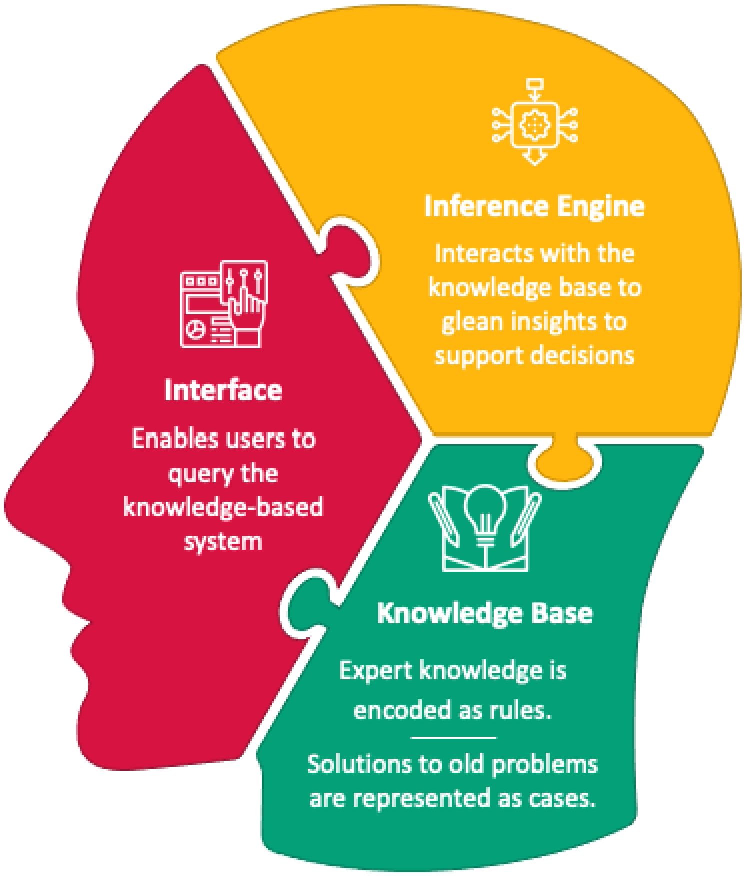


Expert Systems

Expert Systems – Introduction

Expert Systems are computer programs that exhibit intelligent behavior. They are concerned with the concepts and methods of symbolic inference, or reasoning, by a computer, and how the knowledge used to make those inferences will be represented.

Achieving expert-level competence in solving problems in task areas by bringing to bear a body of knowledge about specific tasks is called *knowledge-based* or *expert system*. The term expert system is reserved for programs whose knowledge base contains the knowledge used by human experts. Expert systems and knowledge-based systems are used synonymously. The area of human intellectual endeavor to be captured in an expert system is called the *task domain*. *Task* refers to some goal-oriented, problem-solving activity. *Domain* refers to the area within which the task is being performed. Typical tasks are diagnosis, planning, scheduling, configuration and design.

Expert Systems – Introduction

Building an expert system is known as *knowledge engineering* and its practitioners are called *knowledge engineers*. The knowledge engineer must make sure that the computer has all the knowledge needed to solve a problem. The knowledge engineer must choose one or more forms in which to represent the required knowledge as symbol patterns in the memory of the computer -- that is, he (or she) must choose a *knowledge representation*. He must also ensure that the computer can use the knowledge efficiently by selecting from a handful of *reasoning methods*.

Expert Systems – Building Blocks of Expert Systems

Every expert system consists of two principal parts: the knowledge base; and the reasoning, or inference, engine.

The *knowledge base* of expert systems contains both factual and heuristic knowledge. *Factual knowledge* is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field. *Heuristic knowledge* is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the "art of good guessing."

Expert Systems – Building Blocks of Expert Systems

Knowledge representation formalizes and organizes the knowledge. One widely used representation is the *production rule*, or simply *rule*. A rule consists of an IF part and a THEN part (also called a *condition* and an *action*). The IF part lists a set of conditions in some logical combination. The piece of knowledge represented by the production rule is relevant to the line of reasoning being developed if the IF part of the rule is satisfied; consequently, the THEN part can be concluded, or its problem-solving action taken. Expert systems whose knowledge is represented in rule form are called *rule-based systems*.

Expert Systems – Building Blocks of Expert Systems

The *problem-solving model,* or *paradigm*, organizes and controls the steps taken to solve the problem. One common but powerful paradigm involves chaining of IF-THEN rules to form a line of reasoning. If the chaining starts from a set of conditions and moves toward some conclusion, the method is called *forward chaining*. If the conclusion is known (for example, a goal to be achieved) but the path to that conclusion is not known, then reasoning backwards is called for, and the method is *backward chaining*. These problem-solving methods are built into program modules called *inference engines* or *inference procedures* that manipulate and use knowledge in the knowledge base to form a line of reasoning.

Expert Systems – Forward and Backward Chaining

Forward Chaining

- When using an inference engine, forward chaining is also known as forward deduction or forward reasoning. Forward chaining is a type of reasoning in which atomic sentences in a knowledge base are used to extract more data in the forward direction using inference rules (Modus Ponens).
- The Forward-chaining algorithm starts with known facts, then triggers all rules with satisfied premises and adds their conclusion to the known facts. This
- process is repeated until the issue is resolved.

Forward Chaining Characteristics

- As it moves from bottom to top, it is a down-up approach.
- It is the process of reaching a conclusion based on known facts or data, beginning with the initial state and progressing to the goal state.
- The forward-chaining approach is also known as data-driven because we achieve our goal by utilizing available data.
- The forward-chaining approach is commonly used in expert systems such as CLIPS, business rule systems, and production rule systems.

Backward Chaining

- Also known as backward reasoning, is an inference engine reasoning technique that begins with a hypothetical goal. Backtracking is used to find the most optimal way to resolve a conflict or reach a goal state, where the search begins at the conclusion and goes back to understand the conditions that led to the conclusion.
- This inference engine reasoning technique is used by systems to find the conditions and rules that resulted in a logical result or conclusion.

Backward Chaining Characteristics

- It is referred to as a top-down approach.
- Backward chaining is based on the rule of modus ponens inference.
- Backward chaining divides the goal into sub-goals or sub-goals to demonstrate the truth of the facts.
- A goal-driven approach is used because a list of goals determines which rules are selected and used.
- Backward-chaining algorithms are used in game theory, automated theorem proving tools, inference engines, proof assistants, and a variety of AI applications.
- For proof, the backward-chaining method mostly used a depth-first search strategy.

Expert Systems – Building Blocks of Expert Systems

The *knowledge base* an expert uses is what he learned at school, from colleagues, and from years of experience. Presumably the more experience he has, the larger his store of knowledge. Knowledge allows him to interpret the information in his databases to advantage in diagnosis, design, and analysis.

Though an expert system consists primarily of a knowledge base and an inference engine, a couple of other features are worth mentioning: reasoning with uncertainty, and explanation of the line of reasoning.

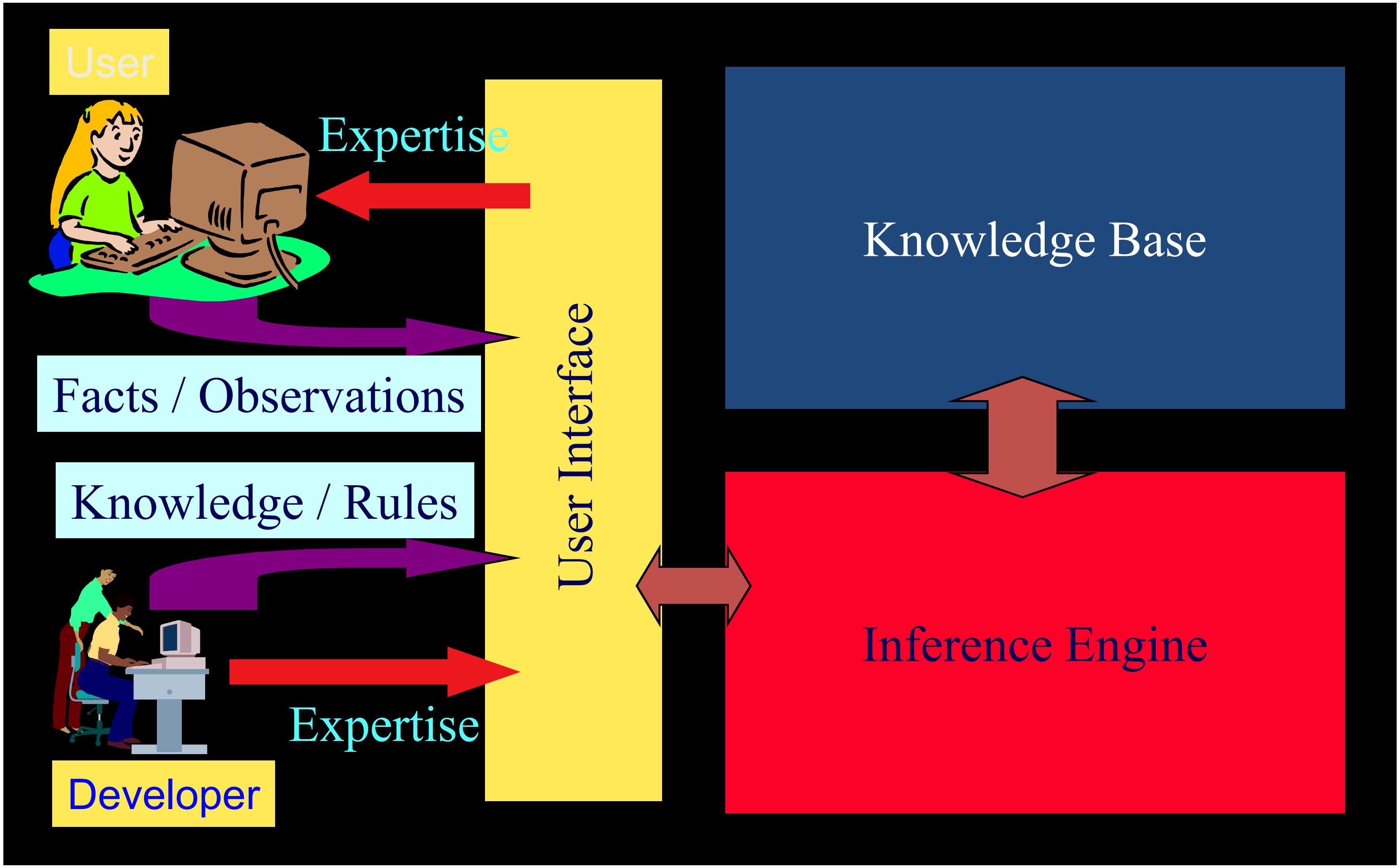
Expert Systems – Building Blocks of Expert Systems

Knowledge is almost always incomplete and uncertain. Thus a rule may have associated with it a *confidence factor* or a weight. The set of methods for using uncertain knowledge in combination with uncertain data in reasoning is called *reasoning with uncertainty*. An subclass of methods for reasoning with uncertainty is called "fuzzy logic," and the systems are known as "fuzzy systems."

Because an expert system uses uncertain or heuristic knowledge (as humans do) its credibility is often in question (as with humans). When an answer to a problem is questionable, we tend to want to know the rationale. If the rationale seems plausible, we tend to believe the answer. So it is with expert systems. Most expert systems have the ability to answer questions of the form: "Why is the answer X?" Explanations can be generated by tracing the line of reasoning used by the inference engine.

Expert Systems – Building Blocks of Expert Systems

The most important ingredient in any expert system is knowledge. The power of expert systems resides in the specific, high-quality knowledge they contain about task domains. Researchers will continue to explore and add to the current repertoire of knowledge representation and reasoning methods. But in knowledge resides the power. Because of the importance of knowledge in expert systems and because the current knowledge acquisition method is slow and tedious, much of the future of expert systems depends on breaking the knowledge acquisition bottleneck and in codifying and representing a large knowledge infrastructure.


User
Expertise
Facts / Observations
Knowledge / Rules
Expertise
Developer
User Interface
Knowledge Base
Inference Engine

Expert Systems – Components of Expert Systems

knowledge base

- contains essential information about the problem domain
- often represented as facts and rules

inference engine

- mechanism to derive new knowledge from the knowledge base and the information provided by the user
- often based on the use of rules

user interface

- interaction with end users
- development and maintenance of the knowledge base

Expert Systems – Concepts & Characteristics of Expert Systems

knowledge acquisition

- transfer of knowledge from humans to computers
- sometimes knowledge can be acquired directly from the environment
  - machine learning, neural networks

knowledge representation

- suitable for storing and processing knowledge in computers

inference

- mechanism that allows the generation of new conclusions from existing knowledge in a computer

explanation

- illustrates to the user how and why a particular solution was generated

Expert Systems – Rules and Humans

rules can be used to formulate a theory of human information processing (Newell & Simon)

- rules are stored in long-term memory
- temporary knowledge is kept in short-term memory
- (external) sensory input triggers the activation of rules
- activated rules may trigger further activation (internal input; "thinking")
- a cognitive processor combines evidence from currently active rules

this model is the basis for the design of many rule-based systems (*production systems*)

Expert Systems – Early Expert Systems Success Stories

DENDRAL (Feigenbaum, Lederberg, and Buchanan, 1965)

- deduce the likely molecular structure of organic chemical compounds from known chemical analyses and mass spectrometry data

MYCIN (Buchanan and Shortliffe, 1972-1980)

- diagnosis of infectious blood diseases and recommendation for use of antibiotics
- “empty” MYCIN = EMYCIN = XPS shell

PROSPECTOR

- analysis of geological data for minerals
- discovered a mineral deposit worth $100 million

XCON/R1 (McDermott, 1978)

- configuration of DEC VAX computer systems
- 2500 rules; processed 80,000 orders by 1986; saved DEC $25M a year

Expert Systems – Keys to Expert Systems Success

convincing ideas

- rules, cognitive models

practical applications

- medicine, computer technology, …

separation of knowledge and inference

- expert system *shell*
  - allows the re-use of the "machinery" for different domains

concentration on domain knowledge

- general reasoning is too complicated

Expert Systems – When not to use an Expert System

Expert systems are not suitable for all types of domains and tasks

They are not useful or preferable, when …

- efficient conventional algorithms are known
- the main challenge is computation, not knowledge
- knowledge cannot be captured efficiently or used effectively
- users are reluctant to apply an expert system, e.g. due to criticality of task, high risk or high security demands

Expert Systems – Architecture of Expert Systems

Knowledge-Base / Rule-Base

store expert knowledge as condition-action-rules (aka: if-then- or premise-consequence-rules)

Working Memory

stores initial facts and generated facts derived by inference engine; maybe with additional parameters like the “degree of trust” into the truth of a fact ≅ certainty factor

Expert Systems – Architecture of Expert Systems

Inference Engine

- matches condition-part of rules against facts stored in Working Memory (pattern matching);
- rules with satisfied condition are active rules and are placed on the agenda;
- among the active rules on the agenda, one is selected (see conflict resolution, priorities of rules) as next rule for
- execution ("firing") – consequence of rule is added as new fact(s) to Working Memory

Expert Systems – Architecture of Expert Systems

Inference Engine + additional components might be necessary for other functions, like

- calculation of certainty values,
- determining priorities of rules,
- conflict resolution mechanisms,
- a truth maintenance system (TMS) if reasoning with defaults and beliefs is requested

Expert Systems – Architecture of Expert Systems

Explanation Facility

provides justification of solution to user (reasoning chain)

Knowledge Acquisition Facility

helps to integrate new knowledge; also automated knowledge acquisition

User Interface

allows user to interact with the XPS - insert facts, query the system, solution presentation

Expert Systems – Rule-Based Expert Systems

knowledge is encoded as **IF … THEN** rules

- Condition-action pairs

the inference engine determines which rule antecedents (condition-part) are satisfied

- the left-hand condition-part must “match” facts in the working memory

matching rules are “activated”, i.e. placed on the agenda

rules on the agenda can be executed (“fired”)

- an activated rule may generate new facts and/or cause actions through its right-hand side (action-part)
- the activation of a rule may thus cause the activation of other rules through added facts based on the right-hand side of the fired rule

Rules can represent relations, recommendations, directives, strategies and heuristics:

Relation

IF the ‘fuel tank’ is empty

THEN the car is dead

Recommendation

IF the season is autumn

AND the sky is cloudy

AND the forecast is drizzle

THEN the advice is ‘take an umbrella’

Directive

IF the car is dead AND the ‘fuel tank’ is empty

THEN the action is ‘refuel the car’

Strategy

IF the car is dead

THEN the action is ‘check the fuel tank’;

step1 is complete

IF step1 is complete

AND the ‘fuel tank’ is full

THEN the action is ‘check the battery’;

step2 is complete

Heuristic

IF the spill is liquid

AND the ‘spill pH’ < 6

AND the ‘spill smell’ is vinegar

THEN the ‘spill material’ is ‘acetic acid’

Expert Systems – MYCIN Sample Rule

Human-Readable Format

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobiocity of the organism is gram anaerobic
THEN there is strong evidence (0.8)
that the class of the organism is enterobacteriaceae

MYCIN Format

```
IF (AND (SAME CNTEXT GRAM GRAMNEG)
        (SAME CNTEXT MORPH ROD)
        (SAME CNTEXT AIR AEROBIC)
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
     TALLY .8)
```

Expert Systems – Advantages

economical
- lower cost per user

availability
- accessible anytime, almost anywhere

response time
- often faster than human experts

reliability
- can be greater than that of human experts
- no distraction, fatigue, emotional involvement, …

explanation
- reasoning steps that lead to a particular conclusion

intellectual property
- can’t walk out of the door

Expert Systems – Problems

limited knowledge

- “shallow” knowledge
  - no “deep” understanding of the concepts and their relationships
- no “common-sense” knowledge
- no knowledge from possibly relevant related domains
- “closed world”
  - the XPS knows only what it has been explicitly “told”
  - it doesn’t know what it doesn’t know

mechanical reasoning

- may not have or select the most appropriate method for a particular problem
- some “easy” problems are computationally very expensive

lack of trust

- users may not want to leave critical decisions to machines

Expert Systems – Summary

expert systems or knowledge based systems are used to represent and process knowledge in a format that is suitable for computers but still understandable by humans

- If-Then rules are a popular format

the main components of an expert system are

- knowledge base
- inference engine

Expert Systems can be cheaper, faster, more accessible, and more reliable than humans

Expert Systems have limited knowledge (especially “common-sense”), can be difficult and expensive to develop, and users may not trust them for critical decisions