

Eine Simulationsunterstützung für Agentenplattformen

Hinnerk Gildhoff

Diplomarbeit

Seite 1/20 VSIS Oberseminar: Zwischenbericht 06.07.2006 UH


## Gliederung

- Motivation / Ziele
- Grundlagen
  - Agenten / Multiagentensystem / Agentenplattform
  - Simulation
  - Multiagentensimulation
- Vorgehen
- Implementierung & Design
- Aktuelle Fragen
- Beispielanwendung
- Ausblick

Seite 2/20 Eine Simulationsunterstützung für Agentenplattformen 06.07.2006 UH

## Motivation / Ziele

- Simulation für Agenten
  - Analyse und Design komplexer Systeme
- Agenten für Simulationen
  - Entwicklung neuer Simulationstechniken
  - Entwicklung von domänenspezifischen Simulationen
- Vereinigung von Simulationstechniken mit den Konzepten einer Agentenplattform
- Entwicklung einer Simulationskomponente mit verschiedenen Simulationsmodi
- Entworfenen Applikation ist bereit für den realen Einsatz und kann *jederzeit* simuliert werden




Seite 3/20 Eine Simulationsunterstützung für Agentenplattformen 06.07.2006 UH

## Agenten

Eine *anerkannte* Definition:  
 "An *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its design objectives".  
 (Wooldridge and Jennings 1995)

Ständige Interaktion kann zu Fehlern führen

- Umgebung ist meist nicht deterministisch
- Agent trifft eigene Entscheidungen



Seite 4/20 Eine Simulationsunterstützung für Agentenplattformen 06.07.2006 UH

## Intelligente Agenten

### Eigenschaften

- reaktiv (zeitliche Reaktion auf ein Ereignis)
- proaktiv (frühzeitig Initiative übernehmen)
- sozial (verhandeln und kooperieren)
- befindet sich in einer Umgebung
- autonom (Entscheidungsfreiheit)
- flexibel (flexible Reaktionen möglich)
- robust (Fehler als ständige Option)
- lernfähig (interne Zustände und Aktionsmöglichkeiten ändern)

Seite 5/20 Eine Simulationsunterstützung für Agentenplattformen 06.07.2006 UH

## Multiagentensysteme

Ein Multiagentensystem (MAS) ...

- beinhaltet autonome Agenten, die gegen- oder miteinander arbeiten
- braucht eine Infrastruktur für die Kommunikation und Interaktion der Agenten untereinander (kooperieren, koordinieren, verhandeln)

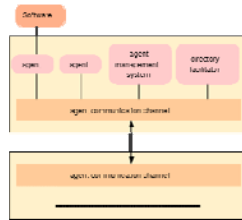
Weitere Eigenschaften (ergeben sich aus der Autonomie):

- Beschränkte Sicht des einzelnen Agenten auf das Gesamtsystem
- Dezentrale Datenhaltung (lokale Datenverwaltung)
- Asynchrone Berechnungen der Agenten (Nebenläufigkeit)
- Idealfall: keine zentrale Kontrolle (wird meist relativiert)

Seite 6/20 Eine Simulationsunterstützung für Agentenplattformen 06.07.2006 UH

## Agentenplattform

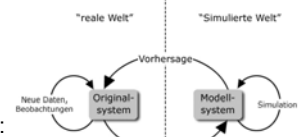
- Stellt eine Infrastruktur zur Verfügung, auf der agentenbasierte Systeme ablaufen und entwickelt werden können
- FIPA-Referenzmodell fordert drei wichtige Komponenten:
  - Agent Management System (AMS)
  - Directory Facilitator (DF)
  - Agent Communication Channel (ACC)



## Simulation

Definition:

A *computer simulation* is a computation that models the behaviour of some real or imagined system over time. (Richard M. Fujimoto)



Gründe für Simulationen:

- Finde die effektivste Strategie
- Experimente am Modell führen zu neuen Erkenntnissen
- Experimente am Realsystem sind zu teuer oder sogar zu gefährlich
- Vorgänge laufen in der Realität zu schnell oder zu langsam ab

## Simulation

Was braucht ein Simulationsprogramm?

- Zustandsvariablen
- Mechanismus für Zustandsänderungen
- Präsentation von Zeit
  - Physical time (Zeit im physikalischen System)
  - Simulation time (Abstraktion der Simulation, um physikalische Zeit zu modellieren)
  - Wallclock time (Eigentliche Ausführungszeit der Simulation – Hardware Clock)

Ausführung der Simulation (temporaler Aspekt)

- real-time (simulation time = wallclock time)
- scaled real-time (konstant schneller oder langsamer)
- as-fast-as-possible (Normalfall)

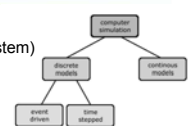
Mapping Funktion: Umwandlung der "wallclock-" in "simulation time"

- $T_s = W2S(T_w) = T_{sStart} + Scale * (T_w - T_{wStart})$

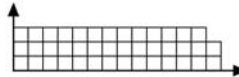
## Klassifikation von Simulationen

Änderungszeitpunkt der Zustandsvariablen

- Kontinuierliche Simulation (Differentialgleichungssystem)
- Diskrete Simulation

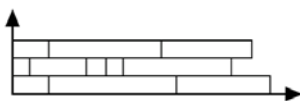


- time-stepped
  - Aufteilung in gleich große Schritte
  - Behandelt Events schrittweise (innerhalb eines „time steps“)
  - Kausal abhängige Events müssen in unterschiedlichen Schritten liegen
  - „time step“ muss gut gewählt sein
  - Einsatzgebiete: Wetterveränderungen, Fortbewegung eines Autos, ... (regelmäßige Events)

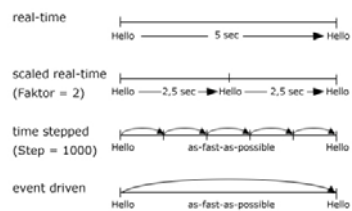


## Klassifikation von Simulation

- event-driven
  - Zustandswechsel erfolgt nur bei einem Event
  - Ein Event besteht aus: Zeitstempel, Typ des Events, weitere Parameter
  - Benutzt eine geordnete Liste von Events
  - Entscheidend ist die Generierung von Events (Initialisierung und zukünftige Events)
  - Einsatzgebiete: Warteschlange vor einer Bank, ... (unregelmäßige Events)



## Simulation - Beispiel



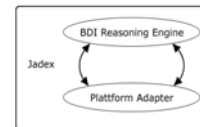
```
while(true) {
    print("hello");
    waiFor(5000);
}
```

## Multiagentensimulation

- Die aktiven Komponenten des zu untersuchenden Systems sind Agenten
- Verwendet Konzepte der Multiagentensysteme
- Ein neuer Weg um großangelegte, komplexe Simulationen zu untersuchen und zu organisieren (biologische, soziale Systeme etc.)
- Realitätsnahe Modellierung (Autonomie, Proaktivität, Sozialität,...)
- Eignet sich besonders für die Modellierung von individuellem Verhalten (Direktheit der Modellierung) und Heterogenität im System
- Weitere Vorteile: schrittweise Konstruktion, verteilte Simulation

## Vorgehen

- Implementierung am Beispiel der Agentenplattform Jadex
- Ansätze
  - Implementierung eines Time-Management-Service auf Anwendungsebene
    - Ein zentralisierter Koordinator (Engpass)
    - Anwendung muss Synchronisationsprotokoll implementieren
  - Implementierung einer Simulationsinfrastruktur auf Plattformebeine
    - "BDI Reasoning Engine" vermeidet Inkonsistenzen eines Agenten
    - "Plattform Adapter" implementiert Simulationsinfrastruktur und sorgt für Wiederholbarkeit



## Implementierung & Design

Wichtige Komponenten:

- BDI Reasoning Engine (bestimmt die Aktionen)
- Plattform Adapter
  - Executor (führt die Aktionen aus)
  - Time Management (wann wird welche Aktion ausgeführt)
  - Scenario Management (initialisiert Simulation und legt Rahmen fest)
- Administrator GUI für Simulationseinstellungen
  - Simulationsstatus (Uhren)
  - Auswahl der Simulationsart und Steuerung der laufenden Simulation
    - Starte, stoppe oder pausiere Simulation
    - Verändere Skalierungsfaktor (scaled real-time)
    - Ändere Time-Steps (time-stepped)
    - Erzeuge Events mittels Event-Generator (event-driven)

## Simulationsinfrastruktur

- Executor
  - Ausführung der Aktionen im Normal- oder Simulationsbetrieb
  - Normalbetrieb arbeitet multi-threaded (Thread Pool)
  - Simulationsbetrieb nutzt sequentielle, schrittweise Ausführung
- Time Management
  - Realisiert Umsetzungen aller Uhren
  - Stellt Simulationsmodi bereit
  - Startet Ausführungsrunden des Executors
- Scenario Management
  - "Scenario Definition File" (SDF) enthält benötigte Agenten, eine Auflistung von Events und die Start- und Endzeit.

## Aktuelle Fragen

- Wie kontrolliert man Aktionen eines Agenten, ohne seine Autonomie zu beschränken und sich trotzdem an das BDI Paradigma zu halten?
- Berechne ein Simulationsschritt: Bei einem "tick" kann jeder Agent *eine* Aktion durchführen. Was genau ist *eine* Aktion?
- Sind alle Anwendungen simulierbar, oder nur eine bestimmte Klasse von Anwendungen (Referenzmodell)?
- Scenario Management: Welche Modellinformationen werden benötigt? Wieviel Arbeit kommt auf den Anwendungsentwickler zu?
- Umsetzung der Wiederholbarkeit?
- Granularität der Zeit Repräsentation?

## Beispielanwendung

- Demonstriert praktische Anwendbarkeit der Konzepte
- Beispiel wird visualisiert und das Ergebnis wird ausgewertet
- Praktische Relevanz wichtig (reale Daten einer Firma / Behörde)
- mögliche Anwendungsdomänen:
  - Luft- und Raumfahrt (zB. Flughafen Hamburg)
  - Verkehrsaufkommen
  - Gesellschaftliche Simulationen

## Ausblick

---

- Schnittstellen für Erweiterungen bereitstellen (Visualisierung, Analyse)
  - dafür benötigt man den aktuellen Zustand eines Agenten und seine Entwicklung
- Ausweitung der Simulationsunterstützung auf mehrere heterogene Agentenplattformen
  - Gateway Agent
  - Synchronisationsprotokoll

## Ende

---

- Fragen?
- Anregungen?
- Kritik?