KBS : Material

- **Introduction**

Knowledge Representation
  - Semantic Nets, Frames, Logic
- Reasoning and Inference
  - Predicate Logic, Inference Methods, Resolution
- Reasoning with Uncertainty
  - Probability, Bayesian Decision Making
- Pattern Matching
  - Variables, Functions, Expressions, Constraints
- Expert System Design
  - ES Life Cycle
- Expert System Implementation
  - Salience, Rete Algorithm
- Expert System Examples
- Conclusions and Outlook

KBS : Introduction

- Motivation
- Objectives
- What is an Expert System (ES)?
  - knowledge, reasoning
- General Concepts and Characteristics of ES
  - knowledge representation, inference, knowledge acquisition, explanation
- ES Technology
- ES Tools
  - shells, languages
- ES Elements
  - facts, rules, inference mechanism
- Important Concepts and Terms
- Chapter Summary

Textbooks

- Required
  - [Giarratano & Riley 1998] Joseph Giarratano and Gary Riley. *Expert Systems - Principles and Programming.* 4th ed., PWS Publishing, Boston, MA, 2004
- Recommended for additional reading
  - [Awad 1996] Elias Awad. *Building Expert Systems - Principles, Procedures, and Applications.* West Publishing, Minneapolis/St. Paul, MN, 1996.
  - [Durkin 1994] John Durkin. *Expert Systems - Design and Development.* Prentice Hall, Englewood Cliffs, NJ, 1994.
  - [Jackson, 1999] Peter Jackson. *Introduction to Expert Systems.* 3rd ed., Addison-Wesley, 1999.
  - [Russell & Norvig 1995] Stuart Russell and Peter Norvig, *Artificial Intelligence - A Modern Approach.* Prentice Hall, 1995.
  - [M. Arhami, 2005] Muhammad Arhami, *Konsep Dasar Sistem Pakar*, Andi, Yogya, 2005.

Motivation

- utilization of computers to deal with knowledge
  - quantity of knowledge available increases rapidly
  - relieve humans from tedious tasks
- computers have special requirements for dealing with knowledge
  - acquisition, representation, reasoning
- some knowledge-related tasks can be solved better by computers than by humans
  - cheaper, faster, easily accessible, reliable

Objectives

- to know and comprehend the main principles, components, and application areas for expert systems
- to understand the structure of expert systems
  - knowledge base, inference engine
- to be familiar with frequently used methods for knowledge representation in computers
- to evaluate the suitability of computers for specific tasks
  - application of methods to scenarios or tasks

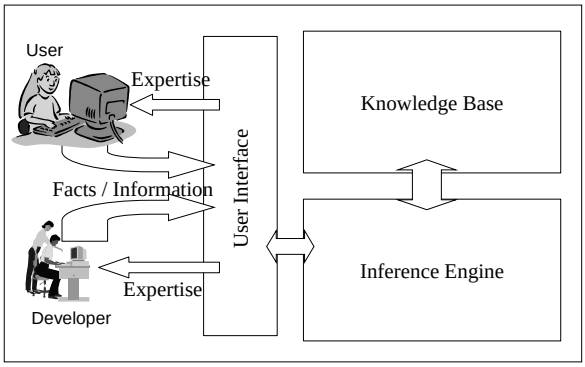
What is an Expert System (ES)?

- relies on internally represented knowledge to perform tasks
- utilizes reasoning methods to derive appropriate new knowledge
- usually restricted to a specific *problem domain*
- some systems try to capture common-sense knowledge
  - General Problem Solver (Newell, Shaw, Simon)
  - Cyc (Lenat)

Definitions "Expert System"

- a computer system that emulates the decision-making ability of a human expert in a restricted domain [Giarratano & Riley 1998]
- Edward Feigenbaum
  - "An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions." [Giarratano & Riley 1998]
- the term *knowledge-based system* is often used synonymously

Main Components of an ES



Main ES Components

- 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

General Concepts and Characteristics of ES

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

Development of ES Technology

- strongly influenced by cognitive science and mathematics
  - the way humans solve problems
  - formal foundations, especially logic and inference
- production rules as representation mechanism
  - `IF … THEN` type rules
  - reasonably close to human reasoning
  - can be manipulated by computers
  - appropriate granularity
    - knowledge "chunks" are manageable both for humans and for computers

[Dieng et al. 1999]

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
  - sensory input or thinking triggers the activation of rules
  - activated rules may trigger further activation
  - a cognitive processor combines evidence from currently active rules
- this model is the basis for the design of many rule-based systems
  - also called *production systems*

Early ES Success Stories

- DENDRAL
  - identification of chemical constituents
- MYCIN
  - diagnosis of illnesses
- PROSPECTOR
  - analysis of geological data for minerals
  - discovered a mineral deposit worth $100 million
- XCON/R1
  - configuration of DEC VAX computer systems
  - saved lots of time and millions of dollars

The Key to ES 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

When (Not) to Use ESs

- expert systems are not suitable for all types of domains and tasks
  - conventional algorithms are known and efficient
  - the main challenge is computation, not knowledge
  - knowledge cannot be captured easily
  - users may be reluctant to apply an expert system to a critical task

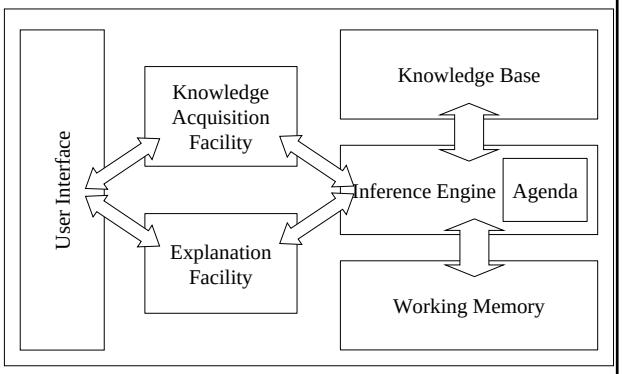
ES Tools

- ES languages
  - higher-level languages specifically designed for knowledge representation and reasoning
  - SAIL, KRL, KQML, DAML
- ES shells
  - an ES development tool/environment where the user provides the knowledge base
  - CLIPS, JESS, Mycin, Babylon, ...

ES Elements

- knowledge base
- inference engine
- working memory
- agenda
- explanation facility
- knowledge acquisition facility
- user interface

ES Structure



Rule-Based ES

- knowledge is encoded as **`IF … THEN`** rules
  - these rules can also be written as *production rules*
- the inference engine determines which rule antecedents are satisfied
  - the left-hand side must "match" a fact in the working memory
- satisfied rules are placed on the agenda
- rules on the agenda can be activated ("fired")
  - an activated rule may generate new facts through its right-hand side
  - the activation of one rule may subsequently cause the activation of other rules

Example Rules

`IF … THEN` Rules

```
Rule: Red_Light
  IF    the light is red
  THEN  stop
Rule: Green_Light
  IF    the light is green
  THEN  go
```

antecedent (left-hand-side)

consequent (right-hand-side)

Production Rules

antecedent (left-hand-side)

```
the light is red ==> stop
the light is green ==> go
```

consequent (right-hand-side)

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  the there is strongly suggestive 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)
```

[Durkin 94, p. 133]

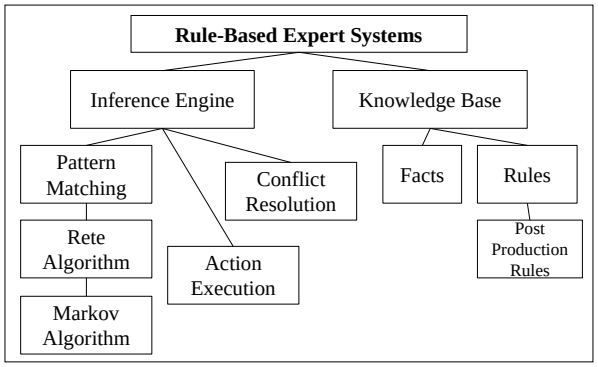
Inference Engine Cycle

- describes the execution of rules by the inference engine
  - conflict resolution
    - select the rule with the highest priority from the agenda
  - execution
    - perform the actions on the consequent of the selected rule
    - remove the rule from the agenda
  - match
    - update the agenda
      - add rules whose antecedents are satisfied to the agenda
      - remove rules with non-satisfied agendas
- the cycle ends when no more rules are on the agenda, or when an explicit stop command is encountered

Forward and Backward Chaining

- different methods of rule activation
  - forward chaining (data-driven)
    - reasoning from facts to the conclusion
    - as soon as facts are available, they are used to match antecedents of rules
    - a rule can be activated if all parts of the antecedent are satisfied
    - often used for real-time expert systems in monitoring and control
    - examples: CLIPS, OPS5
  - backward chaining (query-driven)
    - starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
    - often used in diagnostic and consultation systems
    - examples: EMYCIN

Foundations of Expert Systems



Post Production Systems

- production rules were used by the logician Emil L. Post in the early 40s in symbolic logic
- Post's theoretical result
  - any system in mathematics or logic can be written as a production system
- basic principle of production rules
  - a set of rules governs the conversion of a set of strings into another set of strings
    - these rules are also known as *rewrite rules*
    - simple syntactic string manipulation
    - no understanding or interpretation is required
    - also used to define grammars of languages
      - e.g. BNF grammars of programming languages

Markov Algorithms

- in the 1950s, A. A. Markov introduced *priorities* as a control structure for production systems
  - rules with higher priorities are applied first
  - allows more efficient execution of production systems
  - but still not efficient enough for expert systems with large sets of rules

Rete Algorithm

- developed by Charles L. Forgy in the late 70s for CMU's OPS (Official Production System) shell
  - stores information about the antecedents in a network
  - in every cycle, it only checks for changes in the networks
  - this greatly improves efficiency

ES 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

ES 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 ES 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

Summary Introduction

- expert systems or knowledge based systems are used to represent and process 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
- ES can be cheaper, faster, more accessible, and more reliable than humans
- ES have limited knowledge (especially "common-sense"), can be difficult and expensive to develop, and users may not trust them for critical decisions

Important Concepts and Terms

- agenda
- backward chaining
- common-sense knowledge
- conflict resolution
- expert system (ES)
- expert system shell
- explanation
- forward chaining
- inference
- inference mechanism
- If-Then rules
- knowledge
- knowledge acquisition
- knowledge base
- knowledge-based system
- knowledge representation
- Markov algorithm
- matching
- Post production system
- problem domain
- production rules
- reasoning
- RETE algorithm
- rule
- working memory