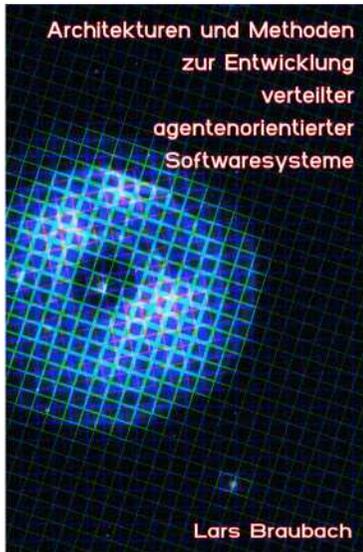


Diese Dissertation ist als Buch erschienen



Architekturen und Methoden zur Entwicklung verteilter agentenorientierter Softwaresysteme

Lars Braubach
ISBN 978-3-00-023107-0
Hardcover, 448 Seiten
Dezember 2007

Druck und Vertrieb über:
Lulu Enterprises Inc. 860
Aviation Parkway
Suite 300
Morrisville, NC 27560
United States of America
www.lulu.com

Inhalt des Buches:

Spezifische Probleme bei der anspruchsvollen und komplexen Aufgabe der Konstruktion verteilter Anwendungen sollen möglichst nicht erst auf Anwendungsebene, sondern soweit wie möglich bereits auf der Basis eines geeigneten softwaretechnischen Entwurfparadigmas adressiert werden. Bereits etablierte allgemeine Softwareentwurfparadigmen, wie z.B. das der Objektorientierung, stellen dazu jedoch bisher nur unzureichende Konzepte bereit.

Deshalb wird in diesem Buch das Agentenparadigma als eine aussichtsreiche Möglichkeit zugrunde gelegt, die konzeptionelle Lücke zwischen den Erfordernissen solcher Anwendungen einerseits und den methodischen Werkzeugen zu deren Konstruktion andererseits geeignet zu überbrücken.

Beschrieben werden Grundlagen von Multiagentensystemen, der Stand der Technik in Bezug auf Agentenarchitekturen, Methoden und Agentenframeworks. Über systematische Anforderungsanalysen werden detaillierte Bewertungen der verfügbaren Lösungen ermittelt und dazu genutzt eine neuartige Agentenarchitektur auf Basis des bewährten Belief-Desire-Intention Modells vorzustellen.

Als praktische Lösung zur Konstruktion von Multiagentensystemen wird das Jadex-Agentenframework eingeführt. Anhand von Beispielanwendungen wird illustriert, wie eine methodische Anwendungsentwicklung mit Jadex durchgeführt werden kann. Damit bietet das Buch einen umfassenden Einstieg in das Thema Multiagentensysteme und vermittelt das notwendige Wissen, um die konzeptionellen Grundlagen, Potentiale und Defizite existierender Agentensoftware richtig einzuschätzen.

Architekturen und Methoden zur Entwicklung
verteilter agentenorientierter Softwaresysteme

Lars Braubach

Zusammenfassung

Die Bedeutung verteilter Systeme nimmt im Kontext aktueller technischer Entwicklungen wie z. B. der Einbeziehung von Alltagsgegenständen in Computernetzwerke stetig weiter zu. Hierbei stellt die Konstruktion verteilter Anwendungen eine anspruchsvolle und komplexe Aufgabe dar, die aufgrund der besonderen Eigenschaften verteilter Anwendungen wie u. a. der hohen Dynamik, Heterogenität und der flexiblen Interaktionen zu neuartigen Problemen führt, die nicht auf Anwendungsebene, sondern vor allem auf der Ebene des zugrunde liegenden softwaretechnischen Entwurfsparadigmas berücksichtigt werden sollten. Etablierte Paradigmen, wie das der Objektorientierung, stellen jedoch nur unzureichende Konzepte bereit. Demgegenüber wird die Agentenorientierung derzeit als eine aussichtsreiche Möglichkeit gesehen, diese konzeptionelle Lücke zu schließen, da sie natürliche Beschreibungsmittel für verteilte Anwendungen anbietet. Trotz der offensichtlichen Vorteile des Agentenparadigmas konnte sich diese Denkweise aufgrund unterschiedlicher Faktoren wie z. B. der Heterogenität des Forschungsfeldes und der damit einhergehenden uneinheitlichen und unausgereiften Ansätze in der Praxis bisher nicht nachhaltig durchsetzen.

In dieser Arbeit wird daher das übergeordnete Ziel verfolgt, einen Beitrag zur softwaretechnischen Nutzbarkeit und Verbreitung des Agentenparadigmas zu leisten. Für die effektive und effiziente Erstellung von Agentenanwendungen werden in dieser Arbeit die methodische Entwicklung und die generische systemtechnische Unterstützung in Form von Agentenframeworks als zentrale Elemente identifiziert. Konzeptionelle Grundlage von Agentenframeworks bilden Agentenarchitekturen, da sie das von Anwendungsentwicklern einsetzbare Agentenbild maßgeblich bestimmen und damit darüber entscheiden, wie gut sich grundlegende Agenteneigenschaften wie Reaktivität und Proaktivität nutzen lassen. Insbesondere die BDI-Architektur weist hier im Vergleich zu anderen Agentenarchitekturen eine Reihe von Vorteilen auf, die es als eine besonders geeignete Ausgangsbasis für Agentenframeworks erscheinen lässt, besitzt jedoch auch noch inhärente Schwächen. Der Kernaspekt der in dieser Arbeit vorgeschlagenen Verbesserungen besteht daher in der Verringerung der konzeptionellen Lücke zwischen dem philosophischen BDI-Modell und dessen softwaretechnischer Interpretation durch die vollständige Umsetzung des zweiphasigen BDI-Schlussfolgerungsprozesses. Zusätzlich wird auf konzeptioneller Seite dafür Sorge getragen, dass auch Erweiterungen in das Modell auf einfache Weise integrierbar sind und die Kernkompetenz der rationalen Verhaltenssteuerung in Zukunft durch weitere Aspekte wie z.B. Emotionen ergänzt werden kann.

Die im Rahmen der Arbeit neu konzipierte BDI-Architektur wird zudem softwaretechnisch verfeinert und innerhalb des ebenfalls im Kontext der Arbeit neu geschaffenen Jadex-Agentenframeworks konkret umgesetzt. Schließlich wird auch darüber berichtet, wie die Praxistauglichkeit des Frameworks und der zugrunde liegenden Architektur durch zahlreiche Einsätze in der akademischen Lehre und innerhalb von Forschungsprojekten praktisch evaluiert und bestätigt werden konnte. Zusätzlich wird sowohl theoretisch als auch anhand konkreter Fallstudien nachgewiesen, dass die methodengestützte und damit systematische Entwicklung mit dem Jadex-Framework im Kontext aktueller BDI-Methoden angemessen unterstützt wird.

Abstract

The importance of distributed systems is steadily increasing especially in the context of actual trends such as the inclusion of everyday objects into computer networks. The construction of such kind of systems is a demanding and complex task that naturally leads to new kinds of problems due to inherent system properties such as the high dynamics, heterogeneity and flexible interactions. These problems should be addressed not on application level but on software paradigm level in order to avoid reinventing generic solutions. In contrast to established paradigms such as object-orientation that do not offer adequate concepts for distributed applications currently agent-orientation is seen as promising approach for closing this conceptual gap as it offers natural description means for distributed applications. Despite the evident advantages of agent orientation this way of thinking could not yet gain sustainable acceptance in practice mainly due to the heterogeneity of the research field and the accompanying inconsistent and insufficiently matured approaches.

Therefore, the primary objective of this dissertation is to contribute to the usability and usage of the agent paradigm for software engineering. For the effective and efficient creation of agent applications two essential influence factors have been identified in this work: the methodological support and the properties of available agent frameworks. The conceptual foundation of agent frameworks are agent architectures as they influence the agent view available for software developers. The agent view in this respect determines to which degree the vital agent characteristics such as reactivity and proactivity can be used. The BDI model of agency exhibits several advantages compared to other agent architectures and makes it an especially promising starting point for agent frameworks. Nevertheless, the BDI architectures conceived so far, also have inherent weaknesses that are one important point for the improvements in this work. The key point of this work consists in the reduction of the conceptual gap between the philosophical view of the BDI model and its software technological interpretation within a new agent architecture that supports the two-phase BDI practical reasoning process. The new architecture additionally ensures with its underlying conceptual and technical extensibility that further aspects such as emotional behaviour control can be integrated easily.

The newly conceived architecture is further concretized for direct software implementation and has been realized within the Jadex agent framework. The suitability for everyday use of the framework and the underlying architecture is demonstrated by numerous usages in the academic as well as in the industrial research field. Additionally it is shown theoretically and practically that a methodological development in the context of actual BDI methodologies of agent applications is supported and promoted by Jadex.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Herausforderungen verteilter Systeme	2
1.2	Konstruktionsansätze für verteilte Systeme	6
1.3	Zielsetzung	9
1.4	Aufbau der Arbeit	11
2	Grundlagen der Multiagentensysteme	13
2.1	Der Agentenbegriff	14
2.2	Softwareagenten	15
2.2.1	Eine grundlegende Agentendefinition	17
2.2.2	Der Agent als graduelles Maß	17
2.3	Intelligente Agenten	19
2.3.1	Definitionen für intelligente Agenten	20
2.3.2	Mentalistische Konzepte	21
2.4	Agentenumgebungen	24
2.4.1	Der Umgebungsbegriff	24
2.4.2	Umgebungseigenschaften	24
2.4.3	Funktionen der Umgebung	26
2.5	Multiagentensysteme	29
2.5.1	Der Begriff des Multiagentensystems	29
2.5.2	Interaktionsformen	29
2.5.3	Koordination	33
2.6	Anwendungsgebiete	40
2.6.1	Industrielle Anwendungsgebiete	41
2.6.2	Kommerzielle Anwendungsgebiete	43
2.6.3	Anwendungsgebiete in der Unterhaltungsindustrie	46
2.6.4	Medizinische Anwendungsgebiete	48
2.6.5	Militärische Anwendungsgebiete	50
2.6.6	Zukunftstrends	52
2.7	Zusammenfassung	55
3	Konstruktion von Multiagentensystemen	57
3.1	Artefakte zur Softwarekonstruktion	58
3.2	Anforderungen	60
3.3	Entwicklungsprozesse	62

3.3.1	Klassifikation von Entwicklungsprozessen	64
3.3.2	Auswahl von Entwicklungsprozessen	66
3.3.3	Vergleiche von Entwicklungsprozessen	67
3.3.4	Anforderungen an Entwicklungsprozesse	68
3.3.5	Gaia	69
3.3.6	Prometheus	75
3.3.7	PASST	82
3.3.8	Tropos	92
3.3.9	Auswertung	101
3.4	Theorien und Architekturen	103
3.4.1	Klassifikation von Theorien und Architekturen	104
3.4.2	Auswahl von Theorien und Architekturen	106
3.4.3	Vergleiche von Theorien und Architekturen	107
3.4.4	Anforderungen an Theorien und Architekturen	108
3.4.5	Die Subsumption-Architektur	109
3.4.6	Das Belief Desire Intention Modell	115
3.4.7	Unified Theories of Cognition	126
3.4.8	Agent-oriented Programming	135
3.4.9	Die 3APL-Agentenarchitektur	140
3.4.10	Das Task-Modell	146
3.4.11	Auswertung	151
3.5	Zusammenhänge zwischen Methoden und Architekturen	152
3.5.1	Beschreibung der Analyse	152
3.5.2	Abbildung von Methoden auf soziale Architekturen	153
3.5.3	Abbildung von Methoden auf interne Architekturen	156
3.6	Zusammenfassung	159
4	Agentenframeworks	163
4.1	Charakteristika von Agentenframeworks	163
4.2	Klassifikation von Agentenframeworks	165
4.2.1	Ad-Hoc Klassifikation	165
4.2.2	Klassifikation nach Merkmalsausrichtung	165
4.3	Auswahl von Agentenframeworks	167
4.4	Vergleiche von Agentenframeworks	169
4.5	Anforderungen an Agentenframeworks	170
4.6	Untersuchungen ausgewählter Agentenframeworks	171
4.6.1	JACK	171
4.6.2	LS/TS	181
4.6.3	JADE	193
4.6.4	Jason	202
4.6.5	Auswertung	210
4.7	Zusammenfassung	212

5	Erweiterung der BDI-Architektur	215
5.1	Anforderungsbeschreibung	216
5.2	Ziele im BDI-Modell	217
5.2.1	Definition des Zielbegriffs	217
5.2.2	Charakteristika von Zielen im BDI-Modell	220
5.2.3	Grundmodell für die Repräsentation und Verarbeitung von Zielen	222
5.2.4	Verfeinerung der Zieltypen	224
5.2.5	Zusammenfassung	235
5.3	Zieldeliberation	236
5.3.1	Interaktionen zwischen Zielen	237
5.3.2	Aspekte von Zieldeliberation	238
5.3.3	Vergleiche von Zieldeliberationsverfahren	240
5.3.4	Die „Easy Deliberation“-Strategie	245
5.3.5	Zusammenfassung	251
5.4	Ein neu konzipierter BDI-Interpreter	251
5.4.1	Existierende Erweiterungen des BDI-Modells	252
5.4.2	Analyse der PRS-Architektur	254
5.4.3	Ein flexibler agendabasierter BDI-Interpreter	256
5.4.4	Umsetzung des grundlegenden BDI-Modells	258
5.4.5	Umsetzung von Erweiterungen	261
5.4.6	Zusammenfassung	264
5.5	Zusammenfassung	265
6	Realisierung der erweiterten Architektur in Jadex	267
6.1	Softwaretechnische Ausgestaltung der PRS-Architektur	268
6.1.1	Agentenspezifikation	268
6.1.2	Wissenselemente	271
6.1.3	Parameterelemente	274
6.1.4	Ziele	275
6.1.5	Pläne	277
6.1.6	Ereignisse	280
6.1.7	Module	282
6.1.8	Zusammenfassung	288
6.2	Integrationsaspekte	288
6.2.1	Ausführungskontextebene	289
6.2.2	Schnittstellenebene	292
6.2.3	Zusammenfassung	296
6.3	Aufbau des Jadex-Frameworks	297
6.3.1	Interpreter	298
6.3.2	Ausführungsumgebungen	300
6.3.3	Werkzeuge	302
6.3.4	Dokumentation	310
6.3.5	Beispielanwendungen	312
6.3.6	Jadex Add-Ons	313
6.3.7	Zusammenfassung	314
6.4	Zusammenfassung	315

7	Anwendungsentwicklung mit Jadex	319
7.1	Systematische Entwicklung von Anwendungen	320
7.1.1	Generisches Evaluierungsframework	321
7.1.2	Analyse für Jadex	322
7.1.3	Zusammenfassung	325
7.2	Anwendungsbeispiele	325
7.2.1	Die Booktrading-Anwendung	326
7.2.2	Das MedPAge-System	346
7.2.3	Weitere Anwendungen und Projekte	370
7.2.4	Zusammenfassung	375
7.3	Bewertung des Jadex-Frameworks	376
7.3.1	Auswertung	383
7.4	Zusammenfassung	385
8	Zusammenfassung und Ausblick	387
8.1	Zusammenfassung	387
8.2	Ausblick	393
	Literaturverzeichnis	397