

Kognitive Modellierung animierter Agenten

BDI-Architektur
(Beliefs – Desires – Intentions)

Frank Schönmann

SS 2003

Überblick

- Einführung/Motivation
- Entwicklung von BDI
- Aufbau eines BDI-Agenten
- Ablauf des *practical reasoning*
- Bewertung von BDI-Systemen
- Zusammenfassung

Deliberative vs. reaktive Agenten

- reaktives Verhalten: folgt unmittelbar auf Sensoreneingaben
- deliberatives Verhalten: es wird eine Modellierung der Welt durchgeführt und anschließend ein optimales Verhalten durch Planen und Schlussfolgern ermittelt

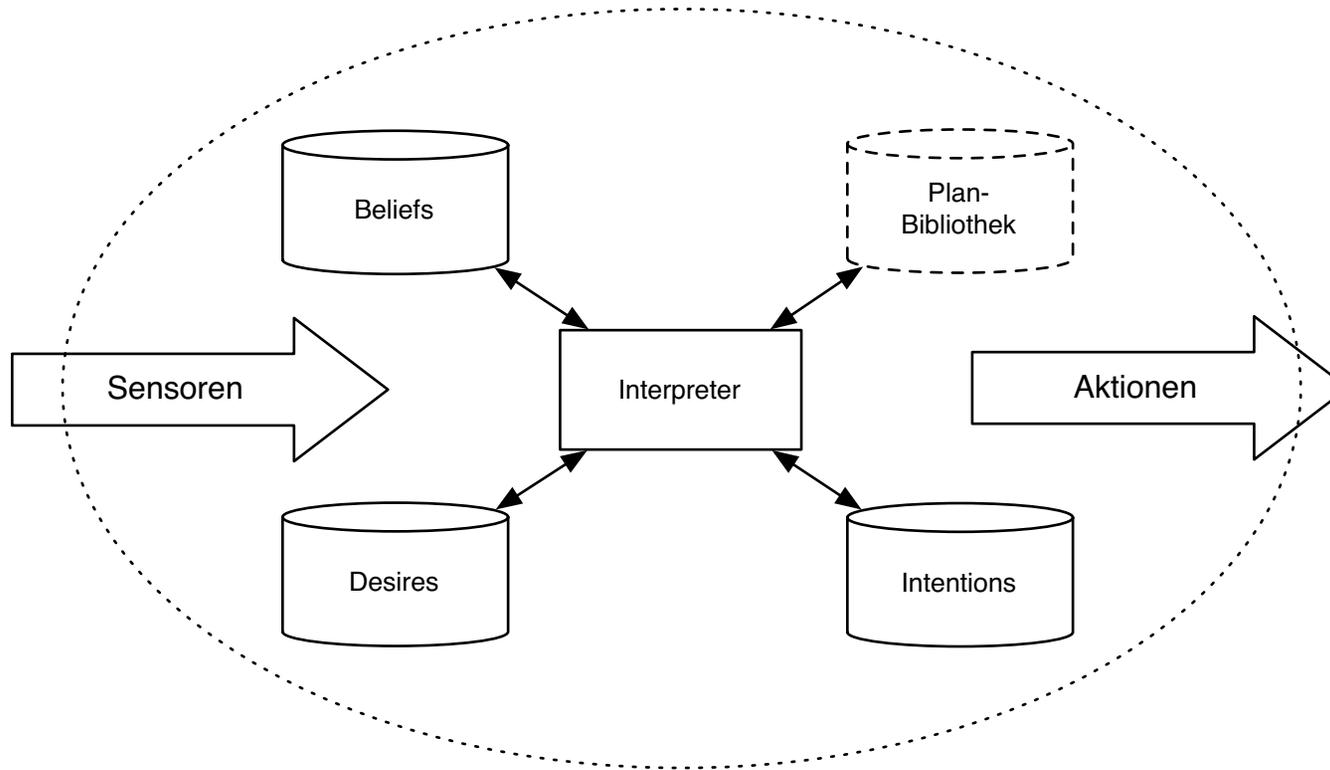
Practical Reasoning

- Prozess der Entscheidungsfindung: welche Aktion soll in jedem Moment durchgeführt werden, um gewünschten Zielen näherzukommen?
- zusammengesetzt aus zwei Teilen:
 - *deliberation*: **welche** Ziele sollen erreicht werden?
 - *means-end reasoning*: **wie** werden die Ziele erreicht?

BDI-Hintergrund

- basiert auf philosophischem Verständnis des *practical reasoning* beim Menschen, insbesondere den Einfluss von Intentionen bei der Entscheidungsfindung (Theorie entwickelt von Michael E. Bratman, 1987)
- ⇒ Agent hat *mental attitudes*: Weltwissen (*beliefs*), Wünsche/Ziele (*desires*) und aktuelle Absichten (*intentions*)
- erste Implementierungen: IRMA (Bratman), PRS (Georgeff & Lansky)

Allgemeiner Aufbau eines BDI-Agenten



Allgemeiner Aufbau eines BDI-Agenten

- Input: Wahrnehmung durch Sensoren
- Output: Aktionen zur Beeinflussung der Umwelt
- Datenstrukturen: *beliefs*, *desires*, *intentions*, Pläne
- Funktionen: Interpreter zur Verwaltung der Daten, zusammengesetzt aus Teilfunktionen

Beliefs

- Fakten über aktuellen Zustand der Umgebung, den internen Zustand des Agenten, Nachrichten von anderen Agenten, Hintergrundwissen
- wird wie bei anderen KI-Systemen auch explizit in einer Wissensdatenbank (*knowledge base*) gespeichert, z. B. in PL1
- kontinuierliche Aktualisierung, beeinflusst von Wahrnehmung (Kameras, Mikrofone, ...) und internen Schlussfolgerungen

Desires

- (Haupt-)Ziele des Agenten: erwünschte Weltzustände
- Zielorientierung erlaubt Wiederaufnahme fehlgeschlagener Aktionen
- Mehrere verschiedene Ziele, die angestrebt werden, können widersprüchlich sein!

Plans

- Abarbeitungsvorschriften, um einen bestimmten Teilzustand der Welt in einen anderen zu überführen
- Bestandteile:
 - Vorbedingung (*pre-condition*): Fakten, die erfüllt sein müssen, um einen Plan auszuführen
 - Nachbedingung (*post-condition*): erreichter Zustand, nachdem dieser Plan erfolgreich abgearbeitet wurde
 - evtl. *context condition*: Bedingung während der Ausführung
 - Anweisungen (*body*): direkt ausführbare Aktionen oder Teilziele

Beispiel-Plan (JAM)

```
Plan: {
NAME: "Stack blocks that are already clear"
GOAL:
    ACHIEVE ON $OBJ1 $OBJ2;
CONTEXT:
BODY:
    EXECUTE print "Making sure " $OBJ1 " is clear (ACHIEVEing it).\n";
    ACHIEVE CLEAR $OBJ1;
    EXECUTE print "Making sure " $OBJ2 " is clear (ACHIEVEing it).\n";
    ACHIEVE CLEAR $OBJ2;
    EXECUTE print "Both objects clear. Moving " $OBJ1 " on top of " $OBJ2 " .\n";
    PERFORM move $OBJ1 $OBJ2;
UTILITY: 10;
FAILURE:
    EXECUTE print "\n\nStack blocks failed!\n\n";
}
```

Intentions

- ausgewählte Pläne, zu deren Ausführung sich der Agent verpflichtet, d. h. der Agent legt sich auf Intentionen fest, bis diese erfüllt oder nicht mehr sinnvoll bzw. erfüllbar sind (*commitment*)
- spielen entscheidende Rolle im *practical reasoning*-Prozess durch Festlegung auf ein gewünschtes Ziel (\Rightarrow *deliberation*)
- führen zu direkten Aktionen oder weiterer rekursiver Auswahl von Plänen (\Rightarrow *means-end reasoning*)
- z. B. als *intention stack* implementiert

Eigenschaften der Intentionen

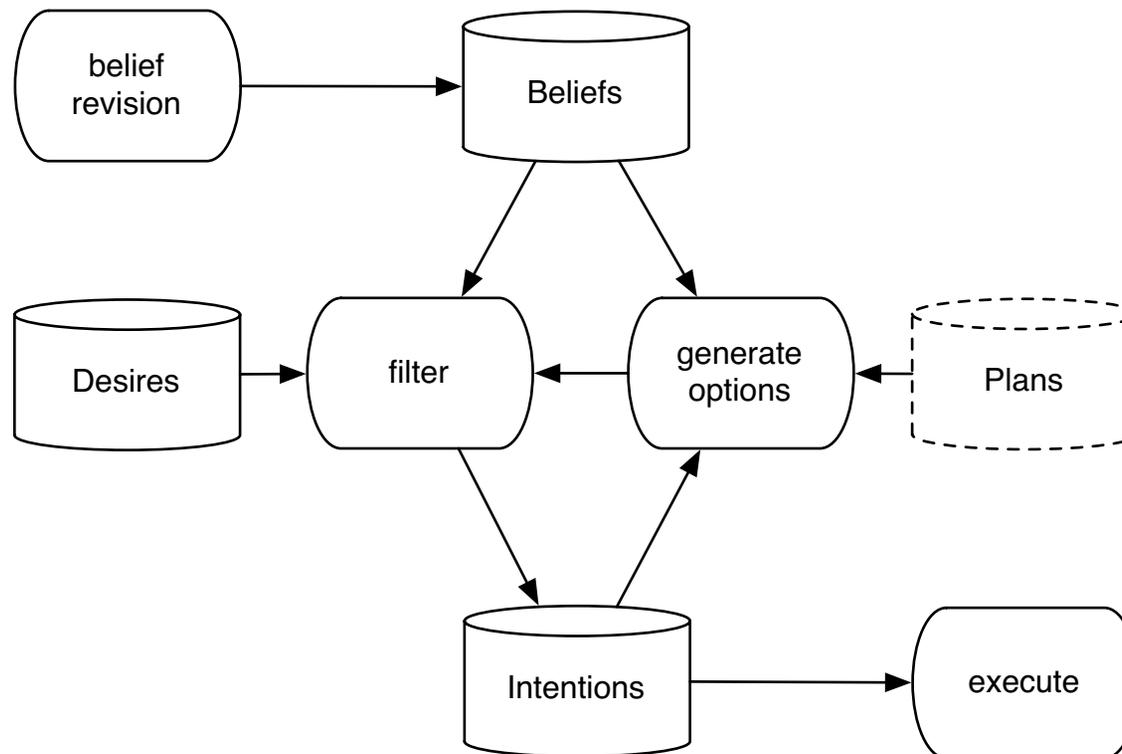
- beeinflussen *means-end reasoning*: Durchführung von Aktionen, um das gewählte Ziel zu erreichen
- beschränken weitere Planung: keine Auswahl von kontraproduktiven Plänen bzw. Zielen
- bleiben bestehen solange es sinnvoll ist: Ziel erreicht oder nicht mehr sinnvoll
- können *beliefs* über die Zukunft beeinflussen: Schlussfolgerungen auf der Annahme, dass das Ziel erreicht wird

Desires vs. Intentions

- Unterscheidung durch verschiedene Eigenschaften:

<i>intentions</i>	<i>desires</i>
<i>commitment</i>	nicht jedes vorhandene Ziel muss erreicht werden
keine Widersprüche	Inkonsistenzen möglich

Ablauf des practical reasoning



Ablauf des practical reasoning

- *belief revision*: Aktualisierung der Wissensbasis, abhängig vom Input und internen Schlussfolgerungen
- *generate options*: alle aktuell möglichen Optionen werden ermittelt
- *filter*: aus möglicherweise konkurrierenden Zielen werden günstigste Möglichkeiten herausgesucht und zu Intentionen hinzugefügt
- *execute/action selection*: direkt ausführbare Intentionen führen zu einer Aktion des Agenten

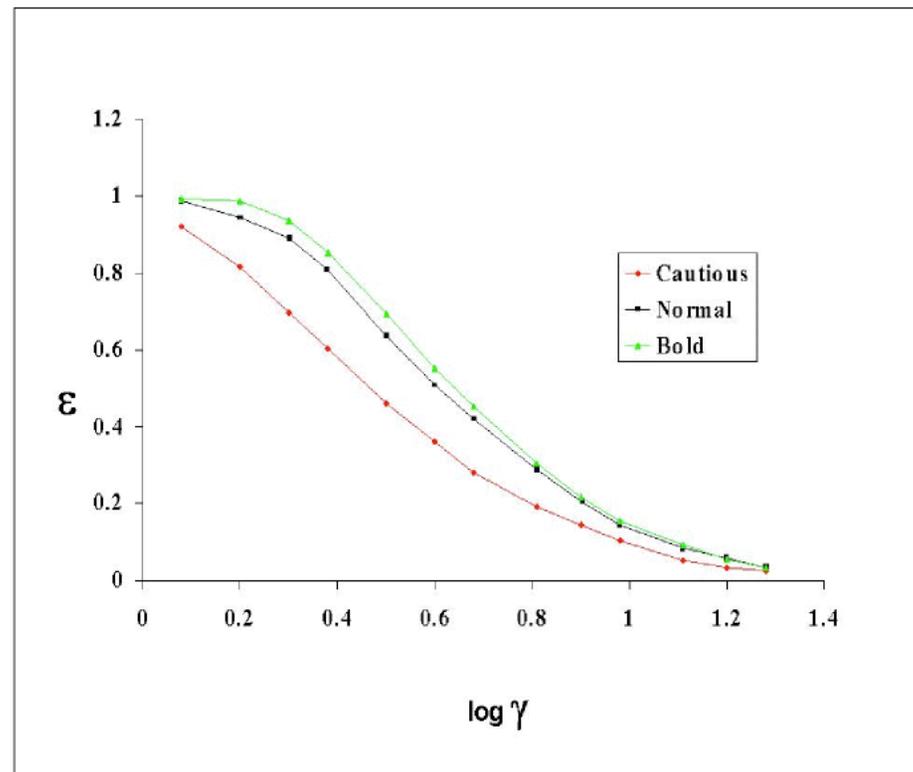
Problem: Verwaltung der Intentionen

- Abwägung nötig zwischen:
 - ständige Überprüfung der Intentionen: verschwendet Zeit und andere Ressourcen
 - seltene Überprüfung der Intentionen: Verfolgung von Zielen, die nicht mehr erreicht werden können

Problem: Verwaltung der Intentionen

- Verschiedene Strategien (David Kinny & Michael Georgeff):
 - *bold*: Überprüfung niemals oder nur in kritischen Situationen
 - *cautious*: ständige Überprüfung der Intentionen
- Erfolg abhängig von der Veränderungsrate der Welt, daher ist eine von der Einsatzumgebung abhängige Einstellung nötig

Vergleich von Strategien



⇒ *commitment* führt zu Effizienzsteigerung, *bold* nicht immer ideal

Vergleich mit SOAR

- SOAR und BDI scheinen kompatibel:

BDI	SOAR
<i>beliefs</i>	aktueller Zustand
<i>desires</i>	Ziele
<i>plans</i>	Operatoren
<i>intentions</i>	ausgewählte Operatoren
<i>commitment</i> -Strategie	Strategien für Operator-Terminierung

- SOARs Lernverfahren und *truth maintenance* fehlen in BDI

Vorteile

- intuitiv, da BDI auf den Prinzipien der Entscheidungsfindung beim Menschen basiert
- bestehende Implementationen und Systeme: IRMA, PRS, dMARS, ...
- mehrere erfolgreiche Anwendungen, u. a. bei der Fehlerdiagnose für das Space Shuttle

Kritikpunkte

- ursprüngliches BDI-Modell nicht geeignet für Systeme, die lernen und ihr Verhalten anpassen müssen
- nicht explizit ausgelegt auf Zusammenspiel mehrerer Agenten

⇒ Erweiterungen möglich

Zusammenfassung

- Ziel: Agenten mit deliberativem Verhalten modellieren
- philosophischer Hintergrund: *mental attitudes*
- *desires/intentions* bringen Ordnung in verfügbare Pläne durch Filterung der Optionen (*deliberation*) und hierarchische Generierung von Intentionen (*means-end reasoning*)