

Gruppe 3: Catchphrase?

$\dots^1, \dots^1, \dots^1$ und \dots^1

FernUniversität in Hagen, Universitätsstraße 47, 58097 Hagen, Deutschland
{...}@studium.fernuni-hagen.de
<https://www.fernuni-hagen.de>

1 Einleitung

TODO Gruppenbeschreibung

2 Kommunikation und Kommunikationsmittel

TODO Gruppentreffen, Github

3 Technische Rahmenbedingungen und Softwarebasisarchitektur

TODO Java, Bdi, 2 Agentensysteme, UML

4 Gruppenbeitrag Heinz Stadler

Nach einer ausführlichen Einarbeitung in die Thematik der Multiagentensysteme, erfolgte die Analyse der Ergebnisse des 15. Multi-Agent Programming Contest [1]. Diese ergab, dass neben der Entscheidungsfindung der Agenten, der Aufbau einer konsistenten und umfangreichen Wissensbasis (vgl. [1, S. 29]) sowie die effiziente Problemfindung (vgl. [1, S. 17]), eine wesentliche Schwierigkeit der Aufgabe bilden.

Als Folge dessen entschied sich der Autor mit dem Aufbau der Wissensverwaltung (siehe 4.2) zu beginnen. Diese umfasst eine Datenstruktur zur Speicherung der von der Simulation übermittelten Informationen, sowie eine Lösung zum Aufbau einer globalen Karte des Simulationsgebiets. Nach Fertigstellung der Karte und deren funktionalen Verifikation, wurde parallel an der Konzeptionierung des Agentensystem V1 (siehe 4.1) sowie der Implementierung einer intelligenten Wegführung (siehe 4.3) gearbeitet. Um die Ergebnisse der Wegführung und das Entscheidungsverhalten der Agenten effektiv verifizieren zu können, folgte die Erstellung eines graphischen Analysewerkzeugs (siehe 4.5) das im weiten Entwicklungsverlauf stetig weiterentwickelt wurde und sich als sehr hilfreich bei der Fehlerfindung zeigte.

4.1 Agent V1 - Architektur

Das Agentensystem V1 wurde federführend vom Autor konzipiert, in der Gruppe abgestimmt und schließlich selbstständig implementiert. Es erweitert das BDI-Konzept [5] um zusätzliche Daten-, Berechnungs- und Entscheidungsebenen, die in Abbildung 1 illustriert sind.

Das System kombiniert das Konzept des BDI-Agenten mit dem Konzept einer bidirektionalen vertikalen Schichtarchitektur [3, S. 61-62], auf die in Abschnitt 4.4 näher eingegangen wird.

Jeder BDI-Agent wurde mit einer Vorgesetzteninstanz kombiniert und in einem Thread gekapselt.

Die Kommunikation zwischen Agenten erfolgt über Nachrichten. Agenten können zu Agentengruppen zusammengefasst werden, welche die im Simulationsverlauf erhaltenen Umgebungsinformationen in einer gemeinsamen Karte verwalten. Die Karte und die Module zur Wegführung werden von einem zentralem Navigationsmodul, das als Einzelstück ausgeführt wurde, verwaltet.

Dies ermöglicht die parallelisierte Entscheidungsfindung und unterstreicht die konzeptionelle Eigenständigkeit der Agenten untereinander. Agenten können zu Agentengruppen zusammengefasst werden. In jeder Agentengruppe bleibt der Vorgesetzte des Agenten mit der niedrigsten Index aktiv und übernimmt die Vorgesetztenrolle der anderen Agenten in der Gruppe. Die

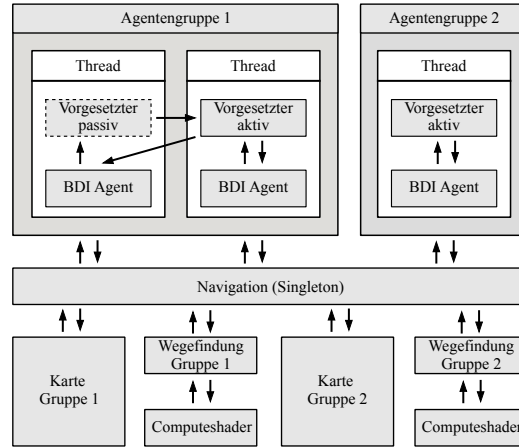


Abb. 1. Dummy figure.

4.2 Wissensverwaltung

Belief, Karten

Die zur Kommunikation mit dem Simulationsserver gewählte Bibliothek EISMASSim [4] ermöglicht die Abfrage der aktuellen Simulationsinformationen in der Form von Percepts. Da Percepts eine universelle Datenstruktur, unabhängig vom jeweiligen Dateninhalt ist, wurde die Java Klasse Belief erstellt. In dieser werden die aktuellen Percepts in einfach zu verarbeitende Java Datenstrukturen umgewandelt.

Zusätzlich werden die aktuellen Informationen aus dem Sichtfeld der Agenten in eine weitere Datenstruktur GameMap, im Folgenden als Karte bezeichnet, gespeichert. Die Karte hat zum Ziel umfangreiches Wissen über die Umgebung der Agenten zu sammeln. Sie besteht aus einzelnen Zellen, die chronologisch Informationen der Agenten persistiert. Die Karte ermöglicht Informationen von Agenten, die nachweislich inkorrekte Daten gesendet haben, zu entfernen. Des

Weiteren können verschiedene Karten zusammengeführt werden, wobei die Chronologie der enthaltenen Daten gewahrt wird.

Da die absolute Position der Agenten zum Simulationsbeginn nicht bekannt ist, stellt das Zusammenführen einzelner Karten eine Schwierigkeit dar. Im Nachfolgenden wird

4.3 Wegfindung

Pathfinding

4.4 Ziel- und Absichtsfindung

Desires

4.5 Verifikation und Problemfindung

Tests / Debugger

4.6 ...

5 Gruppenbeitrag Melinda Betz

6 Gruppenbeitrag Phil Heger

7 Gruppenbeitrag Björn Wladasch

8 Turniere

Turnier 2

Turnier 3

Turnier 4

Turnier 5

Turnier 6

9 Rekapitulation und Ausblick

Vor- und Nachteile der Entscheidung von zwei Architekturen Was sollte noch verbessert werden Wie sind wir zufrieden

Literatur

1. Ahlbrecht, T., Dix, J., Fiekas, N. und T. Krausburg: The Multi-Agent Programming Contest 2021, Springer, Heidelberg, 2021
2. Hart, P. E., Nilsson, N. J. und Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths, in IEEE Transactions on Systems Science and Cybernetics, 4. Auflage, Nummer 2, Seiten 100-107, Juli 1968
3. Weiss, G.: Multiagent Systems, 2. Auflage, The MIT Press, Cambridge, 2000
4. github.com/agentcontest/massim_2022, [agentcontest/massim_2022](https://github.com/agentcontest/massim_2022/blob/main/docs/eismassim.md), https://github.com/agentcontest/massim_2022/blob/main/docs/eismassim.md, EISMASSim Documentation, 21.08.2022
5. Bratman, M.: Intention, plans, and practical reason, Harvard University Press, Cambridge, 1987