Chapter 1:
Introduction to Expert Systems


What is an expert system?

“An expert system is a computer system that emulates, or acts in all respects, with the decision-making capabilities of a human expert.”

Professor Edward Feigenbaum
Stanford University
Expert system technology may include:

- Special expert system languages – CLIPS
- Programs
- Hardware designed to facilitate the implementation of those systems
Expert System Main Components

- Knowledge base – obtainable from books, magazines, knowledgeable persons, etc.
- Inference engine – draws conclusions from the knowledge base

Figure 1.2 Basic Functions of Expert Systems
Problem Domain vs. Knowledge Domain

- An expert’s knowledge is specific to one problem domain – medicine, finance, science, engineering, etc.
- The expert’s knowledge about solving specific problems is called the knowledge domain.
- The problem domain is always a superset of the knowledge domain.

Figure 1.3 Problem and Knowledge Domain Relationship
Advantages of Expert Systems

- Increased availability
- Reduced cost
- Reduced danger
- Performance
- Multiple expertise
- Increased reliability

Advantages Continued

- Explanation
- Fast response
- Steady, unemotional, and complete responses at all times
- Intelligent tutor
- Intelligent database
Representing the Knowledge

The knowledge of an expert system can be represented in a number of ways, including IF-THEN rules:

IF you are hungry THEN eat

Knowledge Engineering

The process of building an expert system:

1. The knowledge engineer establishes a dialog with the human expert to elicit knowledge.
2. The knowledge engineer codes the knowledge explicitly in the knowledge base.
3. The expert evaluates the expert system and gives a critique to the knowledge engineer.
Development of an Expert System

The Role of AI

- An algorithm is an ideal solution guaranteed to yield a solution in a finite amount of time.
- When an algorithm is not available or is insufficient, we rely on artificial intelligence (AI).
- Expert system relies on inference – we accept a “reasonable solution.”
Uncertainty

- Both human experts and expert systems must be able to deal with uncertainty.
- It is easier to program expert systems with shallow knowledge than with deep knowledge.
- Shallow knowledge – based on empirical and heuristic knowledge.
- Deep knowledge – based on basic structure, function, and behavior of objects.

Limitations of Expert Systems

- Typical expert systems cannot generalize through analogy to reason about new situations in the way people can.
- A knowledge acquisition bottleneck results from the time-consuming and labor intensive task of building an expert system.
Early Expert Systems

- DENDRAL – used in chemical mass spectroscopy to identify chemical constituents
- MYCIN – medical diagnosis of illness
- DIPMETER – geological data analysis for oil
- PROSPECTOR – geological data analysis for minerals
- XCON/R1 – configuring computer systems

Table 1.3 Broad Classes of Expert Systems

<table>
<thead>
<tr>
<th>Class</th>
<th>General Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Assemble proper components of a system in the proper way.</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Infer underlying problems based on observed evidence.</td>
</tr>
<tr>
<td>Instruction</td>
<td>Intelligent teaching so that a student can ask why, how, and what if questions just as if a human were teaching.</td>
</tr>
<tr>
<td>Interpretation</td>
<td>Explain observed data.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Compares observed data to expected data to judge performance.</td>
</tr>
<tr>
<td>Planning</td>
<td>Devise actions to yield a desired outcome.</td>
</tr>
<tr>
<td>Prognosis</td>
<td>Predict the outcome of a given situation.</td>
</tr>
<tr>
<td>Remedy</td>
<td>Prescribe treatment for a problem.</td>
</tr>
<tr>
<td>Control</td>
<td>Regulate a process. May require interpretation, diagnosis, monitoring, planning, prognosis, and remedies.</td>
</tr>
</tbody>
</table>
Problems with Algorithmic Solutions

- Conventional computer programs generally solve problems having algorithmic solutions.
- Algorithmic languages include C, Java, and C#.
- Classic AI languages include LISP and PROLOG.

Considerations for Building Expert Systems

- Can the problem be solved effectively by conventional programming?
- Is there a need and a desire for an expert system?
- Is there at least one human expert who is willing to cooperate?
- Can the expert explain the knowledge to the knowledge engineer can understand it.
- Is the problem-solving knowledge mainly heuristic and uncertain?
Languages, Shells, and Tools

- Expert system languages are post-third generation.
- Procedural languages (e.g., C) focus on techniques to represent data.
- More modern languages (e.g., Java) focus on data abstraction.
- Expert system languages (e.g., CLIPS) focus on ways to represent knowledge.

Elements of an Expert System

- User interface – mechanism by which user and system communicate.
- Exploration facility – explains reasoning of expert system to user.
- Working memory – global database of facts used by rules.
- Inference engine – makes inferences deciding which rules are satisfied and prioritizing.
Elements Continued

- Agenda – a prioritized list of rules created by the inference engine, whose patterns are satisfied by facts or objects in working memory.
- Knowledge acquisition facility – automatic way for the user to enter knowledge in the system bypassing the explicit coding by knowledge engineer.

Production Rules

- Knowledge base is also called production memory.
- Production rules can be expressed in IF-THEN pseudocode format.
- In rule-based systems, the inference engine determines which rule antecedents are satisfied by the facts.
General Methods of Inferencing

- Forward chaining – reasoning from facts to the conclusions resulting from those facts – best for prognosis, monitoring, and control.

- Backward chaining – reasoning in reverse from a hypothesis, a potential conclusion to be proved to the facts that support the hypothesis – best for diagnosis problems.
Production Systems

- Rule-based expert systems – most popular type today.
- Knowledge is represented as multiple rules that specify what should/not be concluded from different situations.
- Forward chaining – start w/facts and use rules to draw conclusions/take actions.
- Backward chaining – start w/hypothesis and look for rules that allow hypothesis to be proven true.

Forward/Backward Chaining

- Forward chaining – primarily data-driven.
- Backward chaining – primarily goal driven.
Post Production System

- Basic idea – any mathematical / logical system is simply a set of rules specifying how to change one string of symbols into another string of symbols.

- Basic limitation – lack of control mechanism to guide the application of the rules.

Markov Algorithm

- An ordered group of productions applied in order or priority to an input string.

- If the highest priority rule is not applicable, we apply the next, and so on.

- An efficient algorithm for systems with many rules.
Rete Algorithm

• Functions like a net – holding a lot of information.
• Much faster response times and rule firings can occur compared to a large group of IF-THEN rules which would have to be checked one-by-one in conventional program.
• Takes advantage of temporal redundancy and structural similarity.
• Drawback is high memory space requirements.

Procedural Paradigms

• Algorithm – method of solving a problem in a finite number of steps.
• Procedural programs are also called sequential programs.
• The programmer specifies exactly how a problem solution must be coded.
Imperative Programming

- Focuses on the concept of modifiable store – variables and assignments.
- During execution, program makes transition from the initial state to the final state by passing through series of intermediate states.
- Provide for top-down-design.
- Not efficient for directly implementing expert systems.
Nonprocedural Paradigms

- Do not depend on the programmer giving exact details how the program is to be solved.
- Declarative programming – goal is separated from the method to achieve it.
- Object-oriented programming – partly imperative and partly declarative – uses objects and methods that act on those objects.
- Inheritance – (OOP) subclasses derived from parent classes.

Figure 1.9 Nonprocedural Languages
What are Expert Systems?

Can be considered declarative languages:

- Programmer does not specify how to achieve a goal at the algorithm level.

- Induction-based programming – the program learns by generalizing from a sample.

Summary

- During the 20th Century various definitions of AI were proposed.
- In the 1960s, a special type of AI called expert systems dealt with complex problems in a narrow domain, e.g., medical disease diagnosis.
- Today, expert systems are used in a variety of fields.
- Expert systems solve problems for which there are no known algorithms.
Summary Continued

• Expert systems are knowledge-based – effective for solving real-world problems.
• Expert systems are not suited for all applications.
• Future advances in expert systems will hinge on the new quantum computers and those with massive computational abilities in conjunction with computers on the Internet.