Expert Systems

AN INTRODUCTION

Definition

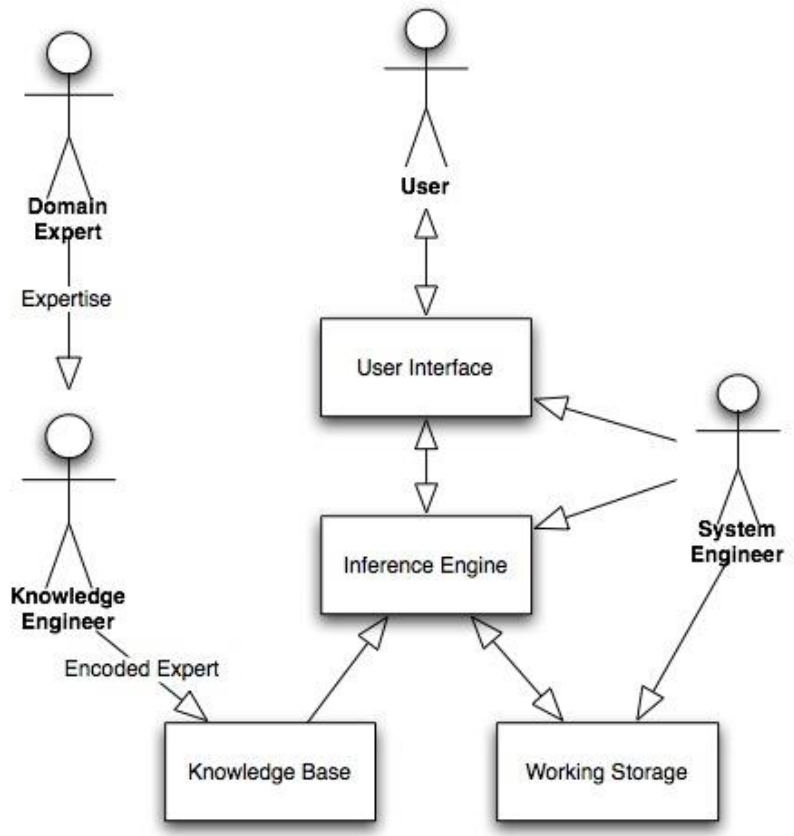
- Knowledge-based expert systems or simply expert systems
- **An expert system** is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules rather than through conventional procedural code. [Wikipedia]
- Use human knowledge to solve problems that normally would require human intelligence
- Contains some non-algorithmic expertise
- Represent the expertise knowledge as data or rules within the computer. Can be called on demand.

Features of ES

- Developed via specialized software tools called shells
- Shells come equipped with an inference mechanism
  - Backward chaining
  - Forward chaining
  - Both
- May or may not have learning components
- They are tested by being placed in the same real world problem solving situation.
- Key idea: problem solved by applying specific knowledge rather than specific technique.

Expert System Main Components

- **Knowledge base –** obtainable from books, magazines, knowledgeable persons, etc, a declarative representation of the expertise, often in IF THEN rules
- **Working storage** - the data which is specific to a problem being solved
- **Inference engine –** draws conclusions from the knowledge base and problem-specific data in working storage. The code at the core of the system.
- **User interface** - the code that controls the dialog between the user and the system



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.

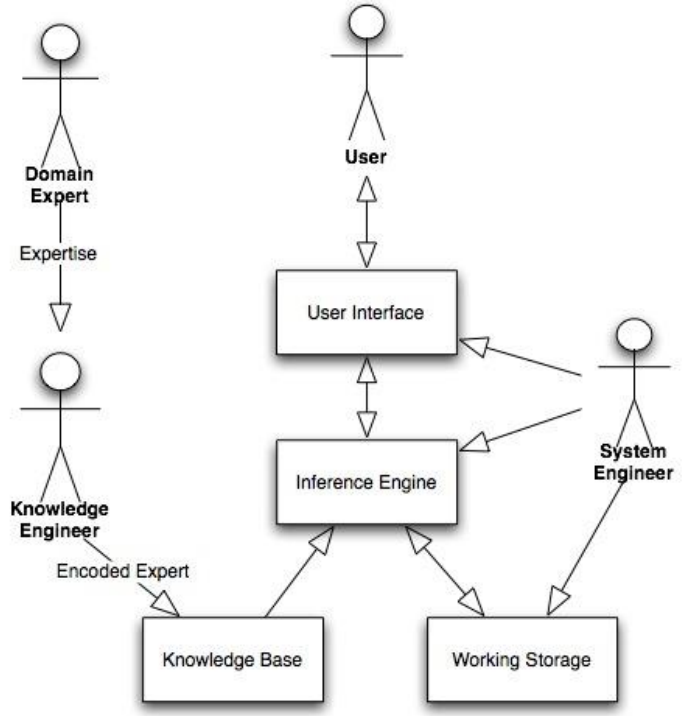
Knowledge Engineering

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.

Main actors

- **Domain expert** – currently experts solving the problems the system is intended to solve
- **Knowledge engineer** - encodes the expert's knowledge in a declarative form that can be used by the expert system
- **User** - will be consulting with the system to get advice which would have been provided by the expert
- Systems built with custom developed shells for particular applications:
  - **System engineer** - the individual who builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine



More about components

- **Shell** - a piece of software which contains:
  - The user interface
  - A format for declarative knowledge in the knowledge base
  - An inference engine
- Major advantage of a customized shell: the format of the knowledge base can be designed to facilitate the knowledge engineering process
- Knowledge engineer and the system engineer might be the same person
  - Depending on the size of the project
- One of the major bottlenecks - knowledge engineering process:
  - The coding of the expertise into the declarative rule format can be a difficult and tedious task
  - The semantic gap between the expert's representation of the knowledge and the representation in the knowledge base should be minimize

Expert System Features

- **Goal driven reasoning or backward chaining** - an inference technique which uses IF THEN rules to repetitively break a goal into smaller sub-goals which are easier to prove
- **Coping with uncertainty** - the ability of the system to reason with rules and data which are not precisely known
- **Data driven reasoning or forward chaining** - an inference technique which uses IF THEN rules to deduce a problem solution from initial data
- **Data representation** - the way in which the problem specific data in the system is stored and accessed
- **User interface** - that portion of the code which creates an easy to use system
- **Explanations** - the ability of the system to explain the reasoning process that it used to reach a recommendation.

Short example & some theory

- http://expertise2go.com/e2g3g/tutorials/ESIntro/

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 system

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

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?

Characteristic	Conventional Program	Expert System
Control by …	Statement order	Inference engine
Control & Data	Implicit integration	Explicit separation
Control Strength	Strong	Weak
Solution by …	Algorithm	Rules & Inference
Solution search	Small or none	Large
Problem solving	Algorithm	Rules

Characteristic	Conventional Program	Expert system
Input	Assumed correct	Incomplete, incorrect
Unexpected input	Difficult to deal with	Very responsive
Output	Always correct	Varies with the problem
Explanation	None	Usually
Applications	Numeric, file & text	Symbolic reasoning
Execution	Generally sequential	Opportunistic rules

Characteristic	Conventional Program	Expert System
Program Design	Structured design	Little or no structure
Modifiability	Difficult	Reasonable
Expansion	Done in major lumps	Incremental

Dendral

- The Dendral system (DENDRitic ALgorithm) was the first expert system, developed in the 1960s
  - The idea was, given mass spectrogram data, determine what the chemical composition was
- The approach: plan-generate-and-test with human feedback
  - This is a constrained search technique
    - Generate a hypothesis: a possible chemical compound
    - Test the hypothesis: use a series of heuristics and subprograms to determine if the chemical compound generated is plausible given the data
    - If so, show it to the user – the user can *steer* the direction that DENDRAL takes next by suggesting what chemical elements should be added or removed from the generated hypothesis compound
    - DENDRAL repeats until an answer is found acceptable that does not violate the constraints as dictated by the data

MYCIN – problem definition

Diagnosing and treating patients with infectious blood diseases

- Difficulties
  - Time Consuming
  - Misuse and overuse of antibiotics
  - Shortage of expertise

System to help physicians

MYCIN – Why expert system

- An expert was required to solve the problem.
- Experts on the problem were scarce or unavailable because of time constraints.
- Immediate expertise was needed in a possibly life treating situation.
- Time constraints required decisions to be made with limited or inexact information
- The computer solution needed to be accommodating to the user, who may have limited experience with computers.
- Existing solutions may be irrational in cases where drug recommendations were inappropriate for the problem.
- Remembering the appropriateness and possible contradictions of a large number of drugs was a challenge for the physician.

What is MYCIN

- A rule-based expert system
- Developed at Stanford University – 1976
- Uses backward chaining for reasoning
- Incorporates about 500 rules
- Written in INTERLISP (a dialect of LISP)

MYCIN - Major Features

- Using Backward Chaining
- Separate Knowledge from Control
- Incorporates Meta-Rules
- Inexact Reasoning
- Remember Prior Sessions
- Accommodates the User - To perform the function of the expert in acceptable manner, MYCIN had to be easy to use and present itself in manner that was natural to the physician.
  - Natural Language Interaction
  - Spelling Checker
  - Provide Explanation
  - Provide Alternative Recommendation

MYCIN - explanation

- MYCIN can
  - Explain *why* it's asking a question
  - Explain *how* it derived a conclusion
  - Explain *why* it found other result implausible
- Provide alternative solutions if requested

S

Advantages

- Consistent answers for repetitive decisions, processes and tasks
- Holds and maintains significant levels of information
- Encourages organizations to clarify the logic of their decision-making
- Never "forgets" to ask a question, as a human might

Disadvantages

- Lacks common sense
- Cannot make creative responses as human expert
- Domain experts not always able to explain their logic and reasoning
- Errors may occur in the knowledge base
- Cannot adapt to changing environments

Sources

- Sepandar Sepehr, Expert Systems, McMaster University, 2008
- Shahram Rahimi, Intro To Expert Systems, Southern Illinois University
- eXpertise2Go's Rule-Based Expert System Web Content
- Joseph C. Giarratano, Gary D. Riley (Expert Systems: Principles and Programming, Fourth Edition