VU Einführung in Wissensbasierte Systeme

WS 2010/11

Hans Tompits

Institut für Informationssysteme
Arbeitsbereich Wissensbasierte Systeme

www.kr.tuwien.ac.at

Organisatorisches

Lehrveranstaltungsüberblick

- Vortragende:

  VO-Teil: Uwe Egly, Hans Tompits
  UE-Teil: Uwe Egly, Hans Tompits,
  Michael Fink, Magdalena Ortiz

- Ablauf und Inhalt des VO-Teils:

  Di., 12.10. Einführung (Tompits)
  Do.,14.10. Suche 1 (Egly)
  Do.,21.10. Suche 2 (Egly)
  Fr., 22.10. Constraint Satisfaction Probleme (Tompits)
  Zeit: 14-16h, Ort: HS 8 (Änderung!)
  Di., 28.10. Logik 1 (Egly)
  Do.,04.11. Logik 2 (Egly)
  Di., 09.11. Nichtmonotones Schließen 1 (Tompits)
  Do.,11.11. Nichtmonotones Schließen 2 (Tompits)
  Di., 16.11. Regelbasierte Systeme 1 (Egly)
  Do.,18.11. Regelbasierte Systeme 2 (Egly)

UE-Teil

- Anmeldung verpflichtend!
- 3 UE-Blätter
  - Blatt 1: Suche & CSP
  - Blatt 2: Logik
  - Blatt 3: Nichtmonotones Schließen & Regelbasierte Systeme
- 10 UE-Gruppen, je 3 Termine entsprechend den 3 UE-Blättern.
  - 1. Termin: Kalenderwoche 45
  - 2. Termin: Kalenderwoche 47
  - 3. Termin: Kalenderwoche 48
- Modus: “Kreuzerl-Liste”
  - die angekreuzten Beispiele müssen an der Tafel vorgerechnet und erklärt werden können.
- Manche UE-Einheiten werden in Englisch abgehalten!

Hinweise & Prüfung

- Folien werden auf der LV-Webseite zur Verfügung stehen
- Prüfung: schriftlich, ohne Hilfsmittel
  - Termin: 09.12.2010, 14:00-16:00
  - Ort: Audi Max

Benotung

- UE-Teil:
  - Jedes angekreuzte Beispiel gibt ein oder mehrere Punkte.
    - Für manche Beispiele, die aus mehreren Teilaufgaben bestehen, wird es auch möglich sein, mehrere Kreuze anzugeben.
    - Die UE-Blätter liefern insgesamt maximal 16 Punkte, auch wenn > 16 Aufgaben!
  - Die maximale Punkteanzahl für ein Vorrechnen an der Tafel ist 9. Für mehr als einmal Vorrechnen wird das Maximum genommen.
- VO-Teil:
  - Die Prüfung liefert maximal 80 Punkte.

➥ In Summe kann man somit 105 Punkte erreichen (100 Punkte + 5 Bonus Punkte).

Benotung

- Für eine positive Beurteilung benötigt man
  - $\geq$ 9 Punkte für die angekreuzten Beispiele,
  - $\geq$ 1 Punkt für das Vorrechnen an der Tafel, und
  - $\geq$ 40 Punkte für die Prüfung.
- Keine Toleranz für unehrliche Methoden (z.B. kopierte Lösungen).
  - Punkteabzug für alle involvierten Personen (bis hin zu negativer Benotung).
- Notenschlüssel:

  $<$ 50 Punkte: Nicht Genügend,
  $\geq$ 50 Punkte: Genügend,
  $\geq$ 62 Punkte: Befriedigend,
  $\geq$ 74 Punkte: Gut,
  $\geq$ 86 Punkte: Sehr Gut.

Literatur

- Unterschiedliche Quellen.
- Bekannte Lehrbücher:
  - C. Beierle, G. Kern-Isberner:
    Methoden wissensbasierter Systeme, 4. Auflage.
    Vieweg 2008
  - S. Russell, P. Norvig:
    Artificial Intelligence: A Modern Approach (Third Edition).
    Prentice Hall, 2009
  - N. J. Nilsson:
    Artificial Intelligence - A New Synthesis. Morgan Kaufmann Publishers, 1998
  - M. Genesereth, N. J. Nilsson: Logical Foundations of Artificial Intelligence, Morgan Kaufmann Publishers, 1987
  - ...

Begleitende & weiterführende Lehrveranstaltungen

- LU 1.0 Einführung in Wissensbasierte Systeme
  → Beginn, Mi., 13.10., 15h, EI 8
- VO 2.0 Logik für Wissensrepräsentation
  → Beginn, Fr., 15.10., 16h, EI 11
- Seminare aus den Masterstudien
  Verschiedene Themen, u.A.
  - Aktuelle Trends der Answer-Set Programmierung
  - Wissensverarbeitung im Next-Generation Web
  - Geschichte der Logik

  → Themenausgabe: Mi., 27.10., 13h, Seminarraum Gödel

Begleitende & weiterführende Lehrveranstaltungen

- VU 2.0 Nichtmonotones Schließen
- VU 2.0 Theorie der Wissensrepräsentation
- VO 2.0 Verarbeitung deklarativen Wissens
- Praktika (jederzeit)
  - Software (Experimente, Installation,. . .)
  - Theorie

Weitere Informationen: `http://www.kr.tuwien.ac.at/`

1. Introduction and Overview

What are knowledge-based systems?

- In general terms, knowledge-based systems are computer systems which can solve problems *for which usually expert knowledge is required.*
- However: this characterisation is too vague.
  - E.g., an accounting software, written in a programming language like C or Java, could then also be regarded as a knowledge-based system.
  - But such systems are usually *not* knowledge-based systems!
- So, how can we properly characterise knowledge-based systems?

➥ Let us first look at some characterisations from the literature.

What are knowledge-based systems? (Ctd.)

- Nils Nilsson (1998):

  This emphasis on the importance of knowledge in applications such as these prompts us to use the phrase knowledge-based systems to describe programs that reason over extensive knowledge bases.

- Feigenbaum, McCorduck, Nii (1988):

  AI programs that achieve expert level competence in solving problems by bringing to bear a body of knowledge are called knowledge-based systems.

➥ Bottom line:

- Knowledge-based systems operate *over a knowledge base*.

What are knowledge-based systems? (Ctd.)

- Most important feature of knowledge-based systems: Separation of
  - the *representation of the knowledge* about the considered domain (given in terms of a knowledge base) and
  - the *processing of that knowledge* for performing reasoning and query answering.

- While the knowledge base contains the *domain-specific knowledge*, the knowledge processing is *domain independent*.

- The knowledge base itself is a data structure storing knowledge *directly*, using a suitable language

  $\Longrightarrow$ knowledge is represented in a declarative way ("declarative": from latin "declarativus" = explanatory, make known).

What are knowledge-based systems? (Ctd.)

- Declarative knowledge representation is in contrast to *procedural techniques*, as used in standard programming languages.
  - There, the knowledge is only implicitly represented by the actions executed by the program.

- Example: Using classical predicate logic, the following formula is a declarative description of the property of a number of being prime:

$$\mathit{prime}(n): \quad (n > 1) \land \forall m(\mathit{divide}(m, n) \rightarrow (m = 1 \lor m = n))$$

  - Here, $\mathit{divide}(m, n)$ is a predicate which is true iff m divides n.

What are knowledge-based systems? (Ctd.)

- General distinction:
  - Knowledge base = facts + explicit relations between facts
  - Knowledge processing: gain new knowledge
    - ➥ achieved by means of *inference techniques* (rules)
    - ➥ *logic* plays a central part!
- Advantages of knowledge-based systems:
  - Flexibility:
    - different tasks can be solved
  - Changeability and extensibility:
    - changes of the knowledge can be incorporated more easily
      ➥ important for corrections!
    - modular architecture
  - Visibility:
    - knowledge is “open”

Interlude: Human Experts

- If the knowledge in a knowledge-based systems is obtained from an *expert*, then such a system is referred to as an expert system.
  - Since this holds for most knowledge-based systems, often the two concepts are used synonymously.

- How can we characterise human experts?
  - Generally speaking: Experts are persons which have a specialised education and a profound practical experience about a dedicated domain.
  - Expert knowledge differs from pure special knowledge in the following aspects:
    - content,
    - scope,
    - the ability to abstract and combine pieces of information and solution steps.

Interlude: Human Experts (ctd.)

In detail:

- Experts have exceptional capabilities to solve problems in a particular area even if a problem is new or has several solutions.
- Experts draw from heuristic knowledge as well as from general knowledge.
- They often base their actions on intuitions which cannot be explained by them properly.

Interlude: Human Experts (ctd.)

- They can solve problems in the face of incomplete and uncertain information.
- Experts are rare and expensive!
- The capabilities of experts varies over time and from day to day.
- Often, a single expert is not sufficient.
- Expert knowledge can get lost (if an expert dies, for instance).
- Most importantly: expert knowledge often *cannot be passed on to other people!*

Required features of knowledge-bases systems

- Should be able to solve problems which require certain expert knowledge.
  - A system which can solve only "trivial" problems can hardly be considered to be a knowledge-based system . . .
- Separation of problem representation and processing.
  - The system should contain
    - a component for storing declarative knowledge
    - and a separate component for *accessing and processing* that knowledge.

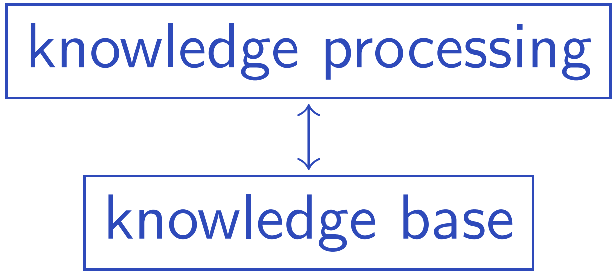
Required features of knowledge-bases systems (ctd.)

- Storage and organisation:
  - The knowledge should be stored in such a way that it can be *localised* within the system.
- Understandability:
  - The way knowledge is presented and the way solutions are found should be *understandable* for users.
- Changeability and extensibility:
  - Since knowledge is usually
    - *incomplete*,
    - often *erroneous*, and
    - *changes over time*,

    it must be possible to adapt the knowledge base.

Architecture

- As discussed, knowledge-based systems possess the following central organisation:



- The language of the knowledge base depends on the application area, but is *logic-based*.
  - Classical predicate logic is often a suitable host language
    - classical theorem provers can then be employed for the knowledge-processing part.
  - But often other formalisms are needed, because, e.g., for handling *incomplete* and/or *inconsistent* information properly.

Architecture (ctd.)

The elements of the knowledge base can be divided into the following components:

- Case-specific knowledge: The most specific form of knowledge, associated with the domain under consideration.
  - E.g., facts from observations (“evidence”).
- Rule-like knowledge: The actual core of the knowledge base, containing, e.g.,
  - Domain-specific knowledge:
    - This can comprise theoretical knowledge as well as knowledge from experience.
  - General knowledge: comprising, e.g.,
    - general problem-solving heuristics,
    - optimisation rules,
    - general relations and properties of objects of the real world.

Architecture (ctd.)

The different kinds of knowledge can occur in different amounts.

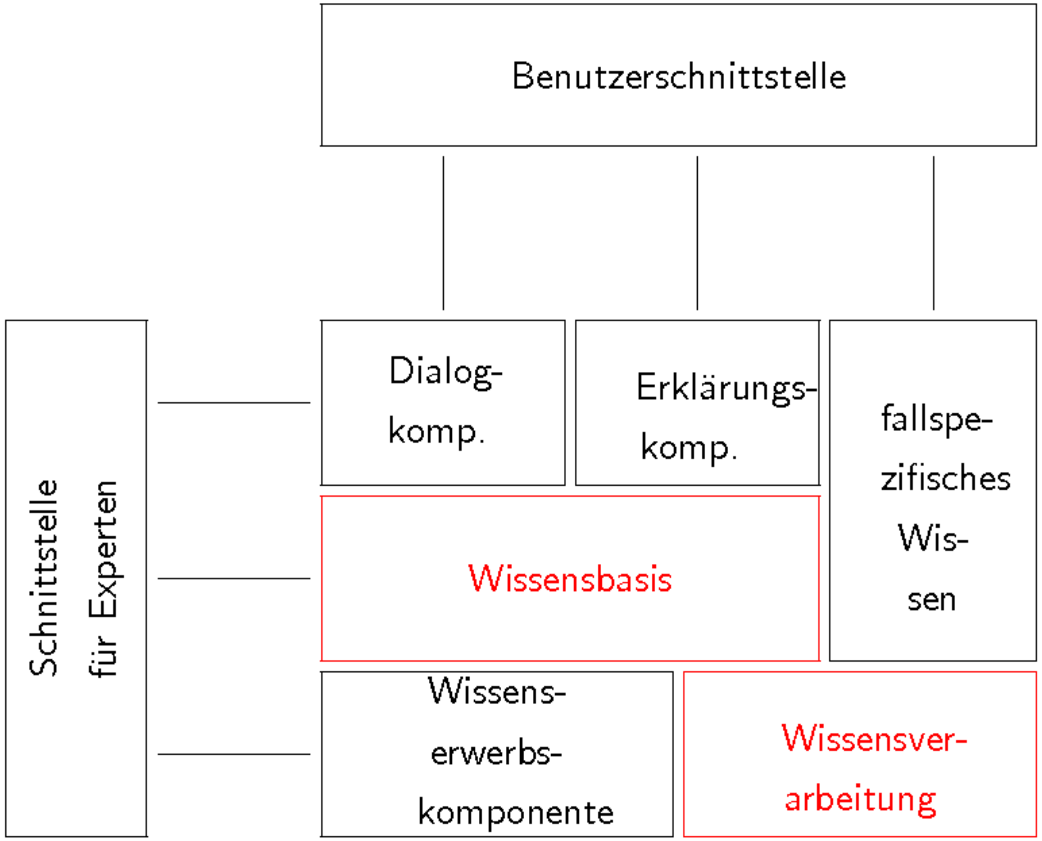
- E.g., a highly-specialised system may contain only few or no general knowledge
- while other systems may be geared towards formalising common knowledge (like the CYC project (Lenat & Guha, 1990)).

Architecture (ctd.)

In general, a knowledge-based system comprises the following components:

- the knowledge base,
- the knowledge-processing component,
- the knowledge-acquisition component, supporting *building* the knowledge base,
- the explanation component, providing explanations for generated solutions,
- the dialogue component, for communicating with the system
  - usually, there are interfaces for experts, administrating the system, and for users.

Architecture (ctd.)



Development

Complex software-engineering task, involving the following steps:

1. Problem description: define the problem to be solved and the functionality of the system.
2. Knowledge sources: choose the sources which provide the required knowledge (e.g., databases, books, human experts).
3. Design:
   - Which structures for representing the knowledge are needed?
   - Which kinds of inferences should be performed?
   - How should the explanation component and the interface work?

Development (ctd.)

4. Development tool: Depending on the specification, different tools can be used or a new one must be developed.
5. Developing a prototype: Assessing the functionality of the desired system.
6. Testing the prototype: Making sure the functionality is satisfied.
7. Refining and generalising: Realising subordinate tasks or new ones which came up during prototyping.
8. Support and administration: Eliminate errors and adapt system for new developments.

Historical remarks

- Knowledge-based systems belong to the area of artificial intelligence (AI).
  - The origin of the term AI goes back to a workshop at Dartmouth College during the summer of 1956.
    - ☞ The workshop was organised by John McCarthy and proposed jointly with Marvin Minsky, Nathaniel Rochester, and Claude Shannon.
- AI research in the Sixties focused on general problem-solving approaches with few or no information about the considered domain ("general problem solver", Newell & Simon, 1963).
- However, these systems performed poorly on complex examples.
  - It was realised that more knowledge about the domain should be used, enabling larger derivation steps of subproblems.
  - ➥ This marks the starting point of knowledge-based systems, which were built for
    - *specific tasks* and *supporting* a human user.

Historical remarks – Pioneering systems

DENDRAL (Buchanan, Sutherland, Feigenbaum, 1969)

- System for determining the structure of molecules from mass spectra, developed at Stanford.
- In chemistry, a usual method is to seek for known patterns of peaks in a mass spectrum to identify common sub-structures.
- Contains large number of special-purpose rules like the following for detecting ketone subgroups:

  IF there are two peaks at x_1 and x_2 such that
  (a) $x_1 + x_2 = m + 28$ (m is the mass of the molecule)
  (b) $x_1 - 28$ is a high peak
  (c) $x_2 - 28$ is a high peak
  (d) at least one of x_1 and x_2 is high
  THEN there is a ketone subgroup

Historical remarks – Pioneering systems (ctd.)

MYCIN (Feigenbaum, Buchanan, Shortliffe, 1973)

- After the experience with DENDRAL, Feigenbaum and others began the *Heuristic Programming Project* (HPP) to look for other areas of the new methodology.
- MYCIN was the next step, developed to *diagnose* blood infections and to suggest suitable *treatments*.
- The system was able to perform as well as experts and considerably better than junior doctors.
- In contrast to DENDRAL, it contained
  - no general theoretical model
  - and the rules had to reflect the *uncertainty* associated with medical knowledge.
- The knowledge modeled by MYCIN was obtained by interviewing experts, containing about 450 rules.

Historical remarks – Pioneering systems (ctd.)

- MYCIN incorporated a new calculus for uncertainty reasoning, called certainty factors.
- A typical rule is, e.g., the following:

  IF (a) the infection is primary-bacteremia, and
  (b) the site of the culture is one of the sterile sites, and
  (c) 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

- The heart of the system was an interactive decision-supporting dialogue system
  - it responded to questions and posed questions in turn.
- Despite its performance it was not widely used
  - mainly due to a lack of trust to an anonymous computer system in the sensible area of medicine.

Historical remarks – Pioneering systems (ctd.)

PROSPECTOR (Duda, Hart, Nilsson, et al., 1974-1983)

- A probabilistic system for locating mineral deposits, based on Bayes' Theorem, again developed in Stanford.
- It reached fame because it correctly predicted a rich source of molybdenum near Mount Tolman in eastern Washington (1980).
- Remark:
  - Peter E. Hart is founder, chairman, and president of Ricoh Innovations, and together with Nils J. Nilsson co-inventor of the A^* search algorithm.

Historical remarks – Pioneering systems (ctd.)

R1 (McDermott, 1982)

- The first commercially successful expert system, developed in cooperation with Digital Equipment Corporation (DEC).
- Served for configurating VAX computers, containing more than thousand rules.
- Saved several millions of Dollars.
- R1 was later renamed into XCON.
- As an aside, DEC was acquired by Compaq in 1998, which in turn merged with Hewlett-Packard in 2002.

↪ Today several thousands of expert systems are in use, employed, e.g., for medical applications, technical systems, financing . . .