

Mehrschichtige und dezentrale Entscheidungsprozesse in Agentensystemen

Gruppe 3: H. Stadler, M. Betz, P. Heger und B. Wladasch

Fachpraktikum Künstliche Intelligenz: Multiagentenprogrammierung
Artificial Intelligence Group,
University of Hagen, Germany

30. September 2022



Entwicklung und Bewertung unterschiedlicher Entscheidungsprozesse von Agentensystemen im Kontext des Multi-Agent Programming Contest 2022

- ▶ 2 Varianten basierend auf der BDI-Architektur mit unterschiedlich stark dezentralisierten Entscheidungsprozessen
 - ▶ Agent V1
 - ▶ Agent V2
- ▶ Leistungsfähigkeit beider Varianten zu bewerten
- ▶ Verschiedene Lösungsansätze zu erhalten, die zwischen den Systemen ausgetauscht werden können

1 Technische Umsetzung

2 Agentensystem V1

3 Agentensystem V2

4 Turniere

5 Rekapitulation und Ausblick

- ▶ Programmiersprache Java Version 17
 - ▶ Wunsch nach umfangreichen Werkzeugen und Bibliotheken zur Verifikation und Problemfindung
- ▶ Beide Agentensysteme basieren auf *javaagents* Gerüst der MASSim (*Multi-Agent Systems Simulation Platform*)
- ▶ Architekturkonzepte mittels UML-Diagrammen entwickelt
- ▶ Projektmanagement über Github-Arbeitsablauf
 - ▶ Arbeitsaufteilung über Aufgaben- und Fehlermanagement
 - ▶ *Feature-Banches* mit Quellcodeüberprüfung durch weiteres Gruppenmitglied

- 1 Technische Umsetzung
- 2 Agentensystem V1**
- 3 Agentensystem V2
- 4 Turniere
- 5 Rekapitulation und Ausblick

Ziele

- ▶ Mehrschichtiger Entscheidungsprozess der das BDI-Konzept erweitert
- ▶ Aufbau einer umfangreichen Wissensbasis
- ▶ Integration der Wegfindung in den Entscheidungsprozess
- ▶ Strategien zur Verifikation und Problemfindung des dynamischen Systems entwickeln

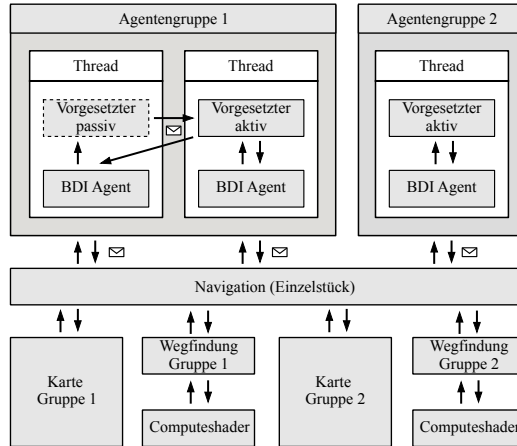


Abbildung: Architektur Agentensystem V1

- 1 Technische Umsetzung
- 2 Agentensystem V1
- 3 Agentensystem V2**
- 4 Turniere
- 5 Rekapitulation und Ausblick

Struktur

- ▶ Der AgentV2 arbeitet mit der Step-Methode
- ▶ Desires mit und ohne Task-Bezug

ohne Task

LocalExploreDesire
GoAdoptRoleDesire
ExploreMapSizeDesire

mit Task

GoAbandonedBlockDesire
GoDispenserDesire
GoGoalZoneDesire
SubmitDesire

MehrBlockTask

MasterMultiBlocksDesire
HelperMultiBlocksDesire
Helper2MultiBlocksDesire
ConnectMultiBlocksDesire

Wie finden die Agenten ihre Desires?

- ▶ In jedem Step werden alle Desires auf Ausführbarkeit geprüft
- ▶ Alle ausführbaren Desires bekommen dynamisch eine Priorität vergeben
- ▶ Das Desire mit der höchsten Priorität wird zur Intention
- ▶ Aus der Intention wird die nächste Aktion des Agenten abgeleitet

Wie arbeiten die Agenten zusammen?

- ▶ Bildung von Supervisor-Gruppen bei jedem Treffen fremder Agenten
- ▶ Bildung von dynamischen Adhoc-Kooperationen innerhalb der Supervisor-Gruppen zur Bearbeitung einer Task
- ▶ Keine zentrale Koordination der Agenten
- ▶ Steuern der Art und Anzahl der Adhoc-Kooperationen über Setup-Variablen
- ▶ Nutzung der Adhoc-Kooperationen auch zur Ermittlung der Mapgröße

- 1 Technische Umsetzung
- 2 Agentensystem V1
- 3 Agentensystem V2
- 4 Turniere**
- 5 Rekapitulation und Ausblick

- 1 Technische Umsetzung
- 2 Agentensystem V1
- 3 Agentensystem V2
- 4 Turniere
- 5** Rekapitulation und Ausblick

Block

This is a regular block.

- ▶ This is an item in a block.

Block

This is an alert block.

- ▶ This is an item in an alert block.

Block

This is an example block.

- ▶ This is an item in an example block.

We can highlight text. There are also a darker option, and a yellow option.

The same options are also available in math mode: $a + b = c$

6 Appendix

This is an appendix.

- ▶ Page numbers start from the beginning.