

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 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 4 illustriert sind.

Das System kombiniert den Aufbau eines BDI-Agenten (vgl. [3, S. 58]) mit einer bidirektionalen vertikalen Schichtarchitektur [3, S. 61-62], auf die in Abschnitt 4.4 näher eingegangen wird.

Jeder Agent wurde mit einer Vorgesetzteninstanz, die eine zusätzliche Entscheidungsebene bildet, kombiniert und in einem Thread parallelisiert. Werden Agenten zu einer Gruppe zusammengeführt, bleibt eine Vorgesetzteninstanz aktiv. Die sonstigen Instanzen der Gruppe werden passiv und übernehmen im Weiteren nur noch die Weiterleitung von Nachrichten an die aktive Entität.

Die Kommunikation zwischen Instanzen in verschiedenen Threads erfolgt über Nachrichten, die in einer threadsicheren Warteschlange zwischengespeichert werden. Die Verständigung des Agenten mit seinem direkten Vorgesetzten wird mittels Methodenaufrufen realisiert.

Agentengruppen aktualisieren eine gemeinsame Karte mit im Simulationsverlauf erhaltenen Umgebungsinformationen. Diese und Module zur Wegführung werden von einem zentralen threadsicheren Navigationsmodul, das als Einzelstück ausgeführt wurde, verwaltet.

4.2 Wissensverwaltung

Jeder Agent hat Zugriff auf eine individuelle Wissensbasis (Beliefs), die von der Simulation bereitgestellte Informationen auswertet und speichert.

Die enthaltenen Umgebungsdaten, beschränken sich auf das aktuelle Sichtfeld des Agenten. Es ist weder die Größe des Simulationsgebiets, noch die absolute Position des Agenten in diesem bekannt. Um aus den partiellen Umgebungsinformationen eine globale Sicht zu erhalten, werden diese an das Navigationsmodul weitergeleitet und in einer chronologisch fortgeschriebenen Karte zusammengeführt.

Beim Simulationsstart erhält jeder Agent, eine Karte mit festgelegter Initialgröße (Abb. 2 Punkt 1), die beim Erkunden der Umgebung erweitert wird (Abb. 2 Punkt 2). Treffen sich zwei Agenten aus unterschiedlichen Gruppen, wird dies vom Navigationsmodul erkannt und an die aktiven Vorgesetzten beider Gruppen gemeldet. Stimmen beide Vorgesetzte einer Vereinigung zu, werden deren Karten

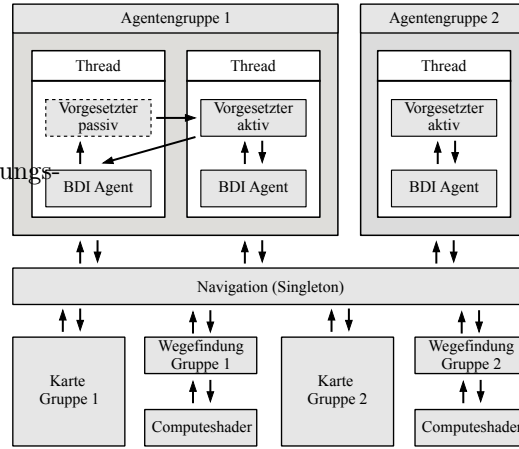


Abb. 1. Architekturdiagramm Agent V1

überlagert und schließlich zusammengeführt (Abb. 2 Punkt 3). Die entstandene Karte ermöglicht nun im weiteren Simulationsverlauf die aktuelle Position aller Agenten einer Gruppe untereinander zu bestimmen. Aus dieser Information kann nun durch jeweils zwei Agenten, die sich in entgegengesetzte Richtungen bewegen und sich durch das kontinuierliche Simulationsgebiet zwangsläufig wieder treffen, die Kartengröße ermittelt werden. Ist dies erfolgt, wird die Karte beschnitten, wobei die Informationen abgeschnittener Bereiche auf der gegenüberliegenden Seite eingefügt werden und somit nicht verloren gehen (Abb. 2 Punkt 4).

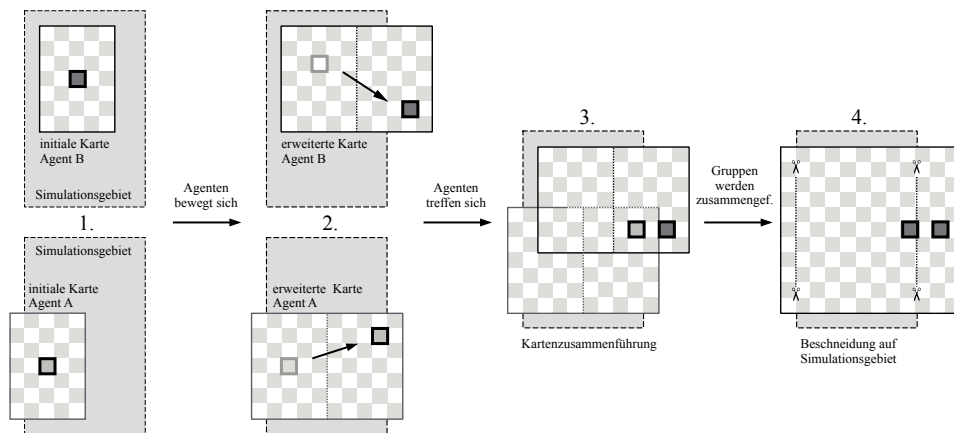


Abb. 2. Erweiterung und Zusammenführung einzelner Karten

4.3 Wegfindung

Die Pathfinding, 2 Stufiges System

4.4 Ziel- und Absichtsfindung

Desires, verticale Schichtarchitektur

4.5 Verifikation und Problemfindung

Tests / Debugger

4.6 Rekapitulation

Zahlen, sonstiges Server Bug, Turnier gehalten, Github erstellt

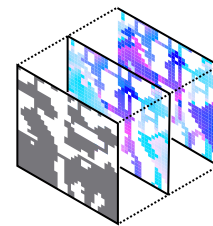


Abb. 3. Datenstruktur Wegfindung

4.7 ...

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, 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

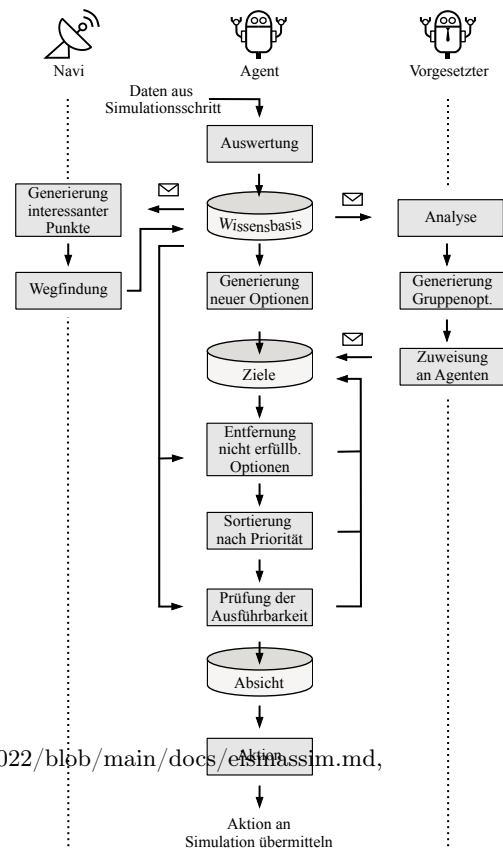


Abb. 4. Architekturdiagramm Agent V1