

Verifikation von Multiagentensystemen

Peter Salchow

INF-M3 - Seminar
Wintersemester 2008
Department Informatik
HAW Hamburg

24. November 2008



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen
 - Verifikation
 - MAS
- 3 Risiken
- 4 Ausblick
- 5 Quellen



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen
 - Verifikation
 - MAS
- 3 Risiken
- 4 Ausblick
- 5 Quellen



Motivation I

- Feld der MAS sehr interdisziplinär - Verschiedenste Sichten auf Agenten
- Agenten als Paradigma für Softwareengineering
 - Interaktion innerhalb von Softwaresystemen oftmals sehr komplex
 - Entwurf solch komplexer Systeme mit vielen dynamischen Komponenten schwierig
 - Modellierung unabhängiger, interagierender Einheiten unter dem Aspekt von MAS
- Ubiquitäre, Vernetzte Systeme
 - Viele mobile Systeme
 - Systeme interagieren miteinander
 - Einsatz von Agenten als Möglichkeit das ganze Potenzial solcher Systeme zu nutzen
- Viele Systeme bereits mit Hilfe von Agenten realisiert

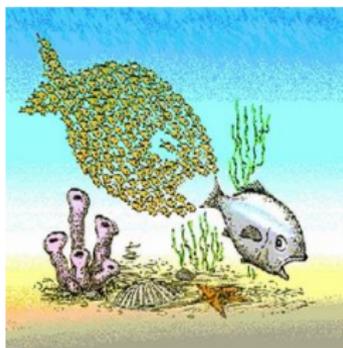


Motivation II

Einsatzgebiete von Agenten

- Unabhängige Einheiten im Meer zur Kontrolle von Ölunfällen (Uni Oldenburg)
- Stundenplanerstellung über Agenten (FH Gelsenkirchen)
- Optimierung des Paketversands von DHL durch Agenten (HAW)

Zusammenspiel vieler Agenten kann emergentes Verhalten hervorrufen - Schwarmverhalten



Motivation III

- Für „normale“ Softwaresysteme existieren Ansätze und Methoden zur Verifikation
- Bereits in diesem Umfeld schwierig
- Wenn sich Paradigma der MAS weiter durchsetzt werden u.U. auch sicherheitskritische Systeme darüber realisiert
- Besonders solch kritische Systeme sind häufig komplex, so dass sich Agenten-Paradigma anbietet
- Multi-Agenten-Systeme skalieren sehr gut



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?
- Welche Unterschiede gibt es zwischen allgemeinen verteilten Systemen und MAS?



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?
- Welche Unterschiede gibt es zwischen allgemeinen verteilten Systemen und MAS?
- Welche Aspekte sind relevant?



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?
- Welche Unterschiede gibt es zwischen allgemeinen verteilten Systemen und MAS?
- Welche Aspekte sind relevant?
- Welche Formalismen (Methoden) bieten sich dabei an?



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?
- Welche Unterschiede gibt es zwischen allgemeinen verteilten Systemen und MAS?
- Welche Aspekte sind relevant?
- Welche Formalismen (Methoden) bieten sich dabei an?
- Welche Vorgehensmodelle erscheinen geeignet?



Vision

Herausforderung

Wie können Multiagentensysteme verifiziert werden?

- Welche Grenzen existieren?
- Welche Unterschiede gibt es zwischen allgemeinen verteilten Systemen und MAS?
- Welche Aspekte sind relevant?
- Welche Formalismen (Methoden) bieten sich dabei an?
- Welche Vorgehensmodelle erscheinen geeignet?



Ziele

Kombination von Multiagentensystemen und formaler Verifikation

- Bis zu welchem Grad können Agenten verifiziert werden?
- Multiagentensysteme eher chaotisch
- Verifikation sehr formal
- Emergentes Verhalten sicherlich nicht verifizierbar
- Welche Eigenschaften sollten lokal geprüft werden, welche für Gruppe von Agenten?



Ziele

Abgrenzung von allgemeinen verteilten Systemen und MAS

- Sind beide Systeme völlig unterschiedlich?
- Wahrscheinlich nicht
- Existierende Verifikationsansätze für verteilte Systeme erarbeiten
- Welche Ansätze lassen sich auf MAS übertragen



Ziele

Bestimmte Aspekte der Multiagentensysteme analysieren und verifizieren

- Was soll am MAS verifiziert werden?
- Verklemmungen? - Ist uninteressant
- Ist das erreichte Ziel semantisch korrekt - Auswertung der Ergebnisse
- Stehen gegensätzliche Interessen im fairen Verhältnis



Ziele

Prüfung der Formalismen auf ihre Tauglichkeit zur Verifikation von MAS

- Welche Formalismen existieren überhaupt?
- Mit hoher Wahrscheinlichkeit gibt es keinen Formalismus der alle Probleme abdeckt
- Vor- und Nachteile der Formalismen für bestimmte Problemstellungen
- Kombination von Formalismen und direkter Codeverifikation



Ziele

Erarbeiten von Methoden zur erfolgreichen Verifikation

- Existieren allgemein gültige Ansätze?
- Für alle MAS - Zu komplex!!!
- Allgemein gültige Ansätze für bestimmte Bereiche von MAS
- Identifizieren von Pattern für bestimmte Probleme



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen**
 - Verifikation
 - MAS
- 3 Risiken
- 4 Ausblick
- 5 Quellen



Formale Verifikation

- System ist korrekt, wenn es die Anforderungen erfüllt
- Dazu gehören
 - Funktionale Anforderungen - Was muss das System können?
 - Safety Anforderungen - Wie zuverlässig und betriebssicher muss das System sein?
 - Performance Anforderungen - Wie schnell muss das System arbeiten?
- Es reicht nicht aus zu zeigen, dass Anforderungen erfüllt werden können
- Ziel ist es zu zeigen, dass System Anforderungen ausnahmslos erfüllt
- Erfüllen der funktionalen Anforderungen relativ einfach zu prüfen
- Erfüllen der Safety und Performance Anforderungen sehr aufwändig
- Zeitaspekte können nur schwierig verifiziert werden - Performance daher nicht Thema der Master-Thesis



Werkzeuge

- FDR - Failures Divergences Refinement 
 - Tool zur Analyse von Sicherheitsprotokollen
 - Modellierung in CSP (Communicating Sequential Processes)
 - Modellierung von kommunizierenden Prozessen
 - Kommunikation steht im Vordergrund - Traces
- SPIN - Simple Promela INterpreter 
 - Formale Verifikation von verteilten Systemen
 - Modellierung in PROMELA (PROcess MEta LAnguage)
 - Prozessorientiert - z.B. jeder Agent ist Instanz eines PROMELA-Prozesses
- Netlab 
 - Analyse von Petrinetzen
 - Modellierung von MAS im Petrinetz sehr aufwändig
 - Verifikation sehr komplex



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen**
 - Verifikation
 - MAS
- 3 Risiken
- 4 Ausblick
- 5 Quellen



Multiagentensysteme (MAS)

Multiagentensysteme

Multiagent systems are systems composed of multiple interacting computation elements, known as agents.

[Wooldrige06]

Agenten:

- Fähigkeit Aktionen autonom auszuführen - Entscheiden selbst, was sie tun müssen um ein Ziel zu erreichen.
- Können mit anderen Agenten interagieren - Kooperation, Koordination, Verhandlung, etc.



Agententypen

- Reaktive Agenten
 - Agenten haben prinzipiell kein eigenes Wissen
 - Reagieren auf Eingaben aus Ihrer Umwelt
 - Reaktion fest vorgegeben
- Deliberative Agenten
 - Verwalten internen Zustand
 - Eigenes Wissen - Suche nach bestmöglichem Verhalten
 - Wägen Vor- und Nachteile von Aktionen ab
- Hybride Agenten
 - Vermischt beide Typen
 - Zuerst reaktives Verhalten anwenden
 - Dann erst Entscheidungsfindung über internes Wissen

⇒ Reaktive Agenten weniger komplex

⇒ Formalismen leichter anwendbar

⇒ Erkenntnisse auf Deliberative Agenten übertragen



AgentSpeak

Ansatzpunkt für die Verifikation: AgentSpeak

- Regelbasierte Programmiersprache für BDI-Agenten
- Beliefs werden als Fakten vorgegeben (ähnlich Prolog)
- Es existieren bereits Ansätze AgentSpeak direkt zu verifizieren
- Ausgangspunkt für weitere Recherche



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen
 - Verifikation
 - MAS
- 3 Risiken**
- 4 Ausblick
- 5 Quellen



Risiken

Risiko

Gegensätzlichkeit beider Themen:
MAS von Natur aus chaotisch vs. Sehr formale Verifikation

Lösung:

- Herausfinden, wie bestehende Projekte an dieses Thema herangegangen sind
- Deterministische Eigenschaften von MAS identifizieren und Formalismen darauf anwenden



Risiko

Das gewählte Thema ist zu komplex, um es in einer Master-Thesis zu bearbeiten.

Lösung:

- Beschränken auf bestimmte Aspekte
- Nach tieferer Einarbeitung Thema klar abgrenzen
- Nicht abstrakter an das Thema gehen als nötig



Risiko

Es stellt sich als unmöglich heraus, eine allgemeine formale Analyse von MAS durchzuführen.

Lösung:

- Erkenntnis erlangen und Grenzen aufzeigen
- Begründung für die Unmöglichkeit suchen
- Formale Analyse auf weniger abstrakter Ebene durchführen



Inhalt

- 1 Einleitung
 - Motivation
 - Vision
- 2 Vorgehen
 - Verifikation
 - MAS
- 3 Risiken
- 4 Ausblick**
- 5 Quellen



Das Ziel ist jetzt klar! Doch was ist zu tun...

- Weitere Quellen recherchieren
- AW2 vorbereiten
- Quellen gezielt auswerten
- Anhand der Erkenntnisse aus Quellen Thema weiter eingrenzen
- Bestimmte Gebiete auswählen und einarbeiten
- Kontakt zu Spezialisten auf den Gebieten aufnehmen



Quellen I

[Alechina06]

Alechina, Natasha: Verifying space and time requirements for resource-bounded agents. In: International Conference on Autonomous Agents. ACM, 2006

[Bordini04]

Bordini, R.H. [u.a.]: Model checking rational agents. In: IEEE Intelligent Systems. 2006

[Dennis08]

Dennis, Louise A. [u.a.]: A flexible framework for verifying agent programs. In: International Conference on Autonomous Agents. ACM, 2008

[Henesey08]

Henesey, Lawrence: Agent based simulation architecture for evaluating operational policies in transshipping containers. Springer, 2008



Quellen II

[Holzmann04]

Holzmann, Gerard J.: The Spin Model Checker : Primer and Reference Manual. 2004

[Kramer07]

Kramer, Jeff: Is abstraction the key to computing?. In: Communications of the ACM: 2007.

[Lam04]

Lam, Dung N. [u.a.]: Verifying and Explaining Agent Behavior in an Implemented Agent System. In: International Conference on Autonomous Agents. ACM, 2004

[Roos07]

Roos, Nico; Witteveen, Cees: Models and methods for plan diagnosis. Springer, 2007



Quellen III

[SpinWeb]

<http://spinroot.com/spin/whatispin.html>

[Sudeikat08]

Sudeikat, Jan; Renz, Wolfgang: A Systemic Approach to the Validation of Self-Organizing Dynamics within MAS. 2008

[Wooldridge06]

Wooldridge, Michael: An Introduction to Multiagent Systems. 2006



Vielen Dank für die Aufmerksamkeit!
Fragen?

