

Projektbericht

Agententechnologie

Projekt "Realisierung verteilter Agentensysteme" im SoSe 2003
von Christian Poulter

Inhaltsübersicht:

1. Einleitung
2. Projektziel / Szenario
3. Aufgaben des Marktagenten
4. Aufbau und Funktionsweise des Marktagenten
5. Das Wissen des Marktagenten (Beliefbase)
6. Die GUI

Einleitung

Das Ziel des Projektes "Realisierung verteilter Agentensysteme" war die Entwicklung eines Multiagentensystems, welches einen Teil des Spiels Dynatech nachempfinden sollte. Das Projekt bestand im wesentlichen aus zwei Phasen. Während des ersten Teils wurde in kleinen Einheiten die Agententechnologie, das JADE-Framework und die JADEx-Erweiterung vorgestellt und anhand einfacher Aufgaben vertieft. Der Übergang in die zweite Phase wurde mit einer Analyse des Spiels eingeleitet. Die Entwicklung der einzelnen Agenten des Systems wurde dann in kleinen Gruppen vorgenommen. Zum Ende der Projektlaufzeit konnte dann ein einfaches, aber lauffähiges Multiagentensystem vorgestellt werden.

2. Projektziel / Dynatech-Szenario

Während des zweiten Teils des Projektes sollte eine einfache Implementation des Spieles "Dynatech" erfolgen. Die unterschiedlichen Komponenten des Spiels sollten als Agenten entwickelt und programmiert werden.

Das Spiel ist eine einfach Wirtschafts- und Handelssimulation, in der der Spieler mit Hilfe von Förderanlagen, Fabriken und Schiffen Produktionsketten aufbauen muss. Die produzierten Waren kann er dann entweder an einen anderen Mitspieler oder der globalen Abnahmeinstanz (Terra) verkaufen bzw. diese weiterveredeln und somit noch höherwertige und damit wertvollere Produkte erzeugen. Während des gesamten Spiels steht er in ständiger Konkurrenz zu anderen (vom Computer simulierten) Mitspielern. Die Förderanlagen bzw. Fabriken können jeweils nur auf unterschiedlichen Planeten aufgebaut werden, so dass der Spieler mit Hilfe von Raumschiffen den Transport der Güter vornehmen muss oder hierzu Aufträge an Konkurrenten erteilen kann. Ziel des Spieles ist es, möglichst viel Geld zu verdienen und so die Konkurrenz zu verdrängen.

Im Projekt wurden die einzelnen Komponenten des Spieles von kleinen Teams mit Hilfe der Agententechnologie realisiert. Hierzu wurde folgende Aufteilung vorgenommen:

1. Manager

Bietet eine Schnittstelle zwischen der GUI und den übrigen Agenten. Er führt alle vom Spieler (oder automatisch) veranlassten Aktionen durch.

2. GUI

Darstellung der Spielinformationen und die eigentliche Interaktion mit dem menschlichen Spieler.

3. Factories
Simulation einer Fabrik bzw. Förderanlage. Diese fördert Rohstoffe oder produziert eine Ware aus anderen Waren oder Rohstoffen.
4. Spaceship
Raumschiffe führen Transporte der Waren und Anlagen zwischen den Planeten durch. Sie können verschleifen und müssen daher regelmäßig gewartet werden.
5. Terra
Tritt als Endabnehmer der Waren am Markt auf und kauft von allen Spielern die angebotene Waren.
6. Ontology
Erarbeitung einer gemeinsamen Kommunikationsschnittstelle.
7. Market
Ein System um Warenverkäufe und Käufe zwischen den einzelnen Spielern zu vermitteln.
8. Timeservice
Um allen Agenten eine gemeinsame Zeitrechnung zu ermöglichen, sendet dieser Agent in regelmäßig Abständen einen Zeitimpuls.
9. JXTA
Bietet die Möglichkeit Agenten auch ohne den DF mit Hilfe von JXTA-Methoden zu suchen.

In meiner Gruppe haben wir den Markt als einen eigenständigen Agenten realisiert. Die andere Agenten des Multiagentensystems (z.B. der Manager oder Terra) können beim Marktagenten Kauf- oder Verkaufsangebote einstellen oder nach Angeboten suchen. So müssen sie nicht alle anderen Agenten des Systems nach verfügbaren Angeboten befragen bzw. ihnen neue vermitteln.

3. Aufgaben des Markt-Agenten

Die Hauptaufgabe des Marktagenten besteht darin, Kauf- und Verkaufs-Angebote zwischen den anderen Agenten zu vermitteln. Um diese Aufgabe zu erfüllen bietet der Marktagent mehre Funktionen wie das Einstellen, Entfernen und Suchen von Aufträgen an. Zusätzlich haben wir eine kleine GUI entwickelt, die den aktuellen Datenbestand in einer Tabelle darstellt.

Damit alle anderen Agenten mit dem Markt in Kontakt treten können, registriert sich der Marktagent nach dem Start am zentralen DF. Um Aufträge, die ein bestimmtes Mindestalter überschritten haben wieder aus der Datenbank entfernen zu können, werden die Zeitimpulse des TimeServiceAgenten genutzt. Dieser wird ebendfalls über den DF gesucht.

4. Aufbau und Funktionsweise des Markt-Agent

Der Marktagent ist grundsätzlich passiv aufgebaut. Er verfügt über nur zwei Goals, die in der Regel kurz nach dem Start erfüllt werden sollten.

Das erste Ziel des Agenten ist es, sich am zentralen DF zu registrieren. Hierzu wird der Standardplan DFRegisterPlan benutzt. Es werden für die angebotenen Dienste (Einstellen, Löschen und Suchen) des Marktagenten drei ServiceDescriptions mit einer entsprechenden Beschreibung des Dienstes angelegt. Diese werden dann dem DF während der Anmeldung übermittelt. Da der DF Agenten nach einer bestimmten Zeit wieder aus seinem Datenbestand entfernt, benutzen wir den DFKeepRegisteredPlan um eine dauerhafte Anmeldung sicherzustellen.

Das zweite Ziel ist die Anmeldung des Marktagenten am Timeservice-Agenten. Um dieses Ziel zu erfüllen wird der ServiceSearchPlan benutzt. Dieser erzeugt wiederum ein neues Ziel, welches durch das Finden eines Timeservice-Agenten über den DF erfüllt wird. Wenn ein passender Agent gefunden wurde, erfolgt die Anmeldung mit Hilfe des in der Ontologie festgelegten TimeRegister-Konzeptes. Wenn die Anmeldung erfolgreich ist, erhält der Marktagent in regelmäßigen Abständen den vom TimeServiceAgenten erzeugten Zeitimpuls.

Nachdem diese beiden Goals erfüllt worden sind, reagiert der Agent nur noch auf Nachrichten anderer Agenten und wird selbst nicht mehr aktiv. Den anderen Agenten bietet der Markt-Agent drei unterschiedliche Dienste an.

Jeder Agent kann

1. ein neues Kauf- oder Verkaufs-Angebot in die Datenbank einfügen.
2. eines seiner Angebote wieder entfernen lassen.
3. nach Angeboten anderer Agenten suchen.

Die Suche beschränkt sich dabei im Moment noch auf die Übermittlung des gesamten Datenbestandes (d.h. es wird die gesamte Liste übermittelt und keine Selektion durchgeführt).

Um diese Aufgaben zu erfüllen, verfügt der Marktagent über drei entsprechende Pläne, die nach Empfang einer passenden Nachricht von einem anderen Agenten ausgeführt werden:

1. BrokerAddContractPlan (Fügt ein neues Angebot in die Datenbank ein)
2. BrokerGetContractsPlan (Generiert eine Liste aller vorhandenen Angebote)
3. BrokerRemoveContractPlan (Entfernt Angebote aus der Datenbank)

Die Daten in den Nachrichten liegen bei der Übermittlung nicht einfach als Strings vor, sondern es wird die von einer anderen Gruppe entwickelte Ontologie benutzt. In jeder Nachricht wird ein oder mehrere Contract-Konzepte übermittelt.

Die einzelnen Pläne extrahieren die Contract-Konzepte wieder und führen dann die weitere Bearbeitung durch.

Der BrokerAddContractPlan generiert beim Erhalt eines neuen Contract-Konzeptes eine eindeutige ID, die dann im Contract abgelegt wird. Danach wird das vervollständigte Contract-Konzept an den Absender zurückgeschickt und in der Datenbank abgelegt.

Mit Hilfe des BrokerRemoveContractPlan kann ein Agent eines seiner Angebote wieder aus der Datenbank entfernen lassen. Hierzu übermittelt er das passende Contract-Konzept. Da im Contract eine ID gespeichert ist, kann der Marktagent das gewünschte Objekt in der Datenbank finden und entfernen.

Der BrokerGetContractsPlan ist zur Suche im Datenbestand des Marktagenten vorgesehen. Diesem kann auch ein Contract-Konzept übermittelt werden, um die Suche zu verfeinern. Alle im Datenbestand gefundenen Contract-Konzepte werden einer LinkedList hinzugefügt und dann als Result der Suche an den anfragenden Agenten übermittelt.

5. Das Wissen des Marktagent (Beliefbase)

Der wichtigste Bestandteil der Beliefbase ist eine Datenbank, in der alle bekannten Kauf- und Verkaufs-Angebote abgelegt werden. Die Datenbank selbst ist als Hashtable realisiert und bietet so einfache Zugriffsmöglichkeiten. Um die Contracts leichter zu verwalten, vergibt der Marktagent allen neuen Contracts eine eindeutige ID. Da diese dem einstellenden Agenten als Antwort übermittelt wird, kann der Agent so sein Angebot gegebenenfalls wieder löschen.

Zusätzlich enthält die Beliefbase noch einige weitere Informationen. Hierzu zählen die AID des Timeservice-Agenten (sofern einer beim Start gefunden wurde) und eine Referenz auf die GUI, die für Updates der GUI-Tabelle genutzt wird.

Damit die Datenbank des Markt-Agenten keine alten Angebote enthält, wird der Datenbestand mit Hilfe des DatabaseCleanerPlan

bereinigt. Dieser reagiert auf Nachrichten des Timeservice-Agenten und prüft so jeden "Tag", ob ein Angebot seine Deadline überschritten hat. Wenn dies der Fall ist, wird das Angebot wieder aus der Datenbank gelöscht.

6. Die GUI

Die GUI besteht im wesentlichen aus einem Objekt, welches von JFrame abgeleitet wird und daher ein Fenster öffnet. Das zweite wesentliche Element ist ein TableModel (TableModelContract), welches das Erscheinungsbild und die Eigenschaften der Tabelle bestimmt. Das TableModelContract enthält eine eigene Liste der darzustellenden Contracts. Diese wird bei jedem Ändern der in der Beliefbase enthaltenen Contract-Datenbank aktualisiert und neu dargestellt. Die Aktualisierung erfolgt über eine eigene Methode, die entsprechend von den Plänen BrokerAddContractPlan bzw. BrokerRemoveContractPlan aufgerufen wird.