the organization.

As a result, a great deal of research in the later years of expert systems was focused on tools for knowledge acquisition, to help automate the process of designing, debugging, and maintaining rules defined by experts.

2. **performance and integration**: The early expert systems were built using tools such as LISP, which executed interpreted (rather than compiled) code. Interpreting provided an extremely powerful development environment but with the drawback that it was virtually impossible to match the efficiency of the fastest compiled languages, such as C. System and database integration were difficult for early expert systems because the tools were mostly in languages and platforms that were neither familiar to nor welcome in most corporate IT environments. As a result, much effort in the later stages of expert system tool development was focused on integrating with legacy environments such as COBOL and large database systems, and on porting to more standard platforms.

The first expert systems were based on imitation of thought steps of human experts. Later expert systems were **normative** in nature, i.e., they were expected to act rationally according to laws of decision theory.