CPE/CSC 481: Knowledge-Based Systems

Franz J. Kurfess

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA, U.S.A.

CAL POLY

Logistics - Jan 10, 2013

- **Enrollment**
  - send me email with reasons for adding
    - major, planned graduation, number of units, need for electives, other
- **Project**
  - Topics and Teams
- **KB Nugget presentations**
  - Topics
  - Signup for Date & Time Slots
- **Quiz**
  - Quiz 0 available
  - background, mostly 480
- **Assignments**
  - A1: Concept Map

Course Overview

- **Introduction**
  - Knowledge-Based Systems (KBS), Expert Systems (ES)
  - Data/Information/Knowledge
- **Knowledge Representation**
  - Semantic Nets, Rules, Frames, Scripts, Logic, RDF
- **Reasoning and Inference**
  - Predicate Logic, Description Logics, Inference Methods, Resolution
- **Reasoning with Uncertainty**
  - Probability, Bayesian Decision Making
- **Approximate Reasoning**
  - Fuzzy Logic
- **Knowledge Exchange**
  - Capture, Transfer, Distribution
- **Knowledge Retrieval**
  - Search, Queries, Data Mining
- **KBS Implementation**
  - Unification, Pattern Matching, Salience, Rete Algorithm
- **KBS Examples**
  - CLIPS/Jess, Prolog, Semantic Web Technologies
- **Conclusions and Outlook**

Overview Introduction

- **Motivation**
- **Objectives**
- **What is a Knowledge-Based System (KBS)?**
  - knowledge, reasoning
- **General Concepts and Characteristics of KBSs**
  - knowledge representation, inference, knowledge acquisition, explanation
- **KBS Technology**
- **KBS Tools**
  - shells, languages
- **KBS Elements**
  - facts, rules, inference mechanism
- **Important Concepts and Terms**
- **Chapter Summary**

Motivation

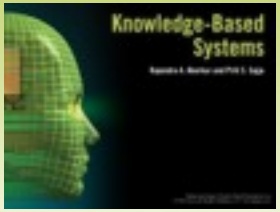
- **utilization of computers to deal with knowledge**
  - quantity of knowledge available increases rapidly
  - relieve humans from tedious tasks
- **computers have special requirements for dealing with knowledge**
  - acquisition, representation, reasoning
- **some knowledge-related tasks can be solved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

Objectives

- **to know and comprehend the main principles, components, and application areas for Knowledge-Based Systems**
- **to understand the structure of Knowledge-Based 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

Terminology

- Data
- Information
- Knowledge



Data Pyramid and Computer-Based Systems

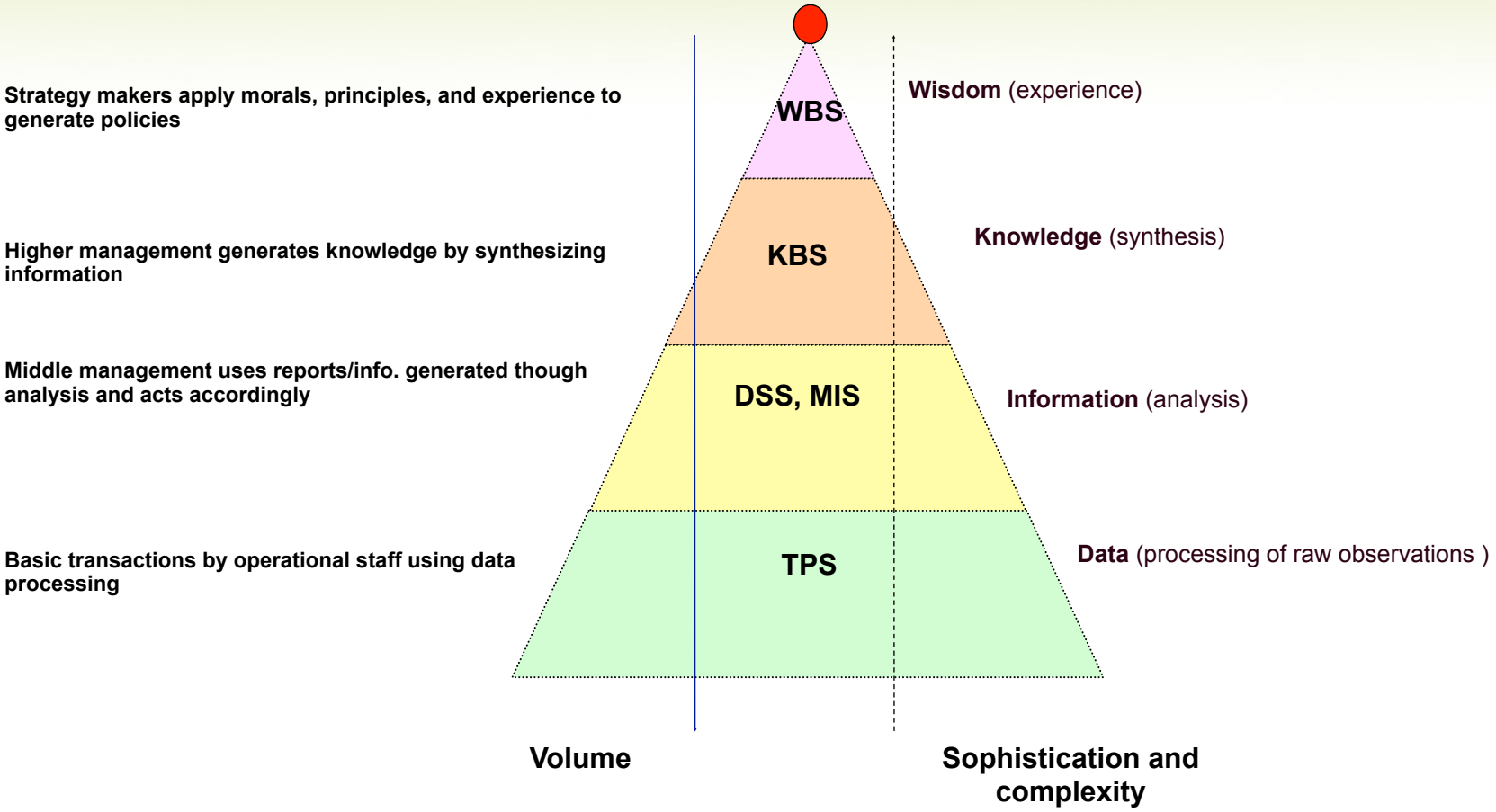
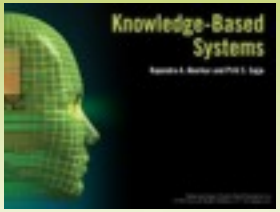


Figure 1.7: Data pyramid: Managerial perspectives



Data Pyramid and Computer Based Systems

Heuristics and models

Rules

Concepts

Raw data through fact finding

Wisdom

Knowledge

Information

Data

Novelty

Experience

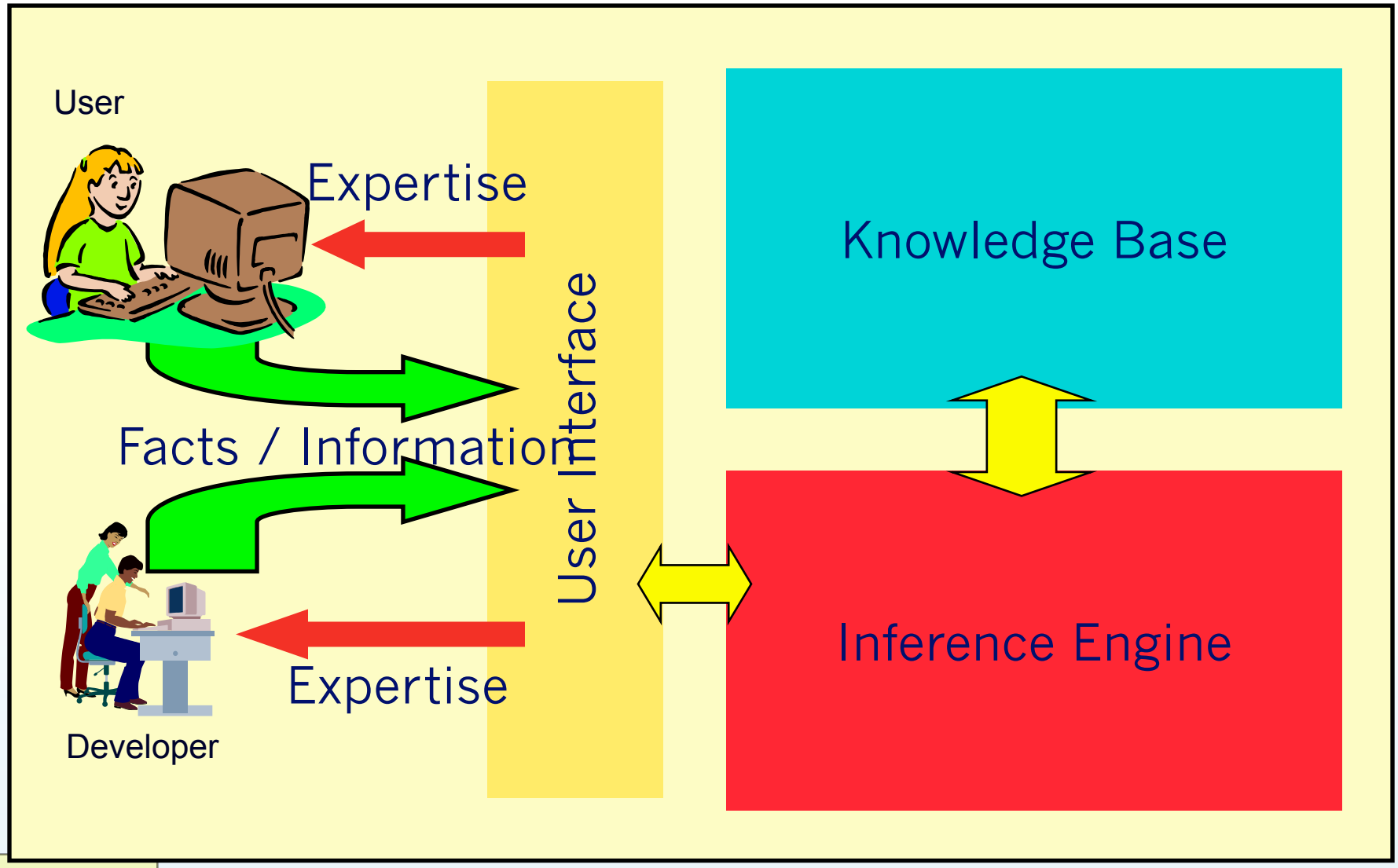
Understanding

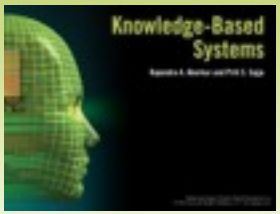
Researching Absorbing Doing Interacting Reflecting

What is an Knowledge-Based System (KBS)?

- 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)

Main Components of a KBS



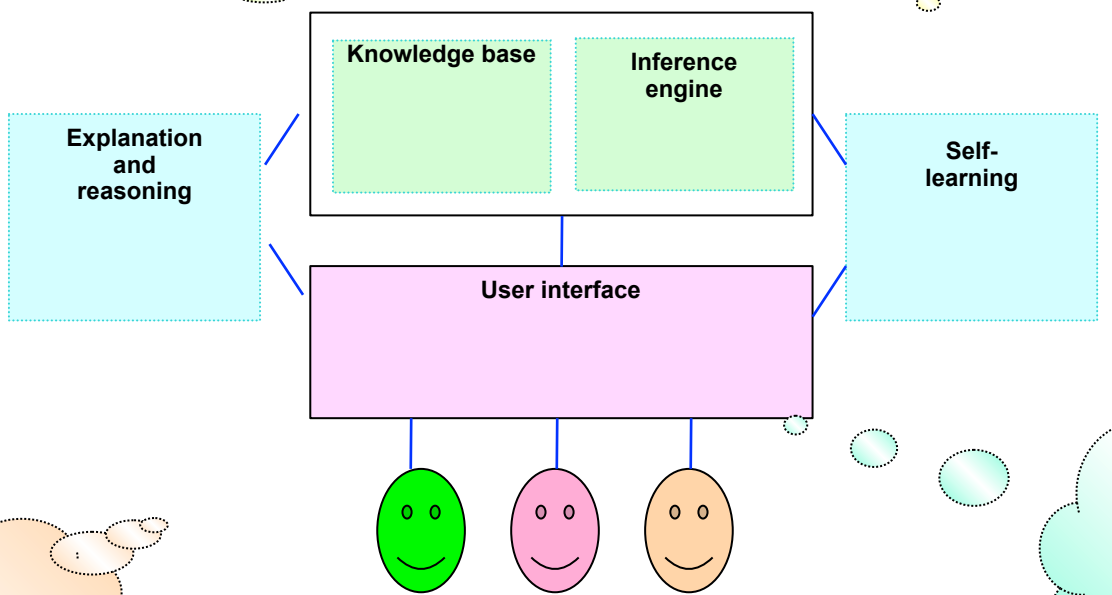


Components of KBS

Knowledge base is a repository of domain knowledge and metaknowledge.

Inference engine is a software program that infers the knowledge available in the knowledge base.

Enriches the system with self-learning capabilities



Provides explanation and reasoning facilities

Friendly interface to users working in their native language

Figure 1.10: General structure of KBS

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

Early KBS 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

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

Related Developments

- **Semantic Web**
  - extension of the World Wide Web
  - includes knowledge representation and reasoning capabilities
- **Decision Support Systems**
  - less emphasis on autonomy
- **Data Mining**
  - extraction of knowledge from large quantities of data
- **Sense-making**
  - computer support for quicker, easier understanding of complex domains or situations

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

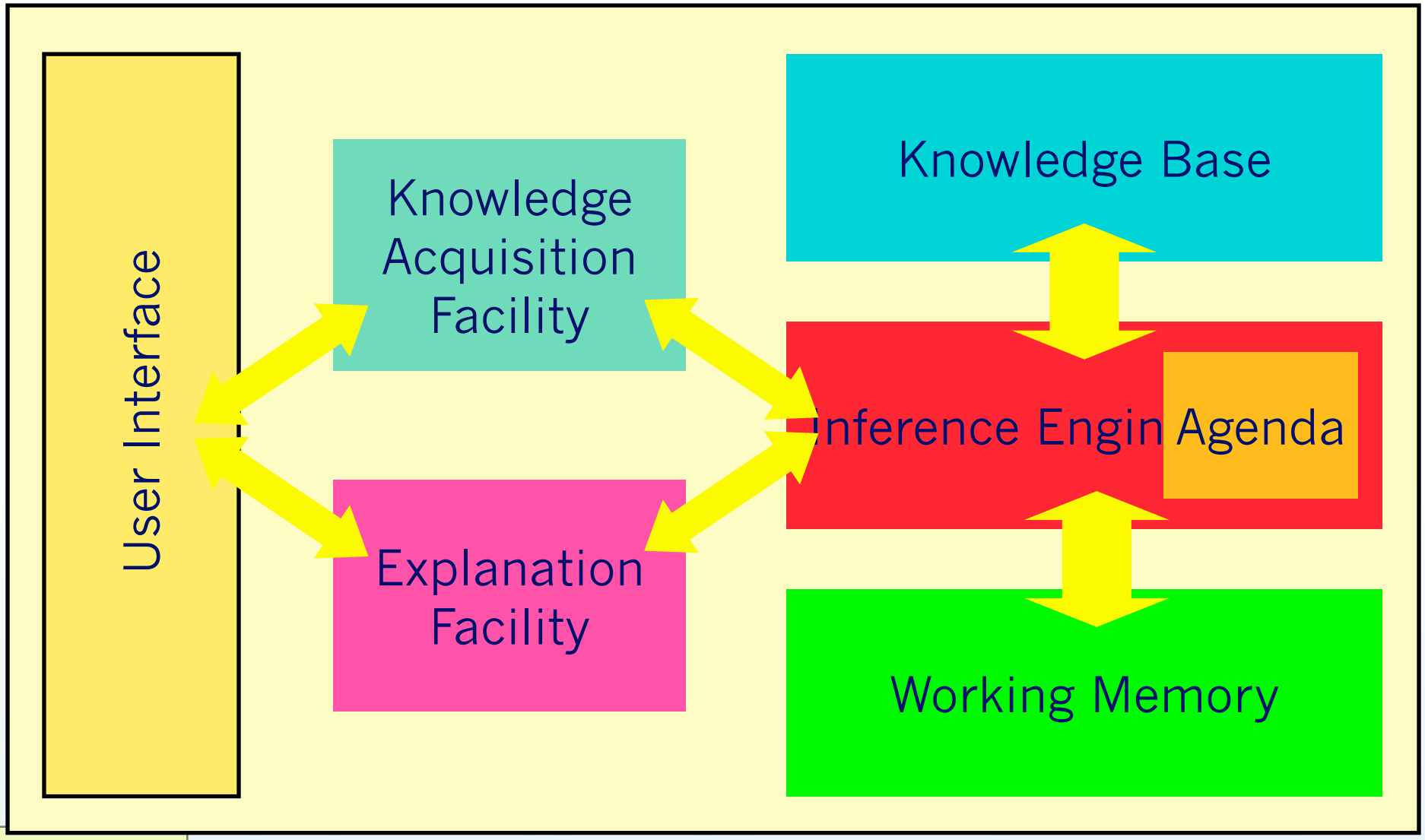
```
IF (AND (SAME CNTEXT GRAM GRAMNEG)
        (SAME CNTEXT MORPH ROD)
        (SAME CNTEXT AIR AEROBIC)
THEN (CONCLUDE CNTEXT CLASS ENTEROBACTERIACEAE
     TALLY .8)
```

[Durkin 94, p. 133]

KBS Elements

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

KBS Structure Details



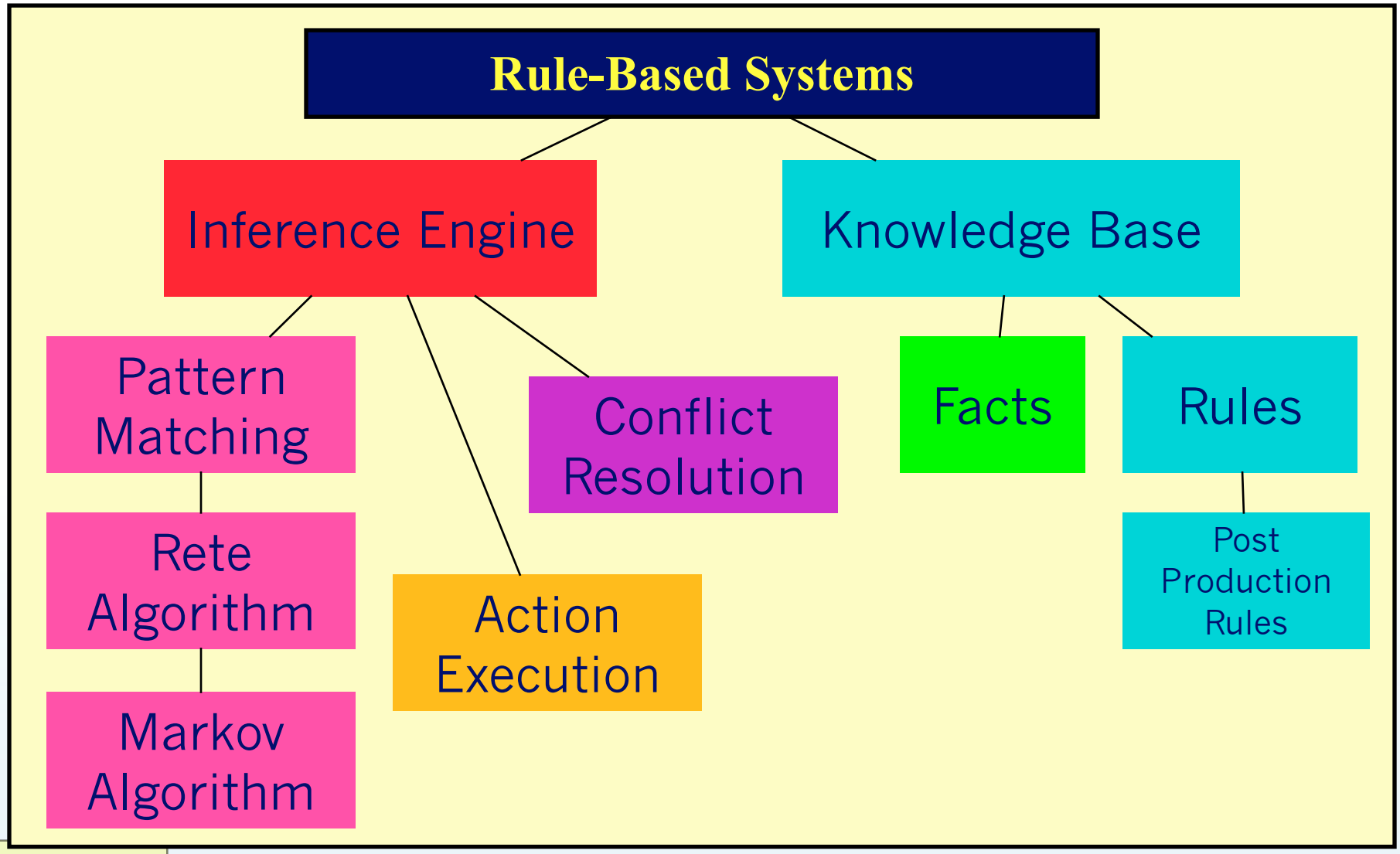
Inference Engine Cycle

- **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**
  - no more rules are on the agenda
  - explicit “stop” command

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 KBSs



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

Emil Post

- **20th century mathematician**
- **worked in logic, formal languages**
  - truth tables
  - completeness proof of the propositional calculus as presented in *Principia Mathematica*
  - recursion theory
    - mathematical model of computation similar to the Turing machine
- **not related to Emily Post ;-)**

http://en.wikipedia.org/wiki/Emil_Post

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 Knowledge-Based Systems with large sets of rules
  - he is the son of Andrey Markov, who developed Markov chains

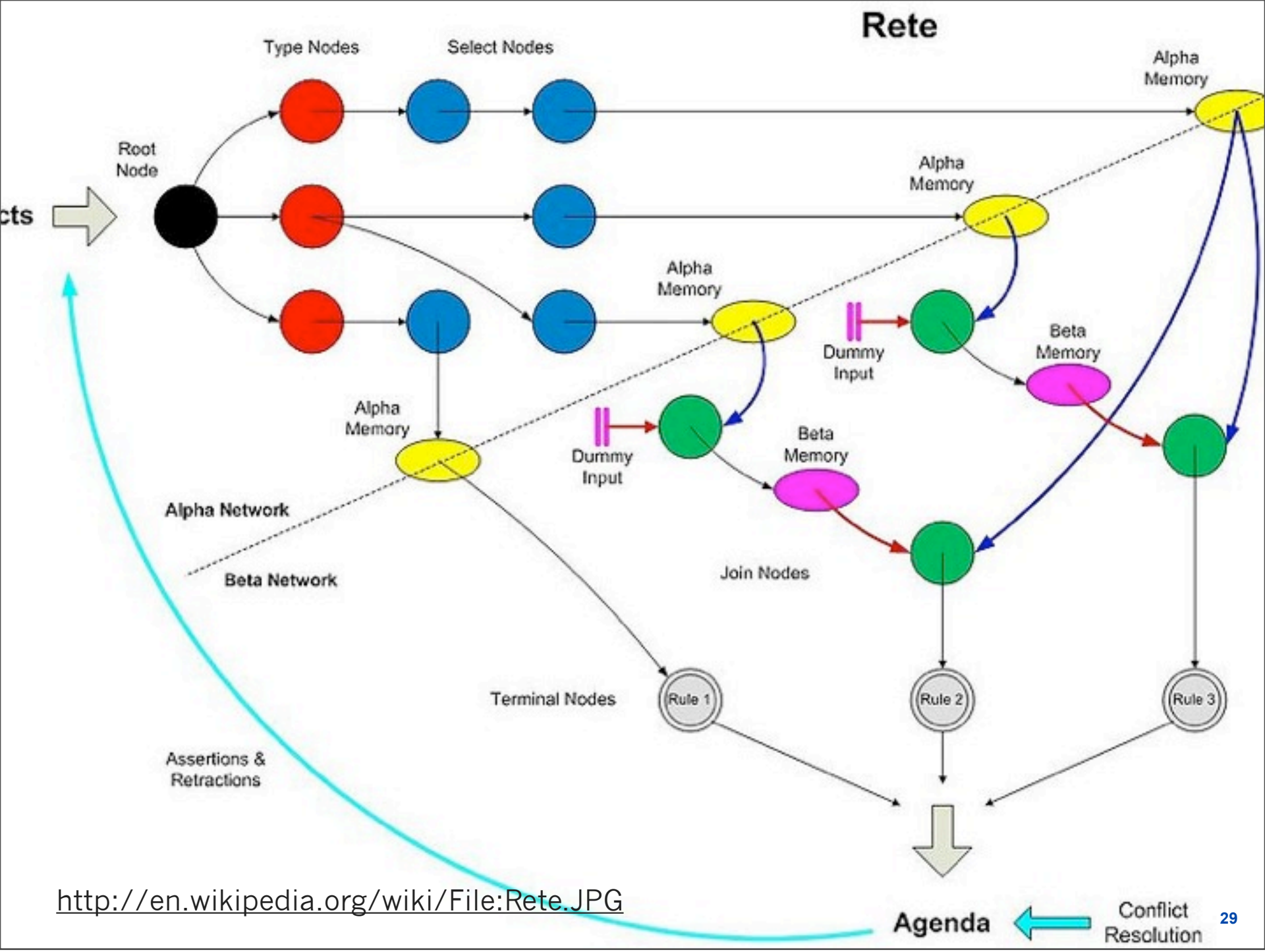
Markov Andrei Andreevich

http://en.wikipedia.org/wiki/File:AAMarkov.jpg
http://www.ras.ru/win/db/show_per.asp?P=.id-53175.ln-en.dl-.pr-inf.uk-0

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

http://rulesfest.org/graphics/2011_speakers/bio_Forgy_Charles.png



http://en.wikipedia.org/wiki/File:Rete.JPG

KBS 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

KBS 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 Knowledge-Based 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