CmSc 310 Artificial Intelligence

Expert Systems

1. Introduction

**Expert systems** - solve real problems which normally would require a specialized human expert (such as a doctor or a lawyer).

**Knowledge characteristics:**
- heuristic in nature, based on useful "rules of thumb" rather than absolute certainties.
- much of it is almost subconscious, or appears so obvious that experts don't even bother mentioning it.

**Domains:** medicine, mathematics, engineering, geology, computer science, business, law, defense and education. Within each domain, they have been used to solve problems of different types.

**Types of problems** involve:
- *diagnosis* (e.g., of a system fault, disease or student error);
- *design* (of a computer systems, hotel etc); and
- *interpretation* (of, for example, geological data).

The appropriate problem solving technique tends to **depend more on the problem type than on the domain.**

2. Components of an expert system
**Expert system knowledge base:**
Domain inference rules and object descriptions represented by means of some representational scheme—usually rules and frames/semantic nets. Some expert systems work without object descriptions.

**Case specific data** (working storage) - the data which is specific to a problem being solved, entered by the user during a particular session with the expert system

**Knowledge base editor:**
A module that allows creating and updating the knowledge base, supports the operations on database systems - insert, delete, modify, display. Used by the **knowledge engineer**, not by the end-user. The role of knowledge engineering is explained further below.

**Inference engine** - A module that chooses domain inference rules to be executed depending on the specific case. It can implement two basic strategies:
- **backward chaining** = goal-driven inference
- **forward chaining** = data-driven inference

**Forward chaining cycle:**
Select applicable rules (i.e. with conditions satisfied by the current case-specific data)
Choose a rule (conflict resolution)
Execute a rule and update the case-specific data

**Backward chaining:** starts from a pre-defined goal or a fixed small number of goals and executes subsequently rules trying to prove the goal(s) by verifying the conditions in the IF-THEN rules

**Coping with uncertainty** - the ability of the system to reason with rules and data which are not precisely known.

**Mixed reasoning:**
- a. collect some initial symptoms
- b. start with forward chaining to identify plausible hypotheses to be verified
- c. continue with backward chaining to verify the preconditions of each hypothesis

**Explanations** - the ability of the system to explain the reasoning process that it used to reach a recommendation.
User interface - the code that controls the dialog between the users and the system.

3. Knowledge acquisition and knowledge engineering

Domain expert - the individual or individuals who currently are experts solving the problems the system is intended to solve.

Common difficulty: the experts generally find it very difficult to express exactly what knowledge and rules they use to solve a problem.

Knowledge acquisition: The process of "extracting" the knowledge from the domain expert and representing the knowledge in a suitable form that can be used by the expert system.

The process is accomplished through iterations: based on feedback both from the expert and from potential users of the expert system. Performed by the knowledge engineers.

Knowledge engineer:

- conducts the process of extracting knowledge and its formalization (i.e. representing the knowledge using some knowledge representation method/scheme)
- works in collaboration with the experts and the end-users.
- knows how to represent knowledge and how to implement programs that control the domain inference rules. Takes decisions what particular representation to be used depending on the domain and the problem.
- must learn something about the domain.

an on-line textbook on Expert systems, provided by AMZI developers)
**System engineer** - the individual who builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine. Depending on the size of the project, the knowledge engineer and the system engineer might be the same person.

### 4. Explanations

Given that the system knows which rules were used during the inference process, it is possible for the system to provide those rules to the user as a means for explaining its current action and the obtained results.

**Types of Explanation**

a. **Why** a particular question is asked. - usually by describing the rule that has as a condition the answer of the question.

b. **How** a particular conclusion has been reached - usually by listing the rules that have been applied.

c. **Why not** a given conclusion. The user puts forward a hypothesis to be verified.

d. **What if** a particular symptom is present. Sometimes the user may want to see whether a particular symptom would prove a hypothesis or not.

e. **Terminology** - explains the meaning of system terminology to achieve better understanding by the user.

### 5. User interface

**Individuals the interact with the expert system:**

- **The end-user:** To enter specific data for the particular case to be solved
  To require and obtain explanation for the solution

- **The knowledge engineer:** To modify update the knowledge base, to maintain the system

- **The expert:** To verify the formulated rules, to test the performance of the expert system.

**General requirements:**

a. To be self-consistent. (i.e. common style of entering data at the various levels/steps of dialogue, common style of printing messages, etc.)

b. To be consistent with widely used interface systems (helps user feel comfortable).

c. To provide navigation help, i.e. optional cancel of input, going back to previous step, suspending processing, resuming action from a specified step.

d. To incorporate "user model" - the level of user expertise, terminological understanding.
6. **Expert system shells**

The shell is a piece of software which contains the user interface, a format for declarative knowledge in the knowledge base, and an inference engine. The knowledge engineer uses the shell to build a system for a particular problem domain.

7. **Expert systems demos**

[http://www.expertise2go.com/webesie/](http://www.expertise2go.com/webesie/)
eXpertise2Go's Web-Enabled Expert Systems Demos