Wissensbasierte Systeme

Julian Schubert

14. Mai 2021

Inhaltsverzeichnis

1 Einführung

1.1 Was sind Wissensbasierte Systeme?

- **Ziel**
  - Lösen eines Problems durch Wissen und Inferenz
- **Unterschied zu Neuronalen Netzen**
  - Lösung erklärbar und kritisierbar
  - Aufwändiger Wissenserwerbsprozess
- **Arten von Wissen**
  - Fakten, Wahrscheinlichkeiten
  - Relationen, REgeln, Constraints
  - Muster, Fälle + Ähnlichkeitsmaß
- **Wissenserwerb**
  - durch Fachexperten
  - durch Lernen aus Fällen
  - durch Extraktion aus Literatur

1.2 Zentrale Aufgaben

Wissensrepräsentation festlegen

- Basiert häufig auf einer Befragung von Fachexperten
  - Für Fachexperten natürlich
  - Präzise zur Herleitung von Schlussfolgerungen
  - Effizient verarbeitbar

Wissen aquirieren
Editor zur Eingabe von Wissen wird benötigt

- Geringe Einarbeitungszeit
- Natürliche Darstellung
- Effiziente Wissenseingabe
- Übersichtlich auch für große Wissensbasen
- Sollte eine Schnittstelle zum Testen des Wissens bieten

Wissen verarbeiten (Reasoning)
Evaluation mit Fällen

1.3 Interaktive und eingebettete WBS

- **Interaktiv**
  - WBS berät Nuzter in geführtem Dialog
  - WBS präsentiert Lösung(en) mit vorheriger Dateneingabe
  - WBS unterstützt Exploration des Lösungsraums
- **Embedded**
  - WBS präsentiert Lösung ohne Dateneingabe
  - WBS gibt Hinweise (Alerts), falls notwendig (z.B. Kritik)
  - WBS handelt autonom

1.4 Lebenszyklus eines WBS

- **Bedarf feststellen:** Ist-Zustand, Ziele
- **Entwickeln:** Methoden, Phasen
- **Bereitstellen:** z.B. Server, Integration in anderes System
- **Nutzen:** GUI, autonom
- **Evaluieren:** Korrektheit, Zeitersparnis, Dokumentation
- **Evolvieren:** Lernen, Weiterentwickeln

1.5 Domain Specific Languages (DSL)

Abgrenzung zu WBS, trotz ähnlicher Zielsetzung: Formale Sprache Programmierung und Wissenformalisierung in einer eingeschränkten Domäne für Domänenspezialisten

1.5.1 Interne DSL

Untermenge einer generellen Sprache, z.B. UML-Profile, domänenspezifische XML-Schemata

1.5.2 Externe DSL

Neu definiert, z.B. SQL, reguläre Ausdrücke

1.6 Wissenserwerb

1.6.1 Heuristische Entscheidungsbäume

Man benutzt nicht einen großen Entscheidungsbaum (ein mal falsch abbiegen und man kommt eventuell nicht mehr auf die richtige Lösung), sondern mehrere kleine Bäume

1.6.2 Überdeckender Diagnose-Score

Tabelle mit Baum und Merkmale, Einträge: Wie Wahrscheinlich ist das Attribut für den Baum

1.6.3 Modellbasierter Wissenserwerb

Modell erstellen und daran lösung erarbeiten

1.6.4 Fallorentierter Wissenserwerb

An vorherigen Fällen orientieren

2 Basiswissensrepräsentationen

- **Regeln:**
  - Gerichtete Beziehungen zwischen Konzepten
- **Constraints**
  - Ungerichtete Beziehunen zwischen Konzepten
- **Konzepte (Klassen, Instanzen):**
  - Haben Eigenschaften
  - Eigenschaften können auch Relationen zwischen Konzepten sein

2.1 Regeln

Wenn <Vorbedingung> **dann** Aktion
Zwei Aktionstypen:

- **Implikation** (Deduktionen): Kommutativ
- **Handlung** nicht kommutatitv

2.1.1 Aufbau eines Regelinterpreters

Programmierer legt fest, was getan wird, die Reihenfolge bestimmt der Regelinterpreter
Komponenten

- Datenbasis
- Regelbasis

Abarbeitungsstrategien

- **Vorwärtsverkettung (Forward Reasoning)**: Eine Regel, deren Vorbedingung durch die Datenbasis erfüllt ist, wird ausgewählt und deren Aktion ausgeführt. Dadruch ändert sich die Datenbasis und der Zyklus wiederholt sich, bis keine Regel mehr answendbar oder ein Abbruchkriterium erreicht ist
- **Rückwärtsverkettung (Backward Reasoning)**: Ausgehend von einem Ziel werden nur die Regeln überprüft, deren Aktionsteil das Ziel enthält. Falls Parameter der Vorbedinugn unbekannt sind, werden sie mit anderen Regeln hergeleitet oder vom Benutzer erfragt

Ripple Down Rules (RDR):
Zu jeder Regel können Ausnahmen (except) geschrieben werden, um ein inkrementelles Erstellen einer Regelmenge zu ermöglichen.

- **except:** es wird eine Ausnahme zu der Regel geprüft: Wenn die Regel gilt und die Ausnahme nicht zutrifft, dann wird der Aktionsteil der Regel nicht ausgeführt, sonst die Ausnahme
- **else:** Es wird mit einer anderen Regel weitergearbeitet, wenn die Regel nicht gilt