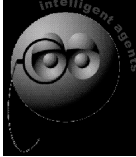


Intro

„Agent-Oriented Software Engineering for Internet Applications“



Tim Conrad
conrad@inf.fu-berlin.de

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

Intro

"The buzzword agent has been used recently to describe everything: from a word processor's help system to mobile code that can roam networks to do our bidding.

The metaphor has become so pervasive that we're waiting for some enterprising company to advertise its computer power switches as empowerment agents."

Peter Wayner "Agents of Change" in Byte (1995).

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

Programm

Worum geht es heute: Das Programm

- Kurze Wiederholung von Agent und MAS
- Prinzipien der Softwareentwicklung für agentenbasierte Internet Anwendungen
- Ein ausgewähltes Beispiel (GAIA)
- Schwachstellen an dieser Technik / Methodik
- Verbesserungen durch soziale Strukturen
- Zusammenfassung / Ausblick

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

WDH

Wiederholung

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

WDH
Software Agent

WDH: „intelligente“ Software Agent

- Engl. „agent“: „im Auftrag Handelnder“

„Agents are software entities that assist people and act on their behalf“
P. Maes (1994)

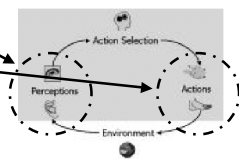
- Programm mit besonderen Eigenschaften
 - situiert
 - autonom
 - kollaborativ
 - mobil

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

WDH
Software Agent – „situiert“

„situiert“

- Agenten arbeiten in einer Umgebung und haben
 - Wahrnehmungen
 - Aktionen



„An agent is a computer system that is **situated** in some environment, and that is capable of autonomous action in this environment to meet its design objectives“
Wooldridge (1999)

Intelligente Internet Agenten – Agent-Oriented Software Engineering for Internet Applications

WDH
Software Agent – „autonom“

„autonom“

- Agenten entscheiden selbständig, wann und welche Aktionen sie ausführen
 - keine zentrale Kontrolle
 - Arten: Reaktiv / Proaktiv („reflexhaft“)

WDH
Software Agent – „kollaborativ“

„kollaborativ“

- Agenten arbeiten in Teams und benutzen Kommunikation
 - Synchron
 - Asynchron
 - „soziales Verhalten“ zu anderen Agenten aber auch Menschen

WDH
Software Agent – „mobil“

„mobil“

- Agenten können ihren Server verlassen und zu einem anderen Server migrieren
- mobil bezogen auf Umwelt -> (Inter-)Net, Mars, ...

1. Server 1 Shutdown request
2. Migration
3. Weiterarbeit auf Server 2
4. Wartung Server 1
5. Migration
6. Arbeit auf Server 1

Internetagenten

Internetagenten

- Autonome „Software Entität“
- Interagiert proaktiv mit seiner Umgebung (Internet, anderen Agenten)
- Ziel: eigene Aufträge erreichen (z.B. Informationen sammeln)

WDH
Agentensystem

WDH: Agentensystem

- Umgebung mit mehreren (typischerweise) interagierenden Agenten
- Stellt grundlegende Dienste für Agenten zur Verfügung
 - Nachrichten-Dienste (Interaktion zw. Agenten)
 - („soziale“) Beziehungen, Abhängigkeiten zwischen Agenten -> Hierarchien
 - Sicherheitsdienste
- Hat oftmals eigenes „Systemziel“

WDH
Agentensystem

Warum Agentensysteme ?

- Neue Technik zur Entwicklung von teilweise hochkomplexen Softwaresystemen in verschiedensten – teilweise unvorhersehbaren - Anwendungsgebieten
 - Künstliche Intelligenz
 - Robotik („mars explorer“)
 - „Personal Assistants“ (workflow systems)
 - ...

AOSE

**Agentenorientierte
Software Entwicklung**

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

AOSE

**„klassisches“ vs. „agentenbasiertes“
Software-Engineering**

- Klassisches Software-Engineering
 - Der Entwickler antizipiert beim Entwurf des Systems jede mögliche Aktion, die das System ausführen können muss („Objektorientiert“)
- Agent-basiertes Software-Engineering
 - Der Entwickler entwirft ein System, das „selbst entscheidet“, auf welche Weise es durch sein Verhalten die Entwurfsziele am besten erfüllt

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

AOSE
Einführung

**Agenten-orientiertes
Software Engineering**

- Keine derzeitige Technik kann derzeit als ausreichend betrachtet werden
- Phasenorientierte Entwicklung endet meist in frühen Designstadien
- Verschiedene Ansätze:
 - architekturabhängig: JAFMAS, TAEMS
 - objektorientiert: Agentbuilder, dMARS, AgentFUSION
 - objektorientiert & Agentensystemmodelliert: GAIA

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

GAIA
Einführung

**Vom Bedarf zum Entwurf:
Entwicklung von Multi-Agenten-Systemen**

- GAIA: Ein Ansatz zur Analyse und Entwurf von Agentensystemen
 - M. Wooldridge, N. R. Jennings, and D. Kinny: „The Gaia Methodology for Agent-Oriented Analysis and Design“ in: Journal of Autonomous Agents and Multi-Agent Systems. 3(3):285-312. (2000)

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

GAIA

Vom Bedarf zum Entwurf: GAIA

- Besteht aus Modellen für
 - Analyse der Anforderungen
 - Design der Softwareumsetzung
- Ziel: agentenorientierte Strukturierung der Anforderungen
- Vorbereitung für softwaretechnische Umsetzung (meist klassisch objektorientiert)

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

GAIA
Überblick

GAIA - Überblick

```

    graph TD
      A[Anforderungen] --> B[Rollenmodell]
      A --> C[Interaktionsmodell]
      B --> D[Agentenmodell]
      B --> E[Servicemodell]
      C --> E
      C --> F[Bekanntschftsmodell]
      subgraph Analyse
        A
        B
        C
      end
      subgraph Design
        D
        E
        F
      end
  
```

Intelligente Internet Agenten - Agent Oriented Software Engineering for Internet Applications

GAIA
Beispiel - Anforderung

GAIA: Beispiel

- Anforderung: Entwicklung eines MAS zum Managen des Begutachtungsprozesses einer (grossen, internationalen) Konferenz
 - Konferenzleitung bekommt Paper
 - Mitglieder der Konferenzleitung bekommen Teile der Paper und verschicken sie an Gutachter
 - Gutachter bewerten Paper und schreiben Bericht
 - Konferenzleitung sammelt Begutachtungen und akzeptiert Paper oder lehnt diese ab

GAIA
Analyse: Rollenmodell

GAIA

- Analyse
- Rollenmodell
 - Strukturierung durch Definition von Rollen und deren Verantwortungen / Rechten
 - Im Beispiel:
 - Konferenzleitung: muss Paper einsammeln, verteilen, Berichte einsammeln, Einstufen
 - Begutachter: muss Paper begutachten und bewerten

GAIA
Analyse: Interaktionsmodell

GAIA

- Analyse
- Interaktionsmodell
 - Beschreibung der Abhängigkeiten und Beziehungen der einzelnen Rollen
 - Definiert Kommunikationspfade
 - Im Beispiel:
 - Konferenzleitung: sucht und beauftragt Begutachter
 - Begutachter: bekommt Paper von Konferenzleitung und sendet Bericht

GAIA
Design - Übersicht

GAIA

- Design
 - Ziel: ausreichend genaue Spezifikation um einzelne Agenten zu entwerfen und implementieren
 - Wie kann mit Agenten Systemziel erreicht werden
 - Welche Aufgaben übernehmen Agenten dabei
 - NICHT: WIE realisieren die Agenten diese Aufgaben

GAIA
Design - Agentenmodell

GAIA

- Design
 - Agentenmodell
 - Welche Agenten sind im System vorhanden
 - Welche Rollen übernehmen welche Agenten
 - Mehrere Rollen können in einem Agenten abgebildet werden
 - Eine Rolle kann von mehreren Agenten übernommen werden (Lastverteilung)

GAIA
Design - Servicemodell

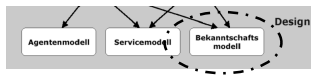
GAIA

- Design
 - Servicemodell
 - Angebotene Services der Agenten
 - Definitionen pro Service (Beispiel: Begutachtung)
 - Input („Paper“)
 - Output („Gutachten“)
 - Vorbedingungen („Paper liegt vor und soll begutachtet werden“)
 - Nachbedingungen („Gutachten ist erstellt und an Konferenzleitung geschickt“)

GAIA
Design

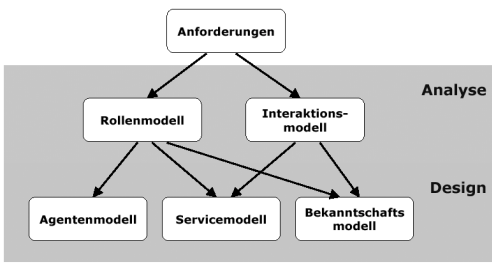
GAIA

- Design
 - Bekanntschaftsmodell
 - Graph von Kommunikationspfaden („Welche Agenten kommunizieren untereinander?“)
 - Soll helfen, „Flaschenhalse“ in der Kommunikation aufzudecken



GAIA
Überblick

GAIA - Überblick



GAIA: Nachteile

GAIA: Nachteile

- GAIA modelliert geschlossene Systeme
 - Art der Agenten muss vorher bekannt sein
 - z.B. Kommunikationseigenschaften
-> Was passiert, wenn Gutachter eigenen Agenten zum „Paper holen“ schickt ?
 - positive Kollaboration wird angenommen
 - -> Was passiert wenn sich „Gutachter-Agent“ 50% aller Paper holt ? Oder 3x dasselbe ?
- Keine überwachende Instanz für „soziale“ Regeln

Probleme

Probleme im offenen System „Internet“

- Internet (bzw. Agent darin) ist
 - offen
 - vielförmig
 - dynamisch (auch in Strukturen)
 - unvorhersehbar
 - nicht immer „ehrlich kooperativ“

Verbesserungen

Verbesserungen: Koordination

- Übergeordnete Kontrollschicht koordiniert
 - Ereignisse
 - Kommunikation zwischen Agenten
 - Sicherheitsbestimmungen
 - Erreichen globaler sowie eigennütziger Ziele

Verbesserungen

Verbesserungen: Koordination

- Übergeordnete Koordinations-Schicht bestehen aus
 - zu koordinierenden Objekte (-> Agenten)
 - Unter-Schichten
 - Überwachung
 - Kanäle (Kommunikation, Datenaustausch, ...)
 - Koordinations-Regeln
 - Definiert Reaktionen auf Interaktionen der Agenten

Verbesserungen

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Arten von Koordinations-Schichten

- Kontrollbasierte Koordinations-Schicht
 - Agenten interagieren mit anderen Agenten oder Aussenwelt über definierte Schnittstellen
 - Kontrolle über überwachte Statusänderungen und Ereignisse an diesen Schnittstellen
- Datenbasierte Koordinations-Schicht
 - Interaktion über Austausch von Datenstrukturen
 - Kontrolle über die Zugriffsmöglichkeiten auf diese Daten im Datenraum der Koordinationsschicht

Vorteile

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Vorteile

- sämtliche Kommunikation läuft über / wird kontrolliert von Koordinations-Schicht
 - „böartige“ bzw. eigennützige Agenten können keinen / wenig Schaden anrichten
- Möglichkeit zur Implementation von „sozialen Regeln“
 - Kommunikation / Interaktion zwischen Agenten aufgrund dieser Regeln
- -> Kein Gutachter-Agent kann bei mehreren Mitgliedern des Planungskomitees Paper holen

Vorteile

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Vorteile

- Interaktionen zwischen Agenten, auch wenn sie nicht „kompatibel“ sind / sich nicht „kennen“ -> Eigener Gutachter Agent kann Paper holen
- Interaktions-Verbindungen zwischen allen möglichen Agenten müssen nicht vorher festgelegt werden
 - Möglichkeit von „unvorhergesehenen“ Agenten
 - Agenten müssen nicht alle „Sprachen“ / Interaktions-Protokolle beherrschen

Vorteile

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Vorteile

- keine Vermischung von „eigennützigen“ Agenten Zielen und „globalen“ System-Zielen
- Am Beispiel:
 - „Load Balancing“: Alle Mitglieder des Planungskomitees werden gleich belastet
 - Kein Gutachter darf mehr als ein Paper desselben Autors haben
 - Kein Gutachter darf mehr als 3 Paper insgesamt haben

Neuer Überblick

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Verbesserte Struktur


Zusammenfassung

Intelligente Internet-Agenten - Agent Oriented Software Engineering for Internet Applications

Zusammenfassung

- WDH: Agent & Multi-Agentensysteme
- agentenbasiertes Software Engineering anhand von GAIA
- Verbesserungen von GAIA durch Einführung „sozialer“ Kontrollsysteme

Ausblick




Intelligente Internet-Agenten — Agent-Oriented Software Engineering for Internet Applications

Ausblick

- Agententechnologie ist aus den belächelten Anfängen erwachsen
- Sinnvolle Konzepte wurden erforscht und vorgestellt
- Agentenorientiertes Software Engineering stellt Methoden zur Verfügung, mit denen Agentensysteme entworfen werden können
- Mehr (zur Zeit) aber auch nicht...


The End



Intelligente Internet-Agenten — Agent-Oriented Software Engineering for Internet Applications

Vielen Dank für die Aufmerksamkeit !

References



Intelligente Internet-Agenten — Agent-Oriented Software Engineering for Internet Applications

Quellenangaben

- F. Zambonelli, N. R. Jennings, A. Omicini, M. J. Wooldridge: „Agent-orientet Software Engineering for Internet Applications“, „Coordination of Internet Agents: Models, Technologies and Applications“, Springer 2001
- C. Dannegger, K. Dorer: „Agenten in Raucherkinos“, Java Magazin 7/2001, S. 65-67
- H. Knublauch, T. Rose, H. Köth, W. Friesdorf: „Modellierung des Wöchnerinnenszenarios mit der Gaia Methode nach Wooldridge et al“, 2. Kolloquium zum DFG-Schwerpunktprogramm "Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien", 11/2000