

Generische Agenten

Entwurf einer Architektur für den
Einsatz in Multi-Agenten-Simulationen

Agenda

- Vorhaben
- Agenten
 - Allgemeine Funktionsweise
 - Klassische Agentenarchitekturen
 - Paper zu generischen Ansätzen
- Ausblick
- Konferenzen
- Personen
- Fragen und Diskussion

Ziel / Vision

- Motivation
- Anforderungen
- Anwendungsfälle
 - Gemeinsamkeiten und Unterschiede

Motivation

- KI: weites, aktuelles und (für mich) interessantes Feld
- Ziel: Entwurf einer generischen Softwareagentenarchitektur
 - "Nicht immer wieder das Rad neu erfinden"
- Einsatz in dem MARS-Framework

Anforderungen

- MAS → Skalierbarkeit und Performanz wichtig
- generischer Ansatz
 - Grundlage für domänenspezifische Implementation
 - muß konfigurierbar sein (DSL)
 - darf nicht einschränkend wirken
 - → Kompromiss zwischen abstrakt und verwendbar

Anwendungsfälle

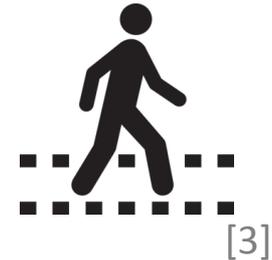
- Abdoulaye-Wald
 - Naturschutzgebiet in Zentral-Togo
- Agententypen:
 - Bäume
 - Bauern
 - Nomaden
 - Regierung



[4]

Anwendungsfälle

- WALK
 - Fußgänger
- Räuber-Beute-Simulationen
 - Gazellen
 - Geparden



[2]



Gemeinsamkeiten und Unterschiede

- Grundlegend: E/A
 - Wahrnehmung der Umwelt
 - Möglichkeiten zur Einflussnahme
 - Fähigkeiten-Repertoire
- Gedächtnis (unterschiedlich stark ausgeprägt)
- Unterschied: Entscheidungsprozess
 - Bäume reagieren auf äußere Einflüsse
 - Menschen und Tiere denken und planen
 - Priorisierung von Zielen

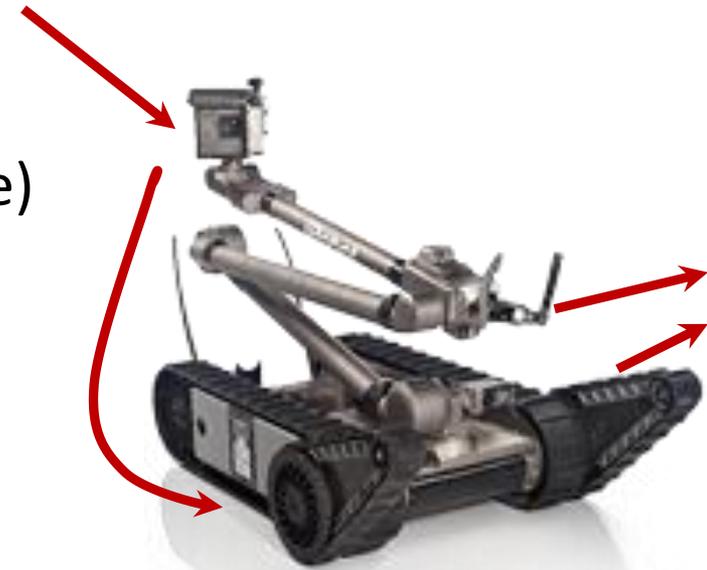
Agenten

- Allgemeines über Agenten
 - Grundlegende Funktionsweise
 - Eigenschaften von Agenten
- Klassische Agentenarchitekturen
 - Reaktive Agenten
 - Proaktive (kognitive) Agenten
 - Hybride Agenten / Schichtenmodelle
- Paper zu generischen Ansätzen
 - GAA
 - GEAMAS

Allgemeines über Agenten

Grundlegendes Funktionsprinzip:

- Sensoren
 - Nehmen die Umwelt wahr (Eingabe)
- Agentenprogramm
 - Informationsverarbeitung
- Aktuatoren
 - Agieren mit der Umwelt (Ausgabe)



PackBot 510, [1]

Eigenschaften eines Agenten

"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

aus: *"Is it an Agent, or Just a Program?"* von Stan Franklin & Art Graesser, 1997

- Autonomie
- Unabhängigkeit
- Evolution

Klassische Agentenarchitekturen

Reaktive Agenten:

- zustandslos (verfügen über kein eigenes Wissen)
- fällen keine Entscheidungen, sondern agieren über Entscheidungsbäume
 - Bedingung → Aktion - Regeln
- Beobachtende Agenten verfügen zusätzlich über ein Gedächtnis.
- → ungeeignet für komplexe Situationen

Klassische Agentenarchitekturen

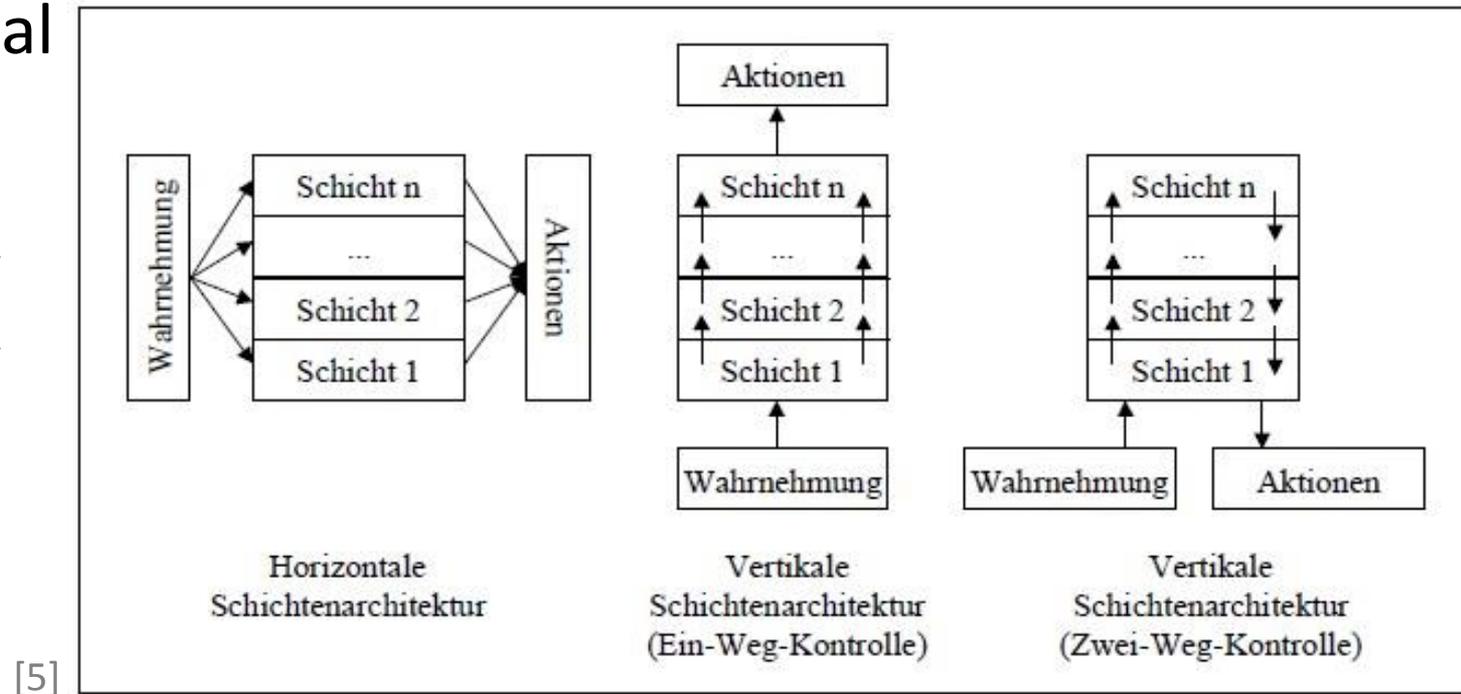
Proaktive Agenten:

- verfügen über eigenes Wissen
- verfolgen Ziele, bilden Pläne
 - Zielbasiert
 - Nutzenbasiert
- → Kognitive (denkende/bewusste) Agenten
 - häufig: BDI-Agenten (*Beliefs, Desires, Intentions*)

Klassische Agentenarchitekturen

Hybride Agenten / Schichtenarchitektur:

- horizontal
- vertikal
 - 1-pass
 - 2-pass



GAA

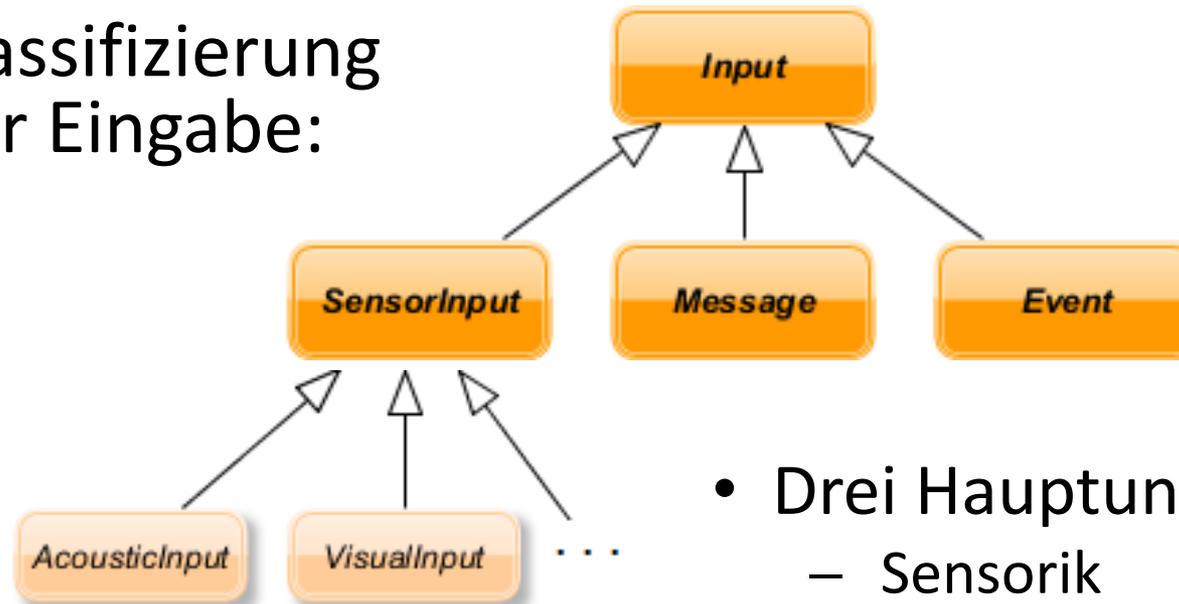
A Generic Agent Architecture for Multiagent Systems

- José M. Vidal (University of South Carolina)
- Paul Buhler (College of Charleston)
- guter Einstieg, viele Grundlagen
- sehr "implementierungsfreundliches" Paper
- wird vorgestellt an *Biter*
 - Implementation für RoboCup
- sowohl reaktiv als auch proaktiv verwendbar!



GAA

Klassifizierung
der Eingabe:



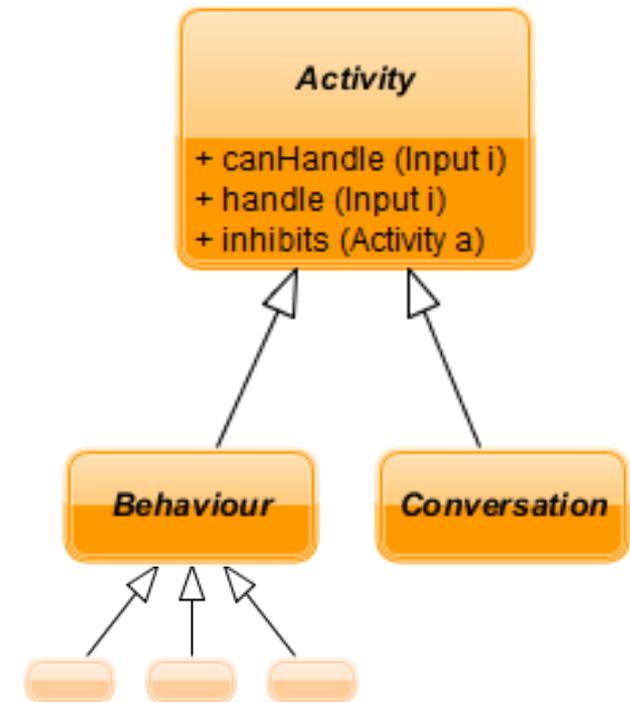
- Drei Hauptunterscheidungen:
 - Sensorik
 - Kommunikation
 - Ereignisse (interne Auslöser)

➤ Der Nutzer erweitert diese Hierarchie domänenspezifisch

GAA

Aktionen:

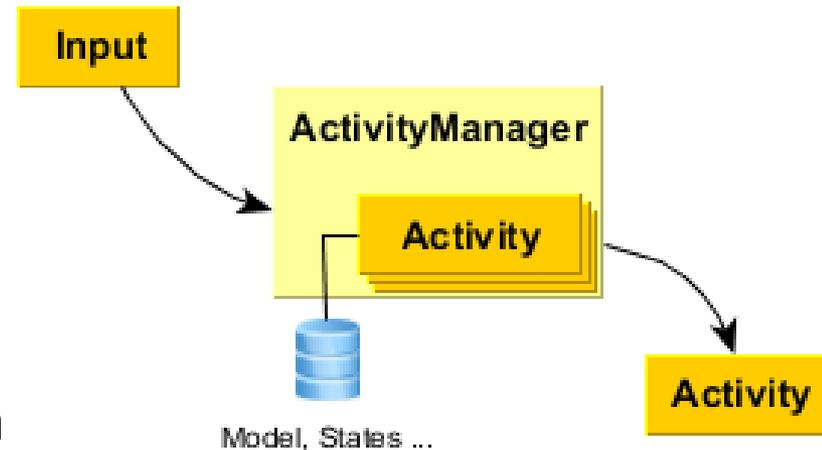
- Zwei Obermengen
 - (Langzeit-) Verhalten
 - Kommunikation
- Erweiterbare Baumstruktur
- Basisfunktionen:
 - *canHandle*: Ist die Aktivität ausführbar? („ob?“)
 - *handle*: Ausführung, erzeugt atomare Aktionen („wie?“)
 - *inhibits*: Blockiert diese Aktivität eine andere? („wann?“)



GAA

Planung und Ausführung:

- *ActivityManager*
 - Verwaltet alle Aktivitäten
- Pro Eingabe:
 - Aktivitäten auf Nutzen prüfen
 - Auf Blockaden testen
 - Eine auswählen und ausführen



GAA

Fazit:

- Sehr einfaches, flexibles Design
 - ... aber auch nur rudimentär!
 - Ziele und Absichten nicht näher spezifiziert
 - BDI-Logik ist selbst hinzuzufügen
 - Beschränkung der kompletten Steuerlogik auf die zwei *Activity*-Funktionen
- Eignet sich nur als Grundlage

GEAMAS

A Generic Architecture for Multi-Agent Simulations

- Pierre Marcenac (Université de La Réunion, Saint Denis)
- Sylvain Giroux (Université de Montréal, Kanada)
- zur Simulation von komplexen Systemen der Natur entwickelt
- Idee: "Agent besteht aus Sub-Agenten"
 - Anordnung in Hierarchiestufen
 - Rekursion
 - → Emergenz

Ein Agent als Subsystem

Drei Abstraktionsstufen:

- Simulationssteuerung
 - *Society Model*
- Kognitive Agenten
 - *Agent Model*
- Reaktive Agenten
 - *Cell Layer*

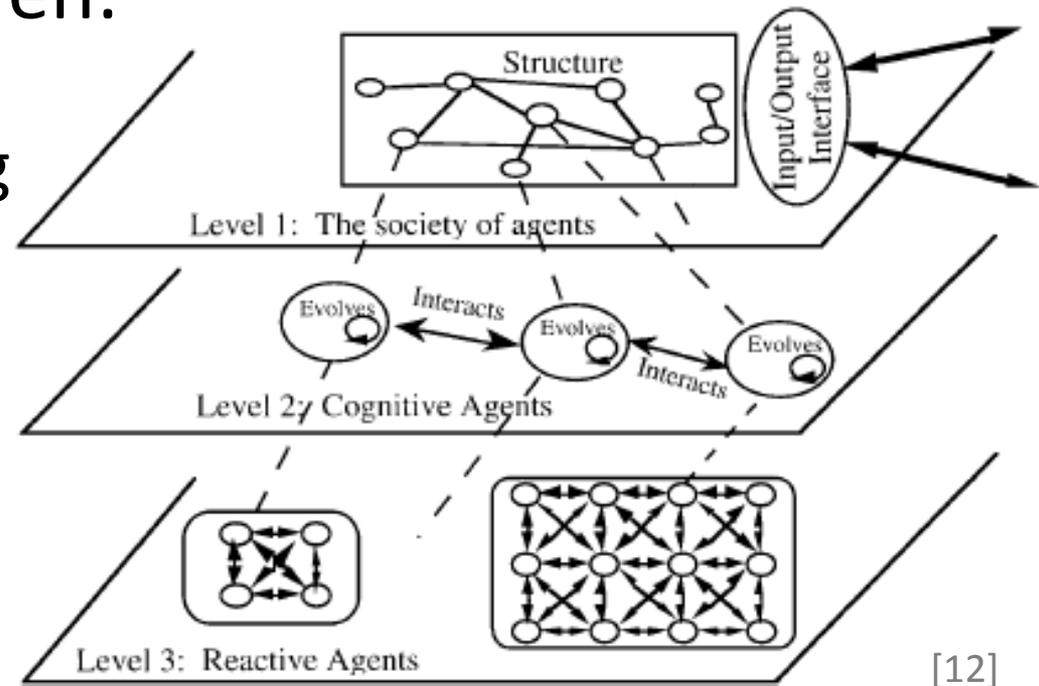
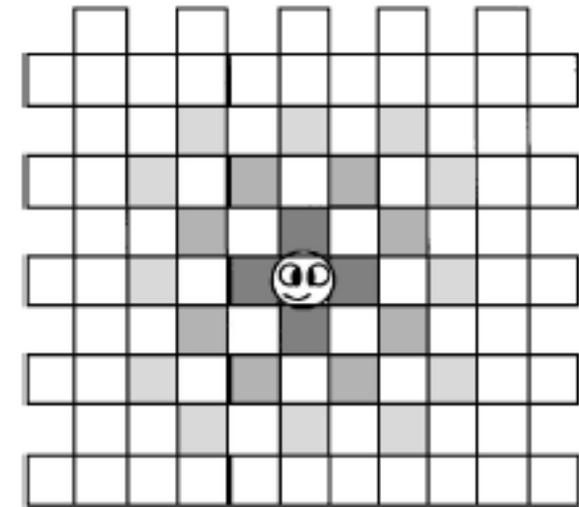


Figure 2. Three abstraction levels to model the complexity of a system.

Interne Kommunikation

Informationsaustausch:

- Interaktion ist der Schlüssel zu Emergenz
- Zwei Kommunikationstypen:
 - Innerhalb einer Schicht
 - Zwischen den Schichten
 - Nach oben: *Rekomposition*
 - Nach unten: *Dekomposition*



Second-level Agent



1st degree of communication
(4 acquaintances)



2nd degree of communication
(12 acquaintances)



3rd degree of communication
(24 acquaintances)

Figure 3. Degree of communication scaling in GEAMAS. [13]

GEAMAS

Fazit:

- Rekursion erleichtert Entwicklung
- Feingliedriger Ansatz
 - Unteragenten können wiederverwendet werden
 - Emergente Effekte innerhalb eines Agenten
- Nur wenig Vorschläge zur Umsetzung

Existierende Frameworks

- Es gibt bereits viele Agententoolkits
 - JADE, MADKit, ZEUS, JATLite, ...
 - Was spricht dagegen?
 - Limitationen
 - Strukturvorgaben
 - Bindung an eine Sprache
- **Eigenbau!**

Ausblick

Weiteres Vorgehen & Zeitplan:

Semester	Fach	Ziel
1. Semester	AW 1	Ideen sammeln, grobe Vorstellung
2. Semester	AW 2 / Projekt 1	Entwurf der Architektur, Beginn der Implementation
3. Semester	Projekt 2	Fertigstellung des Prototypen
4. Semester	Masterarbeit	Evaluation, ??

Konferenzen

- **ACM SIGART**

Special Interest Group on Artificial Intelligence

- **ACM SIGSIM**

Special Interest Group on Simulation

- **IFAAMAS**

International Foundation for Autonomous Agents and Multiagent Systems

- **ECAI**

European Conference on Artificial Intelligence

Wichtige Personen

- M. Wooldridge
- R. A. Brooks
- N. R. Jennings
- José M. Vidal
- S. Franklin

Quellen

Paper:

- GAA: <http://jmvidal.cse.sc.edu/papers/vidaltr0211.pdf>
- GEAMAS: <http://dl.acm.org/citation.cfm?id=590830>

Sonstige Verweise:

- [1] <http://www.irobot.com/~media/Images/iRobot/Robots/Defense/PackBot/Overview/irobot-packbot-landing-left.ashx>
- [2] <http://static.ddmcdn.com/gif/cheetah-speed-2.jpg>
- [3] <http://www.clker.com/clipart-pedestrian-blank-.html>
- [4] http://www.taz.de/uploads/images/684x342/Regenwald_Kongo.20101205-16.jpg
- [5] <http://www.is-frankfurt.de/publikationenNeu/Softwareagenten.pdf>
- [7] <http://www.tzi.de/spl/pub/Website/WebHome/Nao.jpg>
- [12] GEAMAS, Seite 5 (251)
- [13] GEAMAS, Seite 6 (252)

Fragen und Diskussion

