Expert Systems
- Introduction -

Gerald Steinbauer
Institut für Softwaretechnologie
Inffeldgasse 16b/2
A-8010 Graz
Austria
References

- Skriptum (TU Wien, Institut für Informationssysteme, Thomas Eiter et al.) ÖH-Copyshop, Studienzentrum


- Vorlesungsfolien TU Graz (teilweise basierend auf den Folien der TUWien)
Clear Need for KBS
Goals

Introduction (KBS)
Example Applications
What is a KBS?

A **Knowledge-based System (KBS)** is a system implemented with the goal to imitate human problem-solving behavior using **Artificial Intelligence (AI)** techniques.

Development and maintenance of **KBS** is the objective of **Knowledge Engineering (KE)**.

Many commonalities with **Software Engineering (SE)**.
Artificial Intelligence?

**Artificial intelligence** (AI) research is concerned with the automation of tasks requiring intelligent behavior.

**Strong AI** supposes that it is possible for machines to become sapient, or self-aware, but may or may not exhibit human-like thought processes.

**Weak AI** supports the accomplishment of specific problem solving tasks not encompassing the full range of human cognitive abilities.
What is Knowledge Engineering?

Knowledge engineers work with domain experts to acquire domain knowledge. This knowledge is then implemented in a knowledge-based system (KBS) using AI methods such as search, machine learning, and game playing. The technical knowledge is also implemented by the knowledge engineers.

Domain knowledge acquisition is a bottleneck. End users are involved in the implementation by end-users.
What is Knowledge?

• **Data:** symbols
  - 1, 2
  - Jaguar
  - -2°
  - rainy

• **Information:** data with given meaning (semantics)
  - temp= -2°
  - weather= rainy

• **Knowledge:** application of data & information
  - temp<0° → street-temp=cold
  - street-temp=cold → driving-cond=slippery
Types of Knowledge

- Implicit knowledge vs. explicit knowledge: skills, insights, etc. vs., for example, a technical manual.

- Deep knowledge vs. shallow knowledge: generalized knowledge vs. “surface level” knowledge, e.g., “excellent driving behavior in mountains (X5)” vs. “4-wheel tech is excellent for mountain tours …”

- Meta-Knowledge: knowledge about knowledge, e.g., $p(\text{type}(X, \text{car}) \land \text{producer}(X, \text{BMW})) = 0.001$

- Declarative vs. procedural knowledge: for example, “two CPU’s of type CPUa have to be part of the PC configuration” vs. “if CPU1 is inserted continue with the insertion of CPU2”
Types of Knowledge

**Declarative:**
\[ \text{prim}(n): (n>1) \land \forall m (\text{teilt}(m,n) \rightarrow (m=1 \lor m=n)) \]

**Non-Declarative:**

```pascal
function prim(n: integer): Boolean;
var i: integer;
begin
  if n <= 1 then prim := false
  else begin
    i := 2;
    while (i <= trunc(sqrt(n))+1) and
      (n mod i <> 0) do i := i+1;
    prim := (n=2) or (n mod i <> 0)
  end
end /* prim */
```
Requirements regarding KBS

Strict separation of
- domain knowledge
- problem solving knowledge

Storage & organization of knowledge

Knowledge should be understandable for (end) users

Extensibility and changeability for new knowledge

Transparency: traceability of solution search
Knowledge Representation

Cognitive AI:
- Knowledge engineering & processing follows patterns of (human) intelligence
- Theory of intelligent behavior

Rational AI:
- Knowledge engineering & processing is result-oriented
- No direct correspondence with intelligent behavior

KBS:
- Typically follow the Rational AI approach
- Formalisms for knowledge representation & processing: logics
- KBS often part of a complex system
History of AI

Roots in philosophy, mathematics, psychology, and CS

Early phase: 1943 – 1956

Early works:
- Neural networks (McCulloch and Pitt, 1943)
- Chess programs (Turin & Shannon)
- Logic Theorist – Theorem Prover (Newell & Simon)

AI – hour of birth:
- Dartmouth Workshop Summer 1956 (J. McCarthy)
Era of Strong AI (1956-1970)

Powerful problem solving mechanisms:
- General Problem Solver (Newell & Simon, 1963)
- LISP (McCarthy, 1958)
- Resolution Method (Robinson, 1965)

Disillusion followed:
- Chess games not performing well
- Low quality automated translators
- „Up-scaling“ problems – transition from toy to real-world problems
Era of Weak AI (1970-1980)

Systems for specific tasks
Supporting human agents
Domain-specific solutions
Sometimes incomplete, heuristics used

Example: **Expert systems**
- Knowledge permanently available
- Low-cost development & maintenance
- Examples follow …
Classic Expert Systems (XPS)

Dendral (~1970 – Feigenbaum, Buchanan et al.)
  – used for determining the structure of molecules from mass spectrograms of chemical compounds

MYCIN (~1973 – Shortliffe et al.)
  – diagnosis of bacterial blood infections + antimicrobial recommendations of therapies
  – detailed explanations for solutions
  – More than 100 rules
  – certainty factors, for example …

If 1) the infection is primary-bacteremia, and
  2) the site of the culture is one of the sterile sites, and
  3) the suspected portal of entry of the organism is the gastro-intestinal tract,
Then there is suggestive evidence (0.7) that the identity of the organism is bacteroides
Classic Expert Systems (XPS)

- supports geologists in the identification of ore ("Erz") deposits
- based on statistical approaches (probabilities)
- was successful in identifying molybdenum deposits

R1/XCON (~1982 – McDermott)
- rule-based system for the configuration of VAX systems (DEC)
- saved millions of dollars
- 17,500 rules, 30,000 components

Today: thousands of XPS applications in different domains such as medicine, technical systems, and services.
AI as Industry (1980-1991)

Successful XPS applications triggered boom in AI research
- AI programming languages
- Knowledge representation systems (embedded in applications)
- Knowledge engineering as a discipline

5th generation project (Japan, 1981 – 1991)
- Prolog as Hardware language

CYC project (Lat, Guha et al., end of 1980s)
- goal: complex knowledge base for general knowledge
- focus: representation of large chunks of knowledge

Big projects not successful
- Complexity underestimated
- Deep knowledge about problem domain needed (knowledge hypothesis of Lenat & Feigenbaum, 1987)
State of the Art Systems

PEGASUS: flight booking via NL communication

XPS MARVEL: control of the NASA Voyager Mission

DEEP BLUE (IBM) beats G. Kasparow (in May 1997) – massive exploitation of computational power

ILOG (optimization libraries): e.g., scheduling & configuration (constraint satisfaction technologies)

AMAZON.COM: intelligent recommendation services

SHYSTER: legal information system (case-based reasoning)

TROLLCREEK (oil drilling): case-based reasoning, model-based reasoning, semantic web

Crowd Simulation: agent-based systems (e.g., in Lord of the Rings or King Kong)

Watson (Jeopardy!): combination of large knowledge bases, effective search and machine learning
Architecture of KBS

User Interface

Explanation

Inference Engine

Knowledge Base

Knowledge Acquisition

User

Knowledge Engineer
Architecture of KBS

Knowledge Base:
- Knowledge in declarative form
- Typically: rules and facts
- e.g., dog(pippin), \( \forall x. \text{dog}(x) \rightarrow \text{eats}(x, \text{bones}) \)
- generic knowledge vs. case-specific knowledge

Inference Engine:
- Knowledge processing on the basis of facts and rules
- Different inference types, e.g., temporal or spatial reasoning
Architecture of KBS

Knowledge Acquisition:
- Manual or automated adaptations
- Consistency checks, diagnosis & repair

User Interface:
- End-user / domain expert (interactive dialog)
- Knowledge engineer (development & maintenance)

Explanation:
- How was the solution derived?
- Why was a certain question posed? Why does the system propose a set of repair actions?
Agents

- **perceive** the environment through sensors (→ percepts)
- **act** upon the environment through actuators (→ actions)

- **examples**: humans and animals, robots and software agents (softbots), temperature control, ABS, . . .
Rational Agent

• … does the “right thing”!
• rational behavior is dependent on
  • performance measures (goals)
  • percept sequences
  • knowledge of the environment
  • possible actions
• for each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.
Environment of Rational Agents I

- **accessible vs. inaccessible** (fully observable vs. partially observable)
  - are the relevant aspects of the environment accessible to the sensors?

- **deterministic vs. stochastic**
  - is the next state of the environment completely determined by the current state and the selected action? if only actions of other agents are nondeterministic, the environment is called strategic.

- **episodic vs. sequential**
  - can the quality of an action be evaluated within an episode (perception + action), or are future developments decisive for the evaluation of quality?
Environment of Rational Agents II

- **static vs. dynamic**
  - can the environment change while the agent is deliberating? if the environment does not change but if the agent’s performance score changes as time passes by the environment is denoted as semi-dynamic.

- **discrete vs. continuous**
  - is the environment discrete (chess) or continuous (a robot moving in a room)?

- **single agent vs. multi-agent**
  - which entities have to be regarded as agents? there are competitive and cooperative scenarios.
## Examples of Environments

<table>
<thead>
<tr>
<th>task</th>
<th>observable</th>
<th>deterministic</th>
<th>episodic</th>
<th>static</th>
<th>discrete</th>
<th>agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>crossword puzzle</td>
<td>fully</td>
<td>deterministic</td>
<td>sequential</td>
<td>static</td>
<td>discrete</td>
<td>single</td>
</tr>
<tr>
<td>chess with clock</td>
<td>fully</td>
<td>strategic</td>
<td>sequential</td>
<td>semi</td>
<td>discrete</td>
<td>multi</td>
</tr>
<tr>
<td>poker</td>
<td>partially</td>
<td>stochastic</td>
<td>sequential</td>
<td>static</td>
<td>discrete</td>
<td>multi</td>
</tr>
<tr>
<td>backgammon</td>
<td>fully</td>
<td>stochastic</td>
<td>sequential</td>
<td>static</td>
<td>discrete</td>
<td>multi</td>
</tr>
<tr>
<td>taxi driving</td>
<td>partially</td>
<td>stochastic</td>
<td>sequential</td>
<td>dynamic</td>
<td>continuous</td>
<td>multi</td>
</tr>
<tr>
<td>medical diagnosis</td>
<td>partially</td>
<td>stochastic</td>
<td>sequential</td>
<td>dynamic</td>
<td>continuous</td>
<td>single</td>
</tr>
<tr>
<td>image analysis</td>
<td>fully</td>
<td>deterministic</td>
<td>episodic</td>
<td>semi</td>
<td>continuous</td>
<td>single</td>
</tr>
<tr>
<td>part-picking robot</td>
<td>partially</td>
<td>stochastic</td>
<td>episodic</td>
<td>dynamic</td>
<td>continuous</td>
<td>single</td>
</tr>
<tr>
<td>refinery controller</td>
<td>partially</td>
<td>stochastic</td>
<td>sequential</td>
<td>dynamic</td>
<td>continuous</td>
<td>single</td>
</tr>
<tr>
<td>interactive English tutor</td>
<td>partially</td>
<td>stochastic</td>
<td>sequential</td>
<td>dynamic</td>
<td>discrete</td>
<td>multi</td>
</tr>
</tbody>
</table>
Model-based Reflex Agent

- in case the agent’s **history** in addition to the actual percept is required to decide on the next action, it must be **represented** in a suitable form (model)
Goal-based Agent

- often, percepts alone are insufficient to decide what to do
- model-based, goal-based agents use an explicit representation of goals and consider them for the choice of actions
Related Literature

Little Exercise (Groups á 2 Persons)

- a simple knowledge-based system
  - select a domain for KBS
  - classify the environment of the system
  - define the types of knowledge the system needs
  - encode some knowledge for the different types

- does your system follow the strong AI or the weak AI paradigm?
  - provide an answer and corresponding argumentation for your answer
Thank You!