

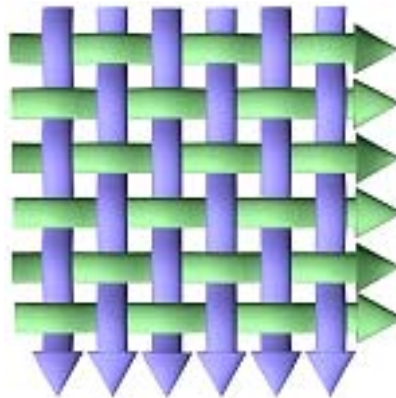


Technical Report 08002

ISSN 1612-1376

Modellierung eines Informationssystems für ein ‚Virtuelles Lager‘

Jörg Zellerhoff



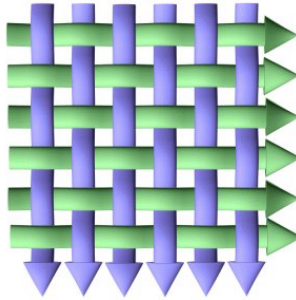
Sonderforschungsbereich 559
Modellierung großer Netze in der Logistik

Universität Dortmund
44221 Dortmund



Sonderforschungsbereich 559

**Modellierung großer
Netze in der Logistik**



Technical Report 08002

ISSN 1612-1376

**Modellierung eines Informationssystems für ein
,Virtuelles Lager‘**

Transferprojekt T01: Modellierung eines ,Virtuellen Lagers‘

Dipl.-Logist. Jörg Zellerhoff

Lehrstuhl für Förder- und Lagerwesen

Emil-Figge-Straße 73

44221 Dortmund

Dortmund, 10. April 2008

Inhalt

1	Einleitung	3
2	Anforderungen an das Logistik-Informationssystem im ‚Virtuellen Lager‘	5
2.1	Funktionale Anforderungen	5
2.2	Technische Anforderungen.....	7
3	Entwicklung des Logistik-Informationssystem mit der PASSI-Methode	9
3.1	Domain Description	9
3.2	Agents Identification	11
3.3	Role Identification	12
3.4	Task Specification.....	16
3.5	Ontology Description	20
3.6	Roles Description.....	21
3.7	Agent Implementation Model	23
4	Ausblick auf die prototypische Umsetzung des Logistik-Informationssystems	31
5	Literatur	32

1 Einleitung

Die effiziente und nachhaltige Nutzung der vorhandenen Ressourcen wird zunehmend zum Synonym für wirtschaftliches und ökologisch sinnvolles Handeln. Im Bereich der Logistik werden diese Ressourcen benötigt, um Leistungsobjekte wie Container, Paletten oder Behälter zur richtigen Zeit am richtigen Ort bereit zu stellen.

In der Verkehrslogistik sind seit einiger Zeit Frachtbörsen im Einsatz, um die Ressource Frachtraum besser auszunutzen [ZEb06]. Verlader und Frachtführer nutzen Internet-Portale, um den Informationsaustausch zwischen Angebot und Nachfrage zu verbessern. So werden Transportaufträge effizienter vergeben und alle Beteiligten können ihre Gewinne erhöhen sowie nebenbei die Verkehrsökologie verbessern.

In der Intralogistik hingegen fehlten bislang außerhalb des Dienstleistungssektors Konzepte, die eine unternehmensübergreifende Nutzung von Ressourcen ermöglichen. Gerade hier zeigt sich aber ein großes Potenzial zur Verbesserung. Lagerkapazitäten eines Unternehmens werden selten konstant ausgenutzt. Als Ursachen sind beispielsweise saisonal bedingte Auftragslagen zu sehen. Aber auch Spekulationsbestände, Saisonwaren und Rabattbestände sind Gründe für Bestandsschwankungen. So entstehen Engpässe im Lager des einen Unternehmens, während ein anderes Unternehmen zur gleichen Zeit, aufgrund einer schlechten Absatzprognose, die Produktion vermindert und seine Bestände senkt. Von einer ökonomisch optimalen Nutzung der Lagerressourcen ist man demnach weit entfernt. Hier setzt das Konzept des Virtuellen Lagers an. Grundgedanke ist der temporäre Zusammenschluss verschiedener Unternehmen zu einem dynamischen Kooperationsverband, in dem die räumlich und organisatorisch getrennten Lagerkapazitäten unternehmensübergreifend genutzt werden können. Dieser Ansatz darf nicht mit einem mandantenfähigen Lager verwechselt werden, da dort zwar ebenfalls Güter unterschiedlicher Unternehmen unter einem Dach gelagert werden, aber langfristige Verträge über fixe Lagervolumen dem Gedanken des Virtuellen Lagers widersprechen.

Im Virtuellen Lager können Mitgliedsunternehmen bei einer geringen Auslastung ihrer Lagerkapazitäten Lagerraum temporär für Kooperationspartner freigeben (Anbieter). Andere Unternehmen, deren eigene Lagerkapazitäten sich für den Freigabezeitraum als zu klein erweisen, können diesen Lagerraum nutzen (Nachfrager). Das Prinzip unternehmensübergreifender Nutzung von Lagerraum in einem großen logistischen Netzwerk führt bei den Teilnehmern zu einer deutlich gleichmäßigeren und höheren Auslastung der Ressource „Lager“ und ist damit wesentlich effizienter. Dieser Effekt ist mit dem Potenzial einer chaotischen Lagerplatzvergabe im Gegensatz zur Festplatzvergabe in einem geschlossenen Lagersystem zu vergleichen [ZJo08]. Dabei weist die Idee des Virtuellen Lagers grundsätzlich keine Restriktionen hinsichtlich Branchen oder Güter auf. Maßgebend sind nur die aktuellen Angebote und Nachfragen. Allerdings ist eine gemeinsame eindeutige „Sprache“ aller Kooperationspartner für nachgefragte und angebotene Leistung notwendig. Der Nachfrager muss genau wissen, welche Leistung vom Anbieter zur Verfügung gestellt wird und ob diese Leistung zu seiner Nachfrage passt. Wenn palettierte Ware bei einem Kooperationspartner gelagert werden soll, können trotz des definierten Ladungsträgers ganz unterschiedliche Lagerprozesse erforderlich sein. Beispielsweise sind Elektroartikel vor Witterungseinflüssen zu schützen oder Tiefkühlware bei einer bestimmten Temperatur zu lagern. Bestimmte Warengruppen brauchen also auf sie zugeschnittene Lagerprozesse. Um die Kommunikation zwischen Anbieter und Nachfrager zu vereinfachen und fehlerhafte Prozesse zu vermeiden, muss im Virtuellen Lager beim Erstellen von Angebot oder Nachfrage auf eine Bibliothek von Standardprozessen zurückgegriffen werden.

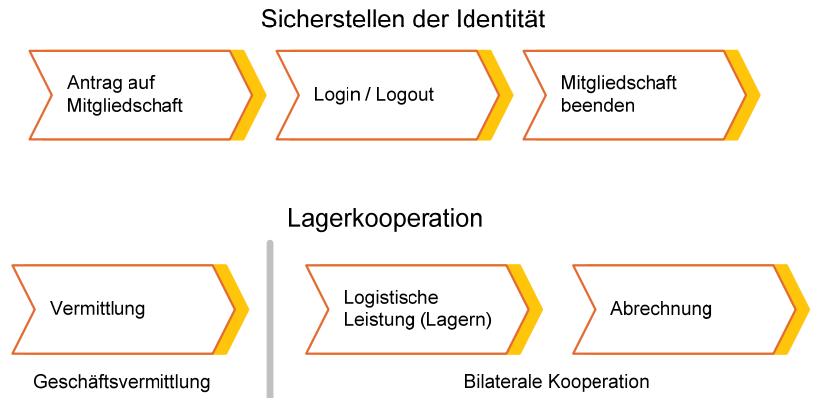


Abbildung 1: Geschäftsmodell des ‚Virtuellen Lagers‘

Abbildung 1 zeigt exemplarisch den Ablauf der Geschäftsprozesse im Virtuellen Lager. Um Zuverlässigkeit und Authentizität der Teilnehmer zu gewährleisten, muss sich zunächst jedes interessierte Unternehmen auf einer Web-Plattform registrieren. Für die Lagerkooperation, welche das eigentliche Ziel des ‚Virtuellen Lagers‘ darstellt, sind die Geschäftsprozesse „Vermittlung“, „Logistische Leistung“ und „Abrechnung“ notwendig. Die Lagerkooperation kann in zwei Phasen unterteilt werden. In der ersten Phase findet die Geschäftsvermittlung statt. Die zweite Phase beschreibt die bilaterale Kooperation zweier Mitglieder.

Generell müssen die Geschäftsprozesse durch ein Informationssystem unterstützt werden. Dieses Informationssystem soll als Bindeglied zwischen den Unternehmen im Kooperationsverband des ‚Virtuellen Lagers‘ fungieren. Die Entwicklung eines solchen LIS (Logistik-Informationssystem) ist Gegenstand dieses Technical Reports und soll im Folgenden genau beschrieben werden. Zunächst sind die Anforderungen an das LIS aus funktionaler und technischer Sicht zu definieren. In einem zweiten Schritt wird PASSI (a Process for Agent Societies Specification and Implementation, siehe auch [BCo02]) als Methode zur systematischen Entwicklung eines Multi-Agenten-Systems verwendet.

2 Anforderungen an das Logistik-Informationssystem im ‚Virtuellen Lager‘

2.1 Funktionale Anforderungen

Die Unified Modeling Language (UML) bietet mit dem Anwendungsfall-Diagramm eine in der Softwaretechnik weit verbreitete, grafische Darstellungsmöglichkeit der funktionalen Anforderungen an das LIS. UML dient zur Modellierung eines Problembereichs und ermöglicht das Spezifizieren, Konstruieren, Visualisieren und Dokumentieren von Softwaresystemen. UML ist als Modellierungssprache allgemein anerkannt und bietet die Möglichkeit, ein Modell der zu erstellenden Software intuitiv aufzubauen.

Die Abgrenzung zwischen dem hier beschriebenen Virtuellen Lager und anderen publizierten Realisierungen zur Bildung eines virtuellen Lagers besteht in der temporären und dynamischen Form der Kooperation. Neben dem Potenzial einer solchen „lockeren“ Kooperation hat die geringe Bindungsintensität auch eine negative Seite. Diese verdeutlicht sich im Misstrauen zwischen den fluktuierenden Mitgliedern, die während ihrer kurzen Kontakte nicht genügend Zeit haben, ein Vertrauensverhältnis aufzubauen. Kompensieren lässt sich das Misstrauen durch Vertrauen in das System selber. Deshalb ist bei der Entwicklung des Virtuellen Lagers eine auf Transparenz und Sicherheit basierende Lösung von Bedeutung. Abbildung 2 zeigt die Anwendungsfälle, welche im Bereich des Geschäftsprozess der Anmeldung/Abmeldung identifiziert werden können. Dieser Geschäftsprozess soll vor allem Sicherheit hinsichtlich der Identität der Kooperationspartner gewährleisten. Hierfür ist vor der ersten Nutzung des Logistik-Informationssystems von dem jeweiligen Anwender ein Antrag auf Mitgliedschaft zu stellen. Der Anwender stellt die übergeordnete Instanz der LIS-Nutzer da, welche sich in Anbieter und Nachfrager unterteilen lässt. Wenn das LIS dem „Antrag auf Mitgliedschaft“ zustimmt, wird für den Anwender ein Account angelegt. Über diesen Account muss sich der Anwender bei Programmstart anmelden (login) und zum Beenden seiner Arbeiten wieder abmelden (logout). Diese Prozesse sind notwendig, um innerhalb des Informationssystems die unterschiedlichen Nutzer identifizieren zu können. Schlussendlich kann die Mitgliedschaft im Virtuellen Lager auch beendet werden, wenn keine weitere Kooperation im Bereich der Lagerkapazitäten vorgesehen ist.

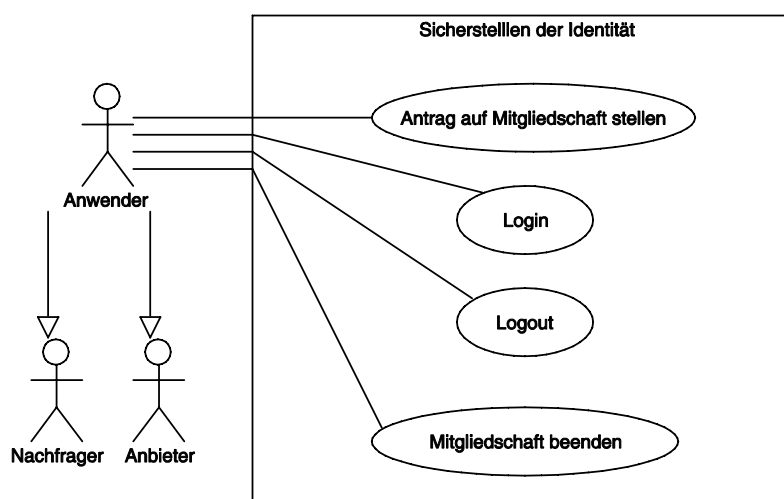


Abbildung 2: Geschäftsprozess „Sicherstellen der Identität“

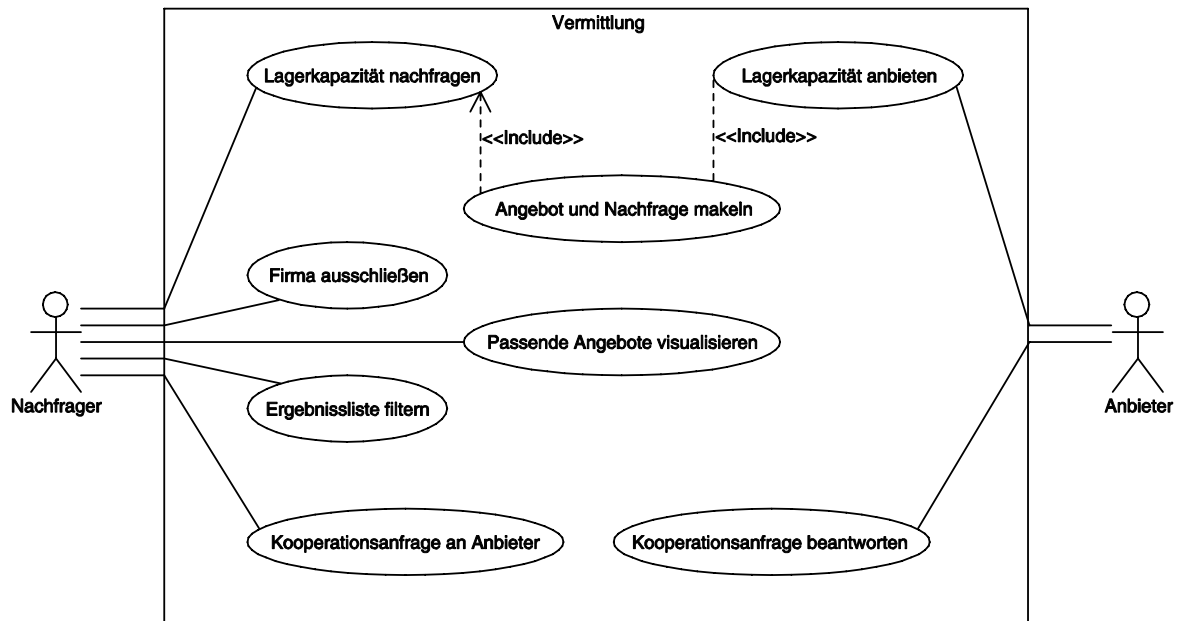


Abbildung 3: Geschäftsprozess "Vermittlung"

Als besondere Herausforderung für das Logistik-Informationssystem ist die Zuordnung zwischen Angebot und Nachfrage zu sehen. Der Geschäftsprozess der Vermittlung beschreibt diesen Vorgang. Abbildung 3 stellt die funktionalen Anforderungen aus Anwendersicht für diesen Geschäftsprozess dar. Die Anwender im LIS können Angebote oder Nachfragen zu Lagerkapazitäten erstellen. Hierzu müssen dem einzelnen Angebot bzw. der Nachfrage Mengen- und Zeitangaben hinzugefügt werden. Weiterhin ist aus der Bibliothek eine Standardprozesskette zu wählen, welche bei einer Nachfrage die geforderte Behandlung der Güter beschreibt. Der Anbieter hingegen kann unterschiedliche Prozessketten mit verschiedenen Serviceoptionen anbieten, die dann vom Nachfrager gewählt werden können. Nachdem für Nachfrage oder Angebot die erforderlichen Daten an die Software übergeben wurden, kann das LIS dem Nachfrager ein passendes Angebot suchen. Entscheidungsbedarf entsteht, wenn beispielsweise zu einer Nachfrage mehrere passende Angebote bestehen. Über die Frage, welches Angebot für einen Vertragsabschluss gewählt werden soll, kann dann der Preis, die geografische Lage oder das Vertrauensverhältnis der Angebots-Unternehmen zu dem Nachfrage-Unternehmen entscheiden. Entsprechend kann der Nachfrager seine Ergebnisliste nach verschiedenen Kriterien filtern. Hat der Nachfrager sich für ein Angebot entschieden, so schickt er eine Kooperationsanfrage an den Anbieter. Dieser muss jetzt seinerseits bestätigen, dass er die Kooperation eingehen möchte. Für den Fall, dass ein Nachfrager nicht mit einem bestimmten Anbieter zusammen arbeiten will, bietet das LIS die Möglichkeit einzelne Firmen auszuschließen. Somit werden Angebote entsprechender Firmen nicht in die Ergebnislisten aufgenommen.

In dem nachfolgenden Geschäftsprozess der „logistischen Leistung“ wird die Koordination und Informationsbereitstellung zwischen Nachfrager und Anbieter unterstützt. Der Nachfrager ist für den Transport der Güter zum Anbieter verantwortlich. Der Anbieter muss die Warenannahme, sowie sämtliche Lagerprozesse bis zur Warenabgabe organisieren und durchführen. Als große Herausforderung stellen sich hier die Terminabsprachen dar, welche zwischen den Warentransporten auf Nachfragerseite und der Warenannahme bzw. Warenbereitstellung auf Anbieterseite getroffen werden müssen (siehe Abbildung 4). Hierzu

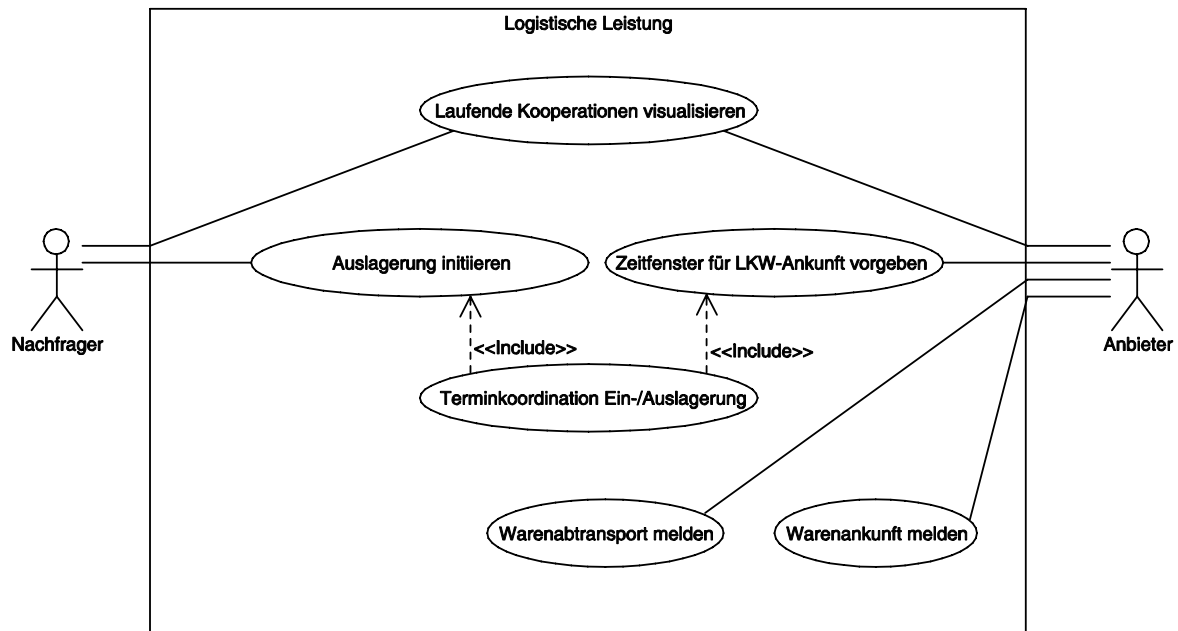


Abbildung 4: Geschäftsprozess "Logistische Leistung"

wird eine Funktion zur Terminabsprache vorgesehen, wie sie beispielsweise in MS Outlook integriert ist. So können beide Seiten die genaue Ankunftszeit bestimmter Güter vereinbaren. Erhöht wird die Komplexität dieser Koordinationsaufgabe, wenn Nachfrager und Anbieter sich einigen, dass der Abtransport auch in Teilmengen erfolgen kann. Die komplexere Koordination wird hier durch eine deutlich größere Flexibilität hinsichtlich der gelagerten Güter gerechtfertigt. Für diese Koordinationsaufgaben, aber auch zur Aktualisierung des Warenwirtschaftssystems des Nachfragers, können sowohl Anbieter als auch Nachfrager den Status der laufenden Kooperationen einsehen. Für die Terminkoordination selber wird bei Ein- und Auslagerung ein Zeitfenster für die LKW-Ankunft durch den Anbieter vorgegeben. Dabei ist das Anlieferungsdatum durch den Kooperationsvertrag festgelegt. Bei einer Auslagerung vor Ende des Kooperationsvertrages teilt der Nachfrager dem Anbieter den gewünschten Tag der Warenabholung mit und bekommt dann ein entsprechendes Zeitfenster vom Anbieter vorgegeben.

Im Geschäftsprozess der „Verrechnung“ initiiert zum Abschluss eines Kooperationsvertrages der Anbieter die Rechnungsstellung für die durchgeführte Dienstleistung (siehe Abbildung 5). Die Rechnungshöhe ergibt sich aus der Anzahl der gemieteten Lagerplätze, der Mietdauer und der zugrundeliegenden Prozesskette für die Handhabung der Ware. Der Nachfrager kann sich im LIS offene Rechnungen anzeigen lassen und diese bezahlen.

2.2 Technische Anforderungen

Neben den dargestellten funktionalen Anforderungen bestehen auch eine Reihe von technischen Anforderungen an das Logistik-Informationssystem. Das Beherrschen der Komplexität ist erforderlich, denn Angebot und Nachfrage einer Vielzahl an Teilnehmern müssen unter Optimierungskriterien zugeordnet werden. Da Angebot und Nachfrage zuverlässig innerhalb vorgegebener zeitlicher Grenzen zu bearbeiten sind, muss die Kommunikationsgeschwindigkeit in dem System sich diesen Anforderungen stellen.

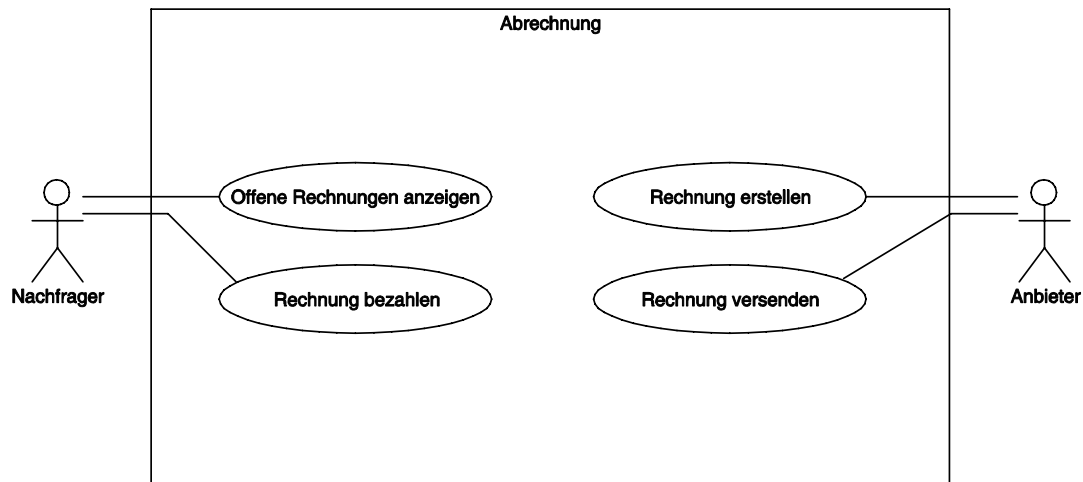


Abbildung 5: Geschäftsprozess "Abrechnung"

Zunächst ist eine Entscheidung zwischen einem zentralen oder einem verteilten System zu treffen. Bei einem zentralen System wird eine Datenbank alle Angebote, Nachfragen und Kooperationen speichern. Ein Programm wird versuchen jeder Nachfrage mögliche Angebote zuzuordnen. Dieser als „makeln“ bezeichnete Prozess (siehe auch Abbildung 3) erfordert bei steigender Zahl an Teilnehmern und einer Vielzahl an Restriktionen, wie Mengenangaben, Kooperationstermine, Prozessketten, ausgeschlossene Firmen, etc. eine hohe Rechenintensität. Außerdem müsste auch die vorgesehene Optimierung der Ressourcenallokation durch eine zentrale Instanz erfolgen. Hierbei soll die Zuordnung von Angebot und Nachfrage nicht alleine durch den Nachfrager auf Grundlage der zum Zeitpunkt der Kooperationsanfrage bestehenden Angebote getätigt werden. Bei dieser Optimierung ist mit Hinblick auf einen bestmöglichen Gesamtnutzen in der Zeit zwischen der Kooperationsanfrage und der tatsächlichen Wareneinlagerung immer wieder zu prüfen, ob die momentane Allokation von Nachfragern und Anbietern geändert werden muss. Da im Virtuellen Lager verschiedene Akteure jeweils eigene Interessen verfolgen, wird eine zentrale Instanz bei einer Optimierung immer einige Akteure benachteiligen, während es andere bevorzugt. Auf einen Verkehrsstau übertragen, wird die Problematik besser verständlich. Was wäre, wenn eine zentrale Steuerungsinstanz vor Staus auf den Autobahnen schützt? Sie wird nur so viele Autos über die Autobahn führen, dass sich kein Stau bildet. Der Rest der Autofahrer muss die längere Umfahrung über Landstraßen in Kauf nehmen. Damit wird zwar das Gesamtsystem optimal arbeiten, der Autofahrer kann aber nicht mehr für sich entscheiden und würde entweder bevorteilt (Autobahn) oder benachteiligt (Landstraße). Das Beispiel zeigt das Unvermögen einer zentralen Steuerungsinstanz, alle Akteure gerecht zu behandeln. Die Alternative bietet hier ein dezentrales Steuerungssystem. Auf das Verkehrsstau-Beispiel übertragen, entscheidet jeder Autofahrer selbst über seine Routenwahl. Wenn im Autoradio ein Stau gemeldet wird, kann der Fahrer alleine entscheiden, ob er eine Umfahrung für besser hält oder den Stau. Die Praxis zeigt, dass sich so bei guter Informationsverfügbarkeit ein Gesamtzustand bildet, der nahe dem Optimum liegt. Neben dieser Selbstorganisation spricht aber auch das Beherrschen eines komplexen Netzwerkes für ein dezentrales System. Hier ist die Leistungsanforderungen an einzelne Komponenten viel geringer als bei einer zentralen Recheneinheit, da sich die Rechenintensität auf die einzelnen Knoten im Netzwerk verteilt.

3 Entwicklung des Logistik-Informationssystem mit der PASSI-Methode

Die bei der Entwicklung des Logistik-Informationssystem verwendete PASSI-Methode bietet ein schrittweises, in Phasen unterteiltes Vorgehensmodell zur Entwicklung von Multi-Agenten-Systemen [PAS08]. PASSI integriert dafür Konzepte der objektorientierten Software-Entwicklung und verwendet die bereits etablierte Entwurfssprache Unified Modeling Language (UML). Die in der Forschung zu Künstlicher-Intelligenz definierten Eigenschaften von Agenten sind bei PASSI berücksichtigt [EYM03]:

- autonom (ein Agent arbeitet unabhängig von Benutzereingriffen),
- proaktiv (ein Agent löst Probleme aufgrund eigener Initiative),
- reaktiv (ein Agent reagiert auf Änderungen seiner Umgebung),
- sozial (ein Agent kommuniziert mit anderen Agenten) und
- zielorientiert (ein Agent versucht ein vom Programmierer oder Benutzer vorgegebenes Ziel zu erreichen).

PASSI begleitet den Entwickler während des gesamten Softwareentwicklungsprozesses bis hin zur Implementierung. Im Softwareentwicklungsprozess schreibt PASSI eine strikte Reihenfolge der zu nutzenden Methoden vor, wobei wichtige Konzepte von Multi-Agenten-Systemen wie Rollen und Kommunikation einbezogen werden. Abbildung 6 zeigt das Vorgehensmodell von der „Domain Description“ bis zur „Code Production“ und weiter nach Test der Agenten bis zur „Deployment Configuration“.

3.1 Domain Description

Im ersten Schritt sieht PASSI die Entwicklung des „System Requirements Models“ vor. Hierbei handelt es sich um ein Modell der Systemvoraussetzungen in Hinblick auf beteiligte

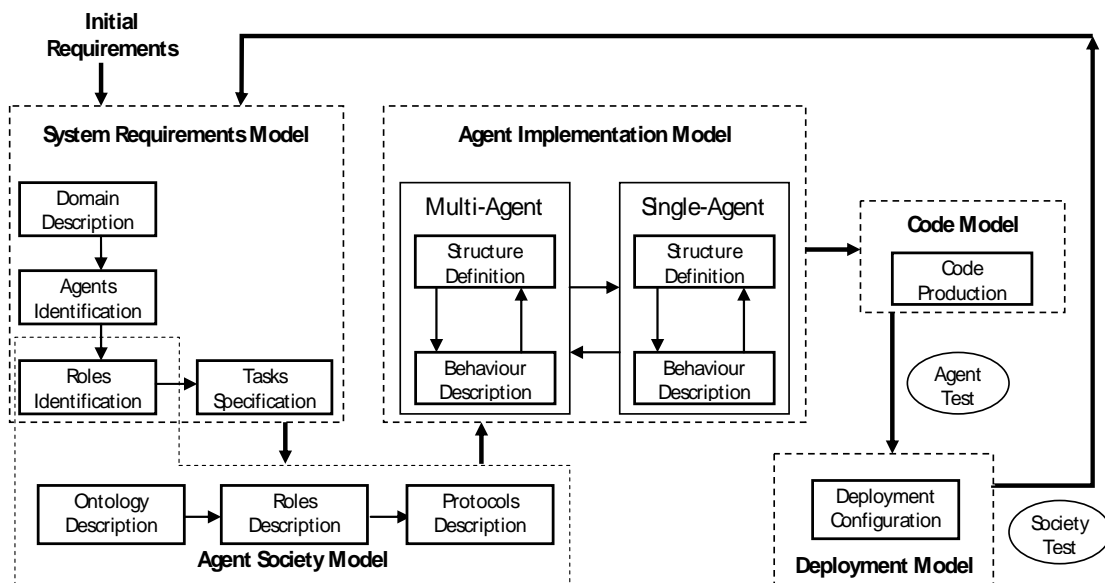


Abbildung 6: The PASSI methodology [PAS08]

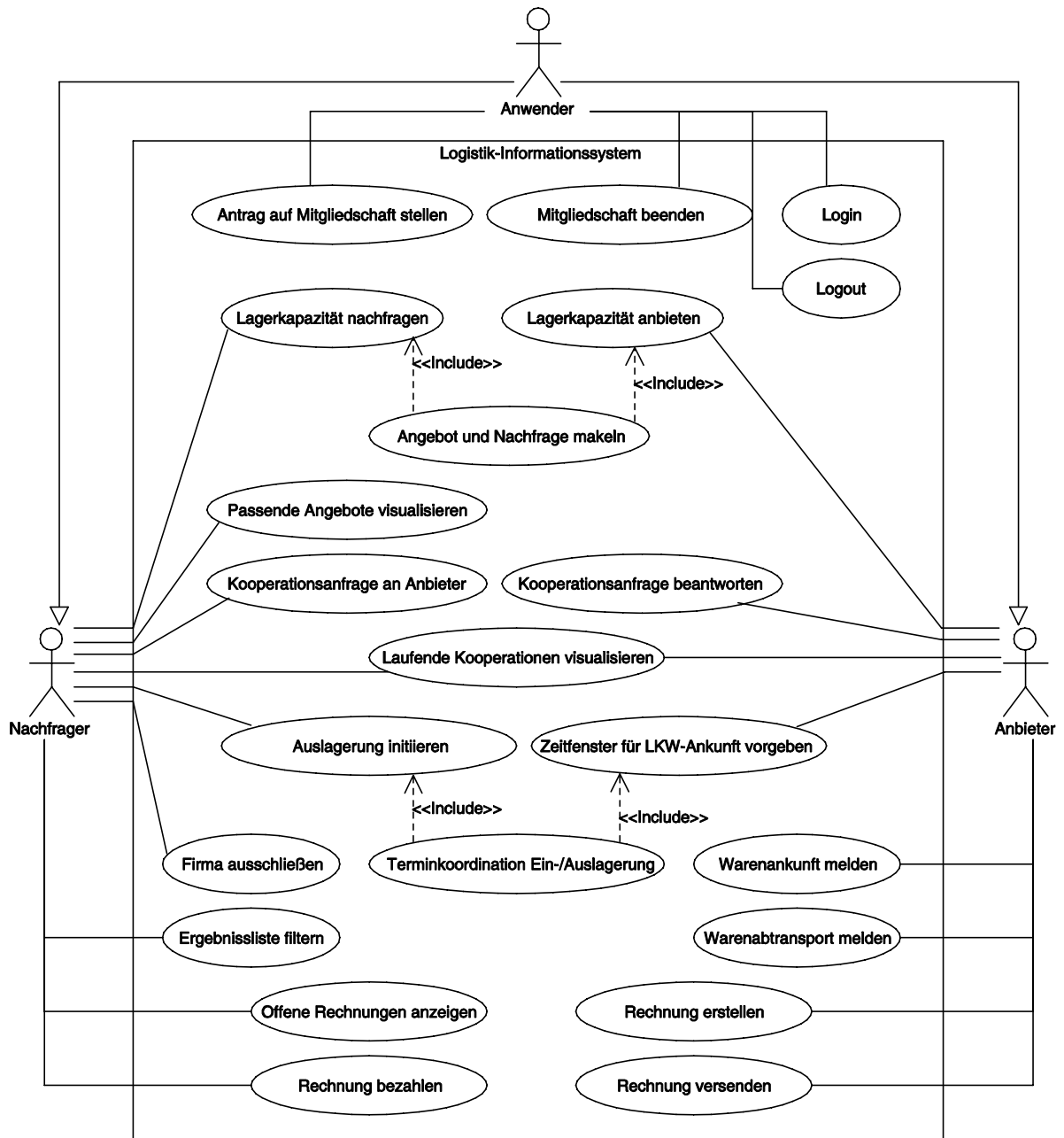


Abbildung 7: Domain Description Diagram

Aktuere und deren Ziele. Dazu wird in der ersten Phase des „System Requirements Models“ die „Domain Description“ konzipiert [CPo02]. Abbildung 7 zeigt das „Domain Description Diagram“, welches eine Zusammenfassung der funktionalen Beschreibung des Systems enthält, die bereits in den Anwendungsfall-Diagrammen in Abbildung 2 bis Abbildung 5 dargestellt sind.

3.2 Agents Identification

In der zweiten Phase des „System Requirements Models“ wird die „Agents Identification“ vorgenommen. Nach der Definition von Agenten, können diese als eine Geschäftsprozess oder eine Gruppe von Geschäftsprozessen gesehen werden. Deshalb werden ausgehend von der „Domain Description“ alle Geschäftsprozesse erfasst und funktionalen Gruppen zugeordnet. So wird eine agentenbezogene Gruppierung der funktionalen Anforderungen erreicht (siehe Abbildung 8).

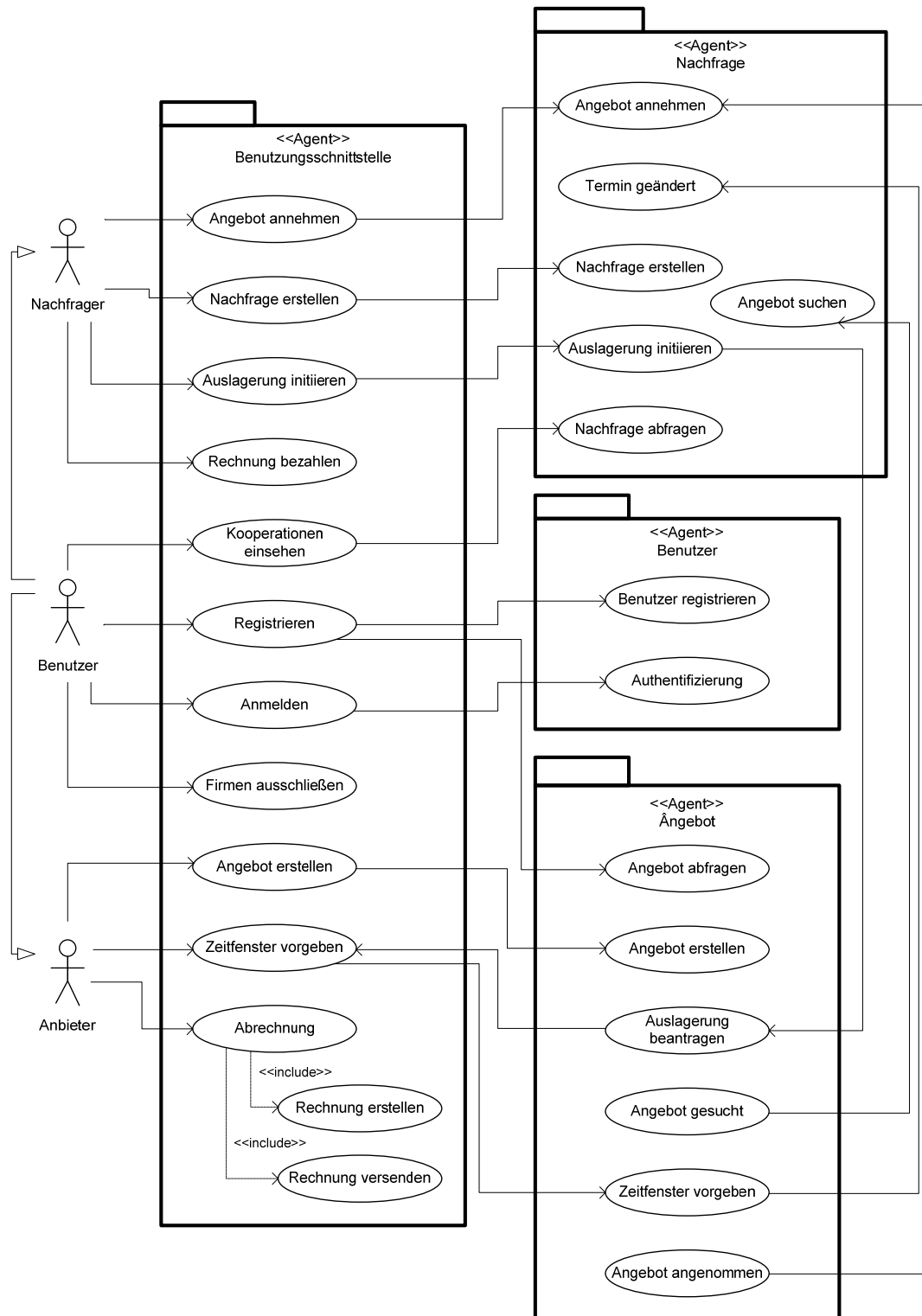


Abbildung 8: Agent Identification Diagram

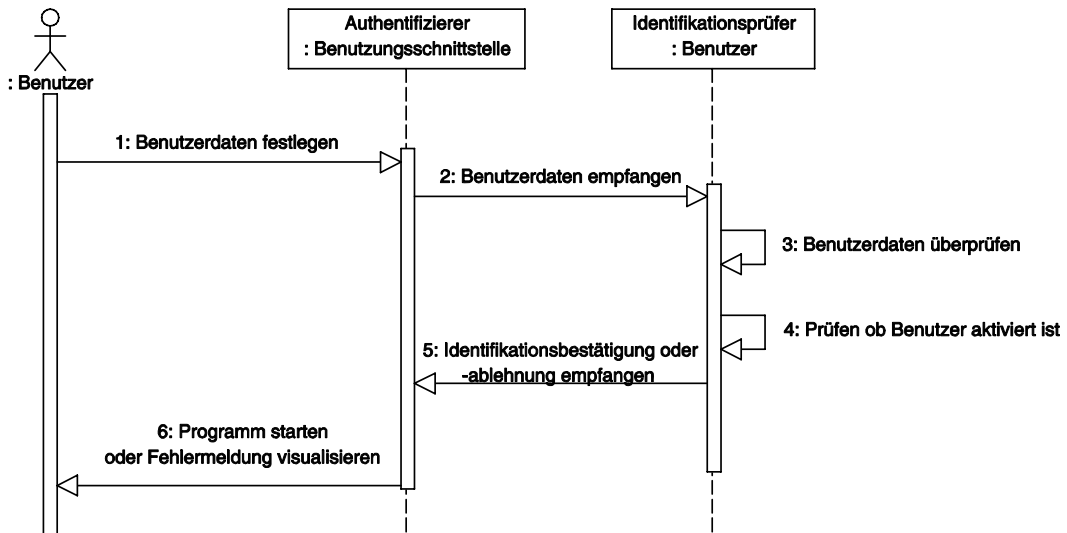


Abbildung 9: Role Identification Diagram - Anmelden

3.3 Role Identification

Die „Role Identification“ wird sowohl für das „System Requirements Model“, wie auch für das „Agent Society Model“ benötigt. Dies ist eine logische Konsequenz des Versuchs, die aus früheren Phasen ableitbaren Szenarien hier zu untersuchen. Die „Role Identification“ bietet zum einen eine Beschreibung von Funktionalität und Verhalten der Agenten, die dem „System Requirements Model“ zuzuordnen ist. Zum anderen werden die Verbindungen der Agenten untereinander beschrieben, was in den Bereich des „Agent Society Model“ fällt. Die wichtigsten Szenarien, die eine Kommunikation zwischen Agenten enthalten, werden mittels der aus UML bekannten Sequenzdiagramme dargestellt. Dazu werden die einzelnen Rollen beschrieben, die ein Agent während der Laufzeit einnehmen kann. Hierbei kann ein Agent sowohl mehrere Rollen in einem Szenario, als auch eine Rolle in verschiedenen Szenarien einnehmen.

Role Identification – Anmelden (Abbildung 9): Der Benutzer gibt seinen Benutzernamen und sein Passwort ein. Diese werden direkt an den Identifikationsprüfer gesendet, welcher überprüft, ob die eingegebenen Daten korrekt sind und ob der Benutzer aktiviert ist. Der Identifikationsprüfer sendet dann eine Bestätigung oder eine Fehlermeldung an den Benutzer.

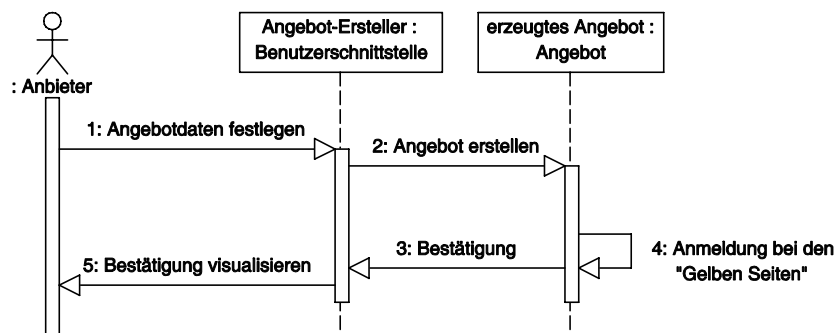


Abbildung 10: Role Identification Diagram - Angebot erstellen

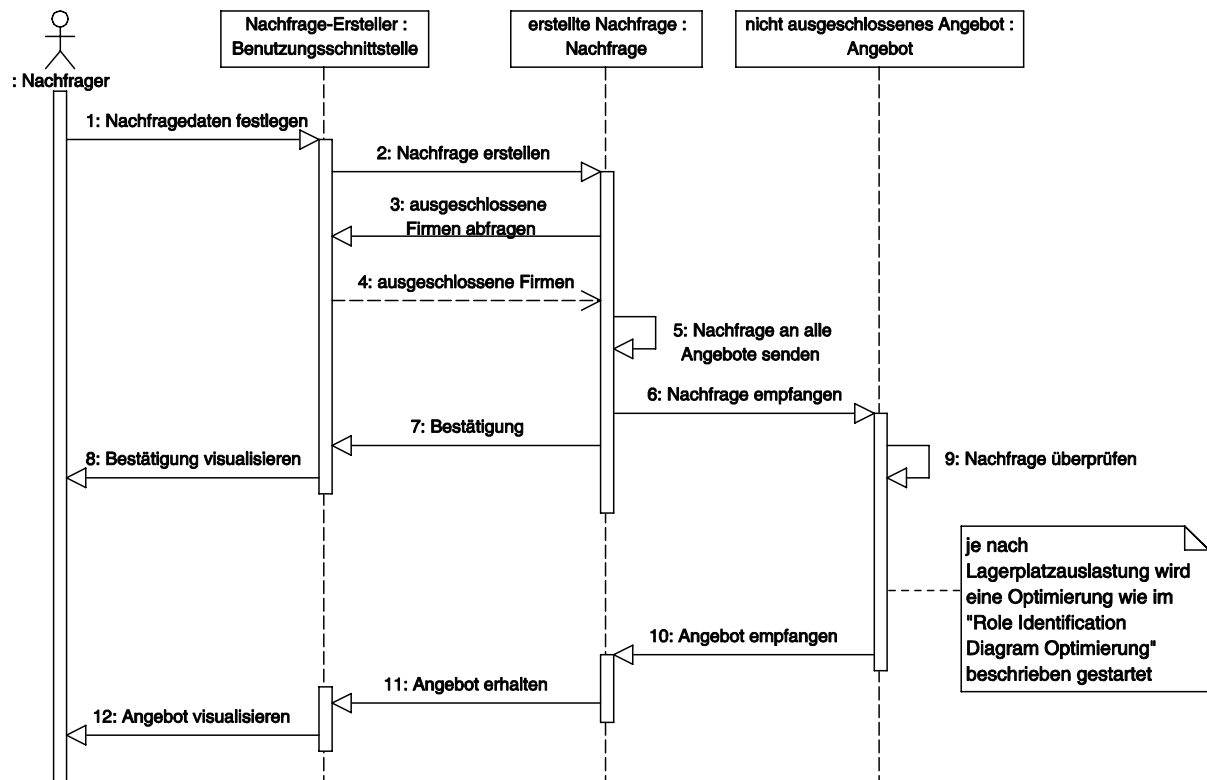


Abbildung 11: Role Identification Diagram - Nachfrage erstellen

Role Identification – Angebot erstellen (Abbildung 10): Der Anbieter gibt die relevanten Angebotsdaten ein. Aus diesen Daten wird das Angebot erzeugt und es erfolgt eine Bekanntmachung des Angebots auf den „Gelben Seiten“. Der Anbieter erhält eine visualisierte Bestätigung.

Role Identification – Nachfrage erstellen (Abbildung 11): Der Nachfrager sendet die Daten seiner Nachfrage (Beginn, Ende, Anzahl, Prozesskette) an die Benutzungsschnittstelle. Diese erstellt einen neuen Nachfrage-Agenten, der zunächst die ausgeschlossenen Firmen bei der Benutzungsschnittstelle abfragt. Dann schickt der erstellte Agent seine Nachfrage an alle nicht ausgeschlossenen Angebots-Agenten. Er meldet der Benutzungsschnittstelle, dass er erfolgreich gestartet wurde. Diese leitet die Bestätigung an den Nachfrager weiter. Danach wartet der Nachfrage-Agent auf Antworten der Angebots-Agenten. Er meldet einkommende Angebote an die Benutzungsschnittstelle. Diese visualisiert dem Nachfrager die verschiedenen Angebote.

Role Identification – Angebot annehmen (Abbildung 12): Der Nachfrager wählt dasjenige Angebot aus, welches er annehmen will. Der aktuelle Nachfrage-Agent sendet dann an den Agenten des ausgewählten Angebots die Nachricht, dass das Angebot angenommen wurde. Der Angebots-Agent prüft, ob die Nachfrage zum Angebot passt und ob genügend freie Kapazitäten vorhanden sind. Je nach Prüfungsergebnis sendet er eine Fehlermeldung oder eine Bestätigung an den Nachfrager. Im Falle einer Bestätigung wird die Terminabsprache gestartet. Der vom Anbieter festgelegte Liefertermin wird an den Nachfrager gesendet.

Role Identification – Optimierung (Abbildung 13): Die Optimierung begründet sich durch die ständig veränderte Marktsituation von Angebot und Nachfrage und baut auf bereits bestehenden Kooperationsverhältnissen auf. Sie wird immer von einem Angebots-Agenten initialisiert und zwar in dem Fall, dass diesem Angebot bereits eine Nachfrage zugeordnet ist und ohne diese Zuordnung die Annahme der aktuellen Nachfrage möglich wäre, wobei die

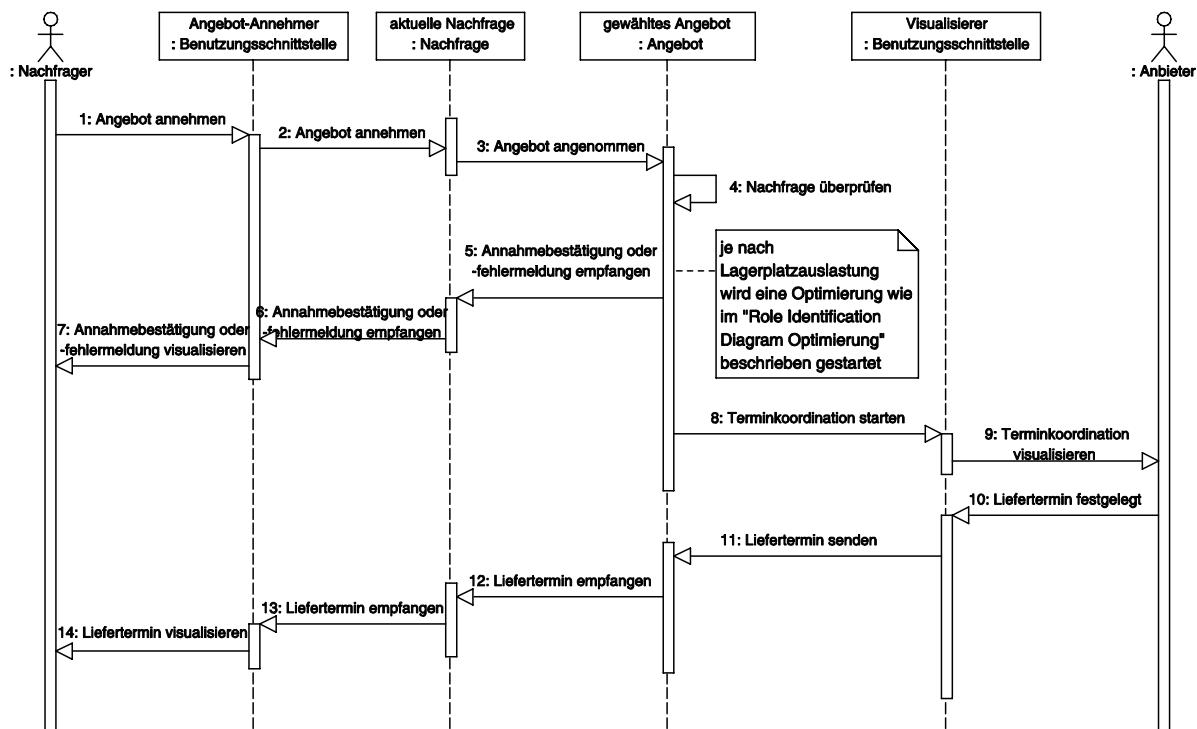


Abbildung 12: Role Identification Diagram - Angebot annehmen

Annahme der aktuellen Nachfrage einen höheren Nutzen ergeben muss. Für einen höheren Nutzen gibt es zwei denkbare Fälle. Entweder wird ein Ansteigen der durchschnittlichen Anzahl vermieteter Lagerplätze über den gesamten Angebotszeitraum erreicht oder bei gleicher durchschnittlicher Anzahl an vermieteten Plätzen weniger Kooperationspartner (Nachfrager). Tritt einer dieser Fälle auf, so ist es besser für den Anbieter die aktuelle Nachfrage anzunehmen und die bereits zugesagte Kooperation mit der vorhandenen Nachfrage aufzulösen. Allerdings kann der Anbieter den bestehenden Kooperationsvertrag nicht auflösen. Er hat nur die Möglichkeit die vorhandene Nachfrage an einen anderen Anbieter abzutreten, wenn sich dadurch der Nutzen des Nachfragers erhöht. Die Erhöhung des Nutzens kann zum Beispiel durch eine geringere Entfernung zum neuen Anbieter entstehen. Alternativ kann der alte Anbieter dem Nachfrager auch monetäre Mittel zukommen lassen, um eine eventuell entstehende Nutzenverringerung durch einen neuen Anbieter zu kompensieren.

Role Identification – Auslagerung initiieren (Abbildung 14): Der Nachfrager wählt auf der Benutzeroberfläche die Option eine Auslagerung zu initiieren und gibt das gewünschte Datum der Auslagerung ein. Der Nachfrage-Agent sendet daraufhin eine Nachricht an den gewählten Angebots-Agenten. In dieser Nachricht informiert er den Angebots-Agenten über die beantragte Auslagerung. Der Angebots-Agent führt eine Terminabsprache mit dem Anbieter durch und sendet danach den festgelegten Termin an den Nachfrageagenten.

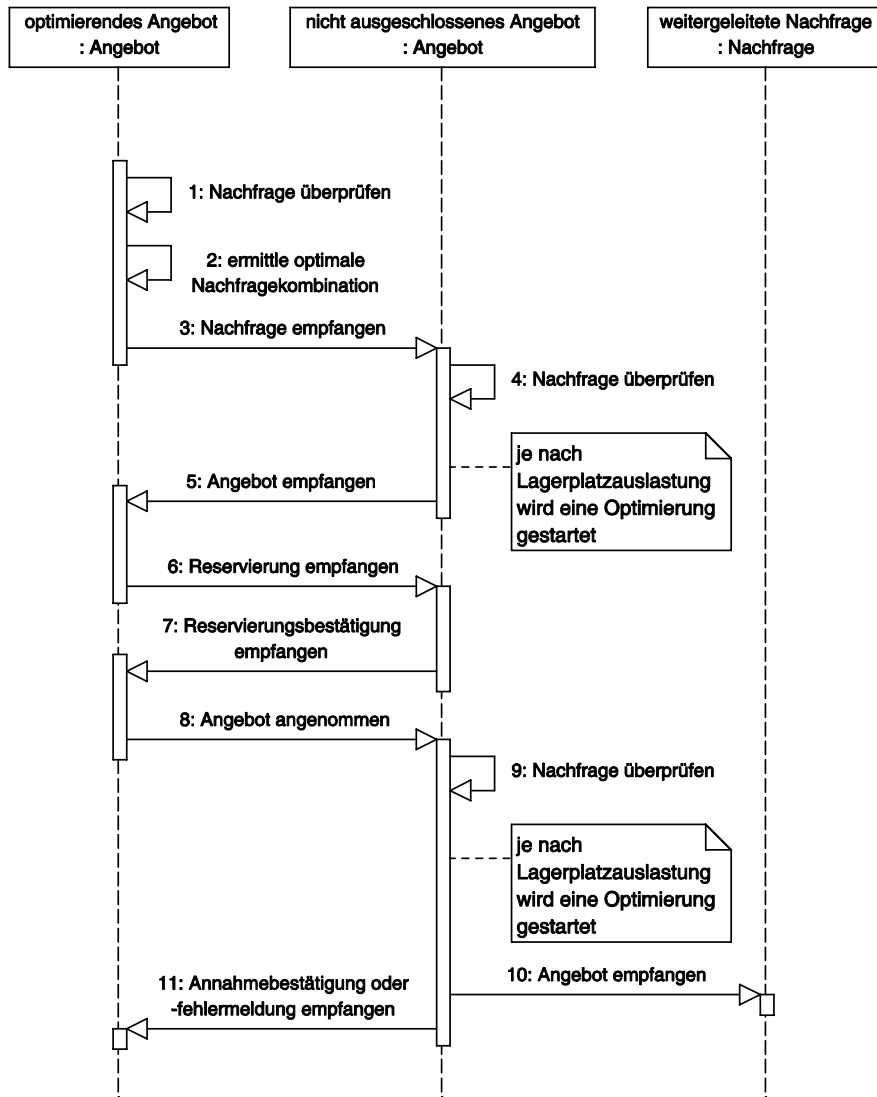


Abbildung 13: Role Identification Diagram - Optimierung

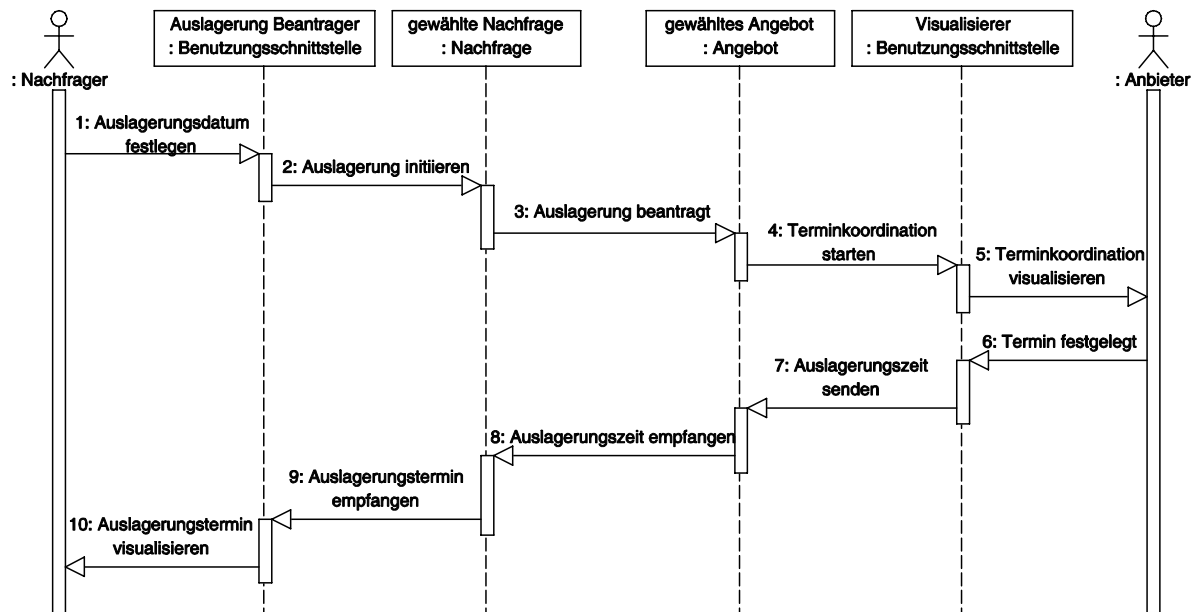


Abbildung 14: Role Identification Diagram - Auslagerung initiieren

3.4 Task Specification

In der „Task Specification Phase“ wird für jeden Agenten ein Aktivitäts-Diagramm erstellt. Ziel dieser Phase ist die Definition der Aufgaben, welche zur Realisierung der in den vorherigen Phasen beschriebenen Funktionalitätsanforderung notwendig sind. Ein Agent besteht dabei aus mehreren Tasks, welche jeweils ein Verhalten des Agenten beschreiben. Jedes „Task Specification Diagramm“ ist in zwei Bereiche geteilt, welche durch eine vertikale Linie getrennt sind. Auf der rechten Seite werden die Aufgaben des jeweiligen Agenten dargestellt. Auf der linken Seite wird die Interaktion mit anderen Agenten gezeigt und somit das Verhältnis der Agenten untereinander definiert. Ein „Task Specification Diagramm“ enthält alle Funktionen eines Agenten, wobei die Rollen des Agenten vernachlässigt werden. Eine Transition zwischen Tasks kann entweder die Kommunikation zwischen verschiedenen Tasks des Agenten oder den Nachrichtenaustausch zwischen einem Task des Agenten und einem anderen Agenten beschreiben. In dem Diagramm müssen alle Aktionen des Agenten aus den Szenarien der „Role Identification Diagrams“, an denen der Agent teilnimmt, betrachtet werden. Ausgehend von den „Role Identification Diagrams“ entstehen so die einzelnen Tasks der Agenten.

In Abbildung 15, Abbildung 16 und Abbildung 17 werden die „Task Specification Diagrams“ für Angebot, Nachfrage und Benutzerschnittstelle vorgestellt.

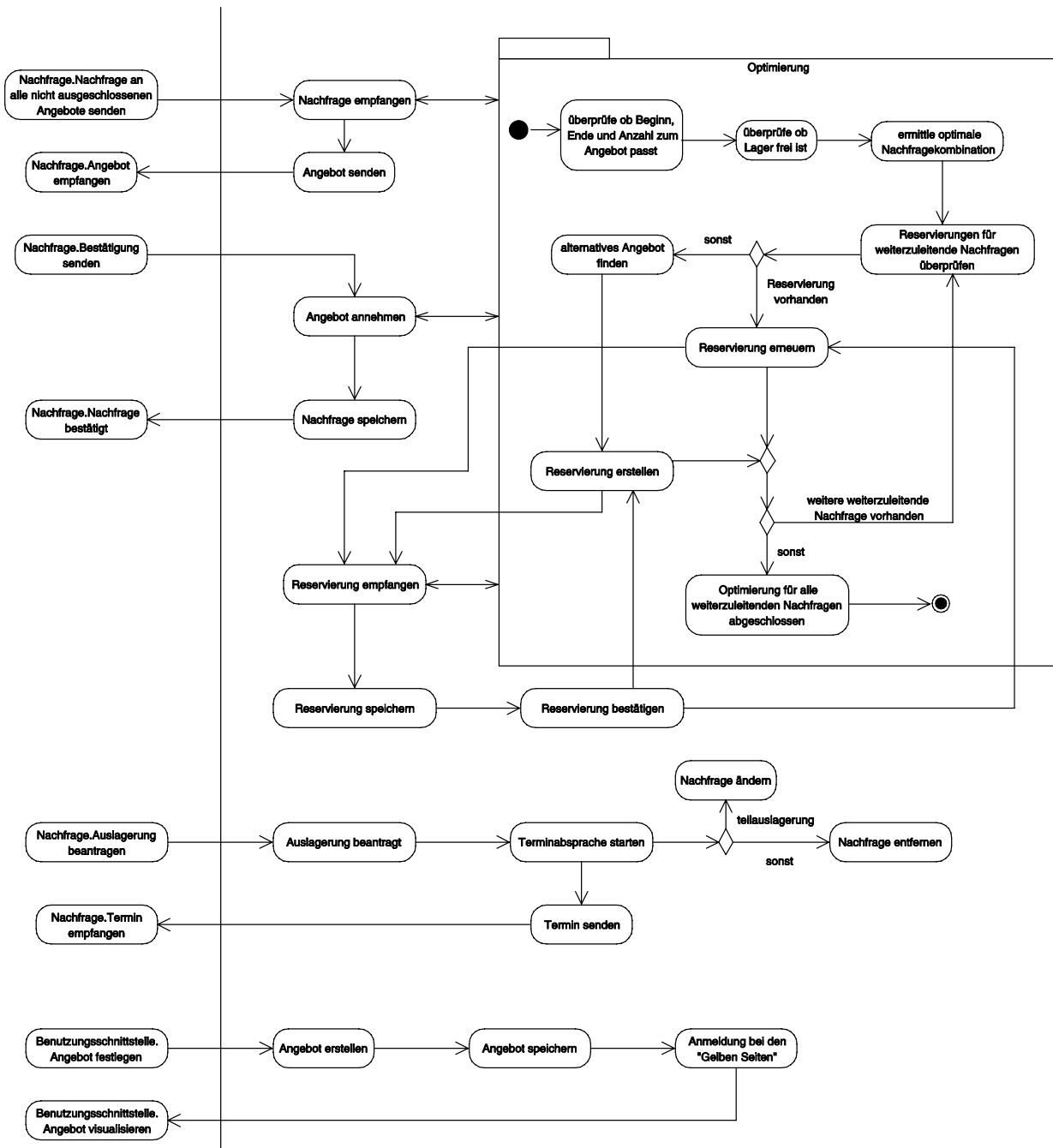


Abbildung 15: Task Specification Diagram - Angebot

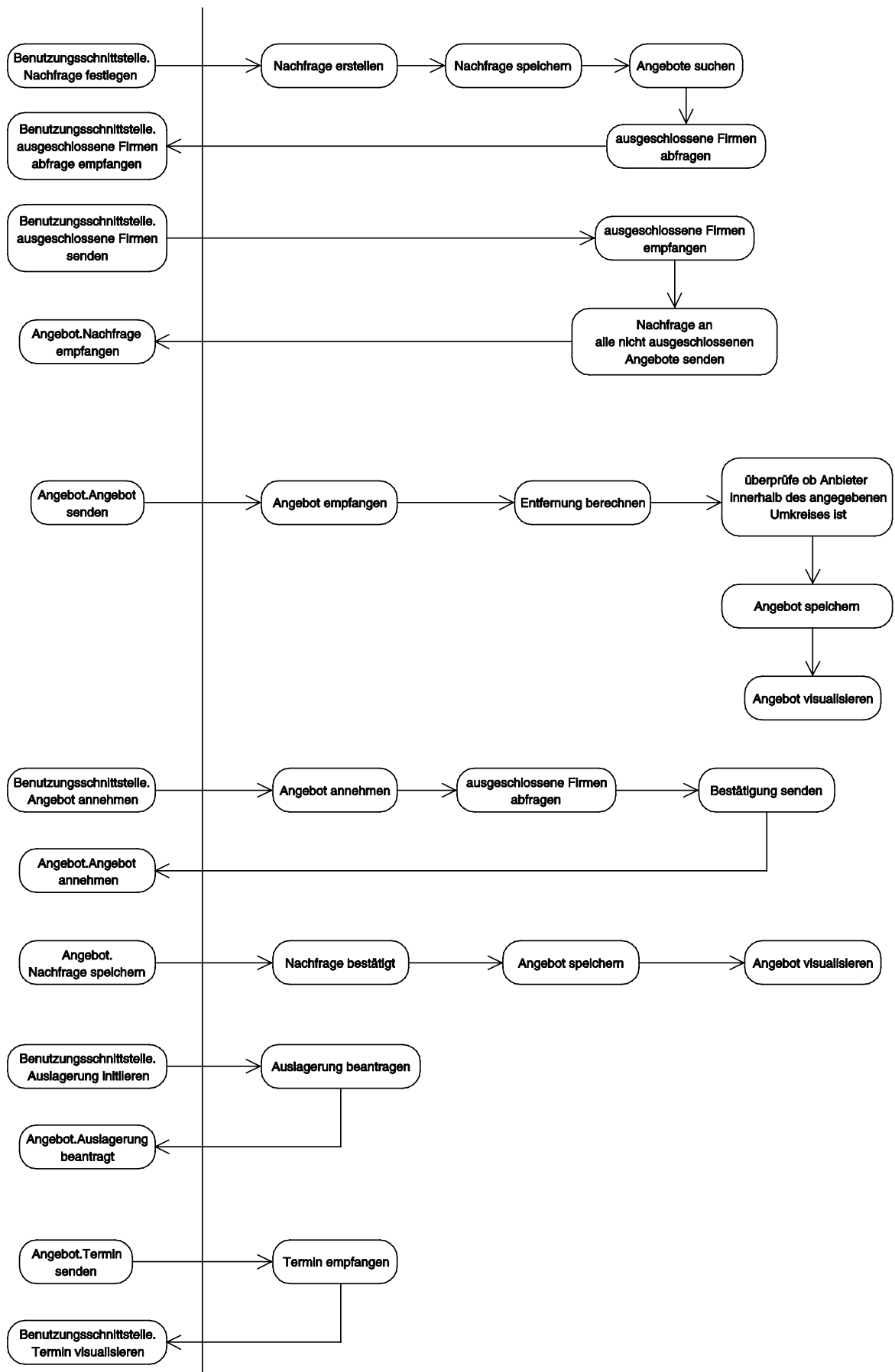


Abbildung 16: Task Specification Diagram – Nachfrage

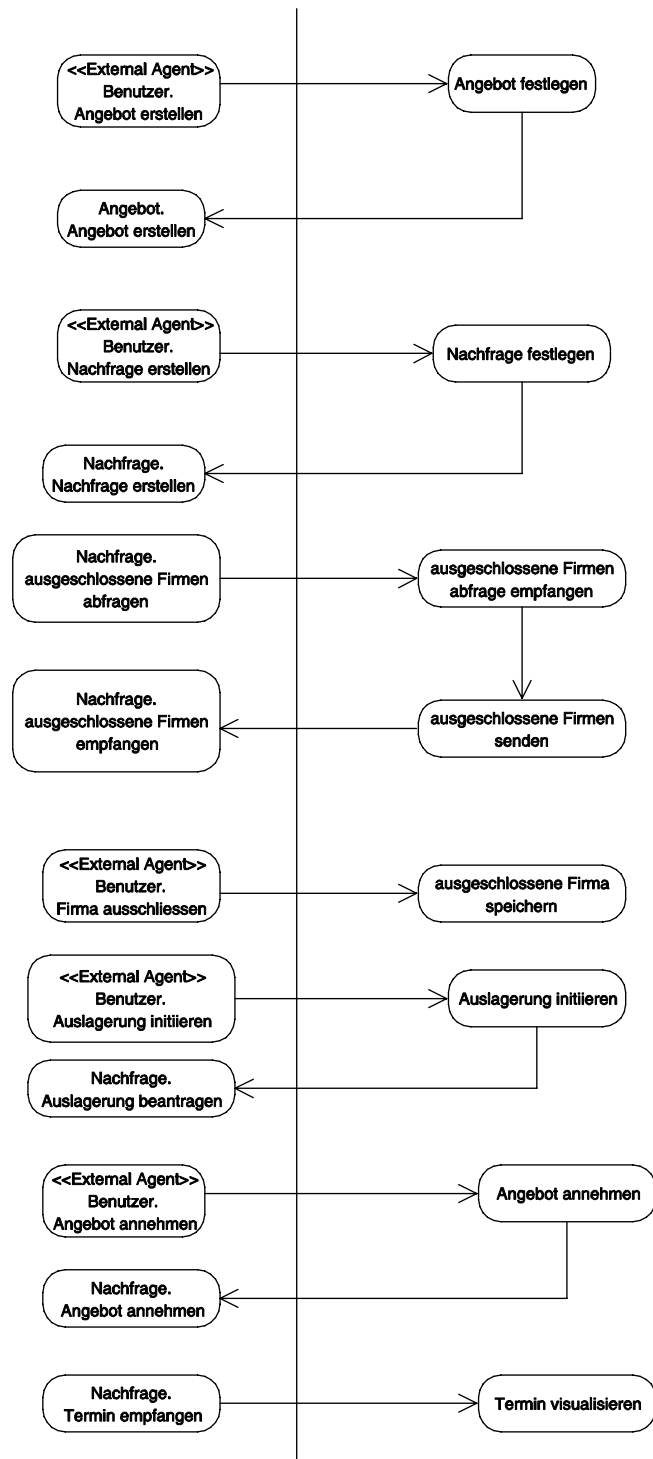


Abbildung 17: Task Spezifikation Diagram – Benutzerschnittstelle

3.5 Ontology Description

Wie in Abbildung 6 gezeigt, wird die Beschreibung der Ontologie dem „Agent Society Model“ zugeordnet. Im „Domain Ontology Description Diagram“ wird das Agenten-System aus ontologischer Sicht durch ein XML-Schema beschrieben (siehe Abbildung 18). Daraus können automatisch XML-Beschreibungen des Inhaltes einer Nachricht generiert werden. Das Diagramm wird als Klassendiagramm dargestellt. Jedes Konzept besteht dabei aus einem oder mehreren anderen Konzepten oder einfachen Datentypen wie Integer, String, etc. Jede Transition beschreibt eine „besteht aus“-Beziehung. Mit Hilfe der Multiplizitäten der Transitionen kann die Länge eines Arrays dargestellt werden.

Im „Communication Ontology Description Diagram“ ist jeder Agent mittels einer Klasse dargestellt (siehe Abbildung 19). Diese beinhaltet sein Wissen (Attribute). Hier steht die Kommunikation zwischen den Agenten im Vordergrund. Jeder Nachrichtentyp wird durch eine Assoziationsklasse eingefügt, welche jeweils ein Attribut für ihre Ontologie, ihre Sprache und ihr Interaktionsprotokoll enthält. In der „Role Identification Phase“ wurde die Kommunikation zwischen den Agenten bereits untersucht (siehe Kapitel 3.3.).

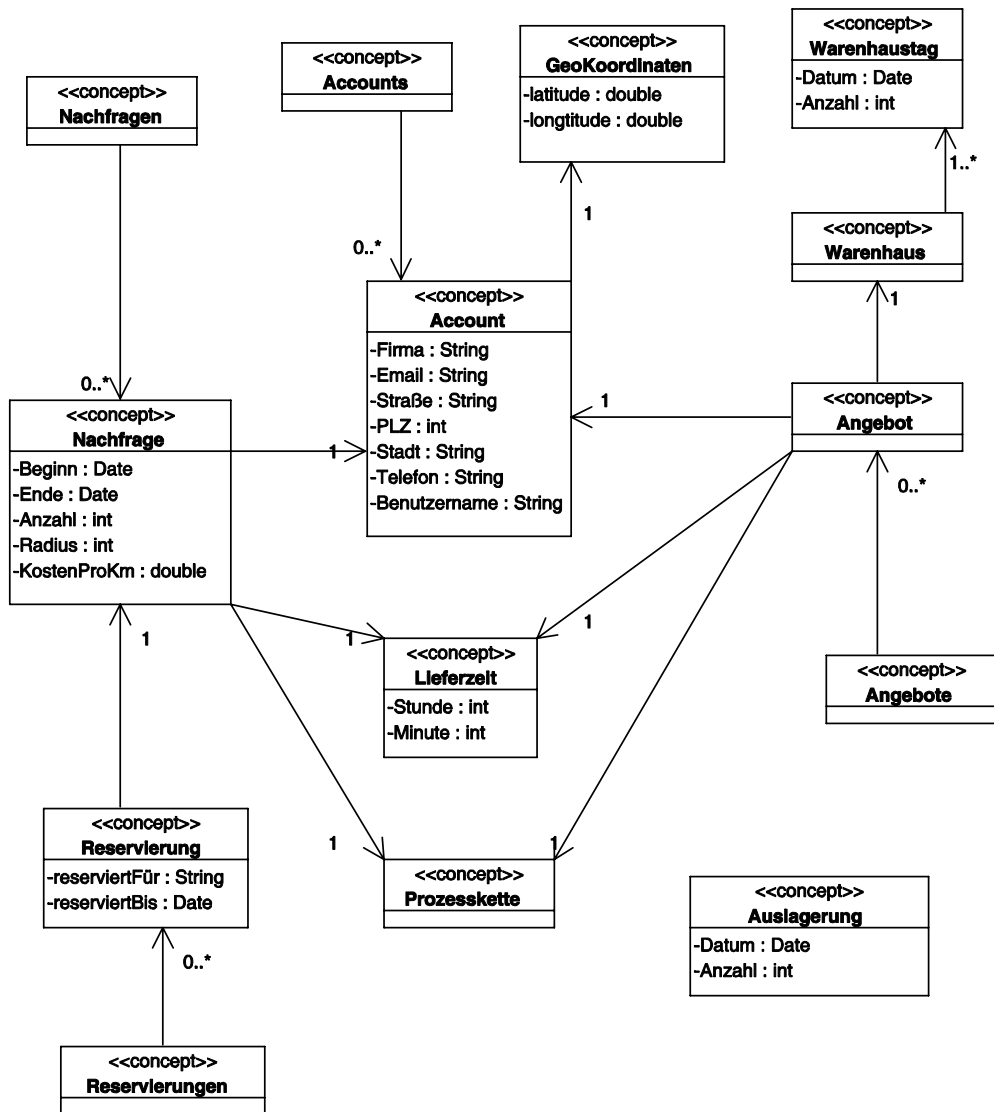


Abbildung 18: Domain Ontology Description Diagram

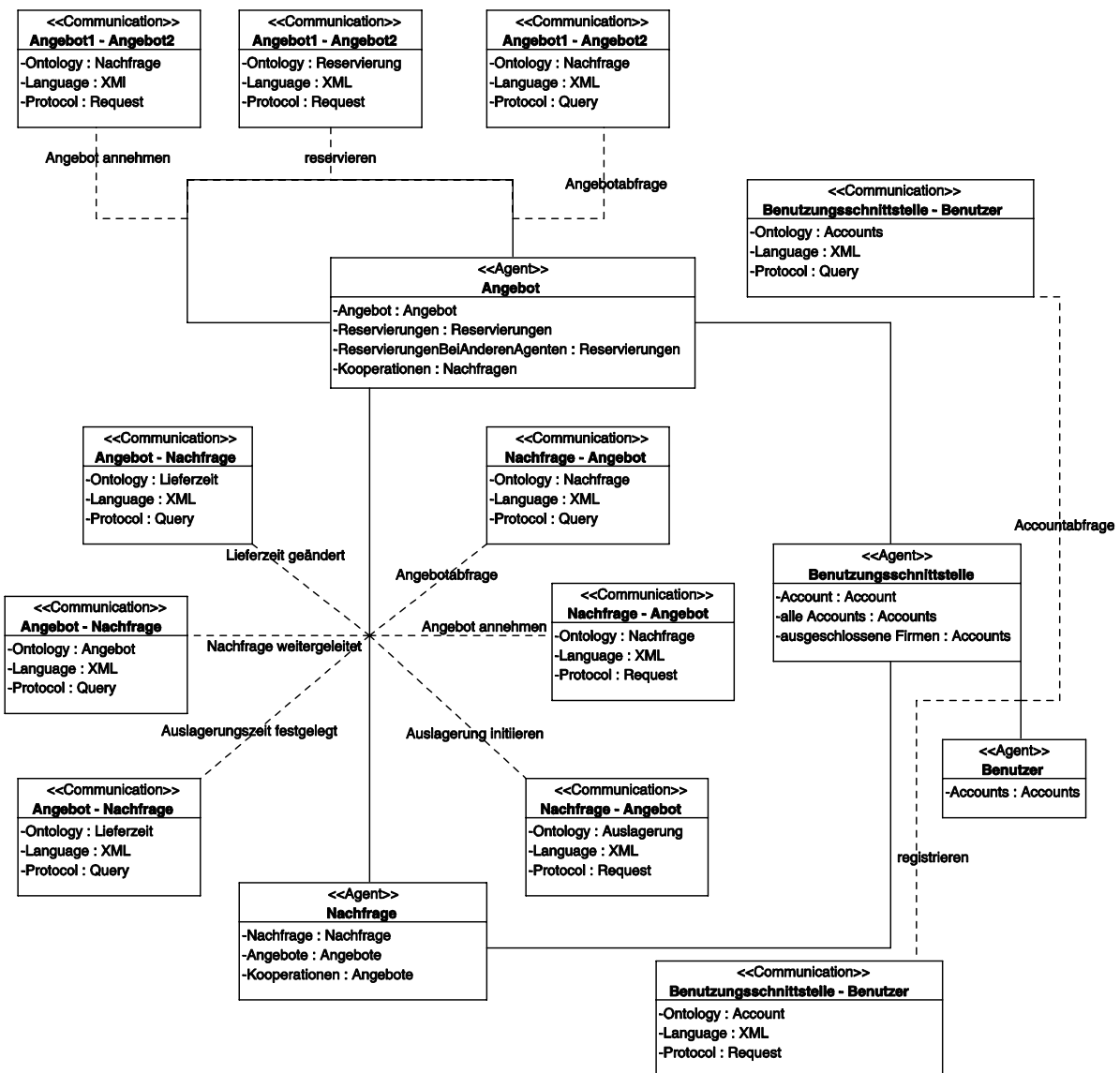


Abbildung 19: Communication Ontology Description Diagram

3.6 Roles Description

In der „Roles Discription Phase“ werden die Aufgaben der Agenten festgelegt. Rollenwechsel, die ein Agent während seines Lebenszyklus durchlebt werden in dem zugehörigen Diagramm (siehe Abbildung 20) dargestellt. Jede Rolle ist in diesem Diagramm eine Klasse, welche in einem Package pro Agenten zusammengefasst werden. Die Tasks sind als Operationen der Agenten eingefügt. Das Diagramm illustriert, welche Fähigkeiten eine Rolle hat. Rollenwechsel werden mit einer gestrichelten Linie mit der Bezeichnung „[ROLE CHANGE]“ dargestellt. Ebenfalls sind die Kommunikationen aus den vorherigen Diagramm eingefügt.

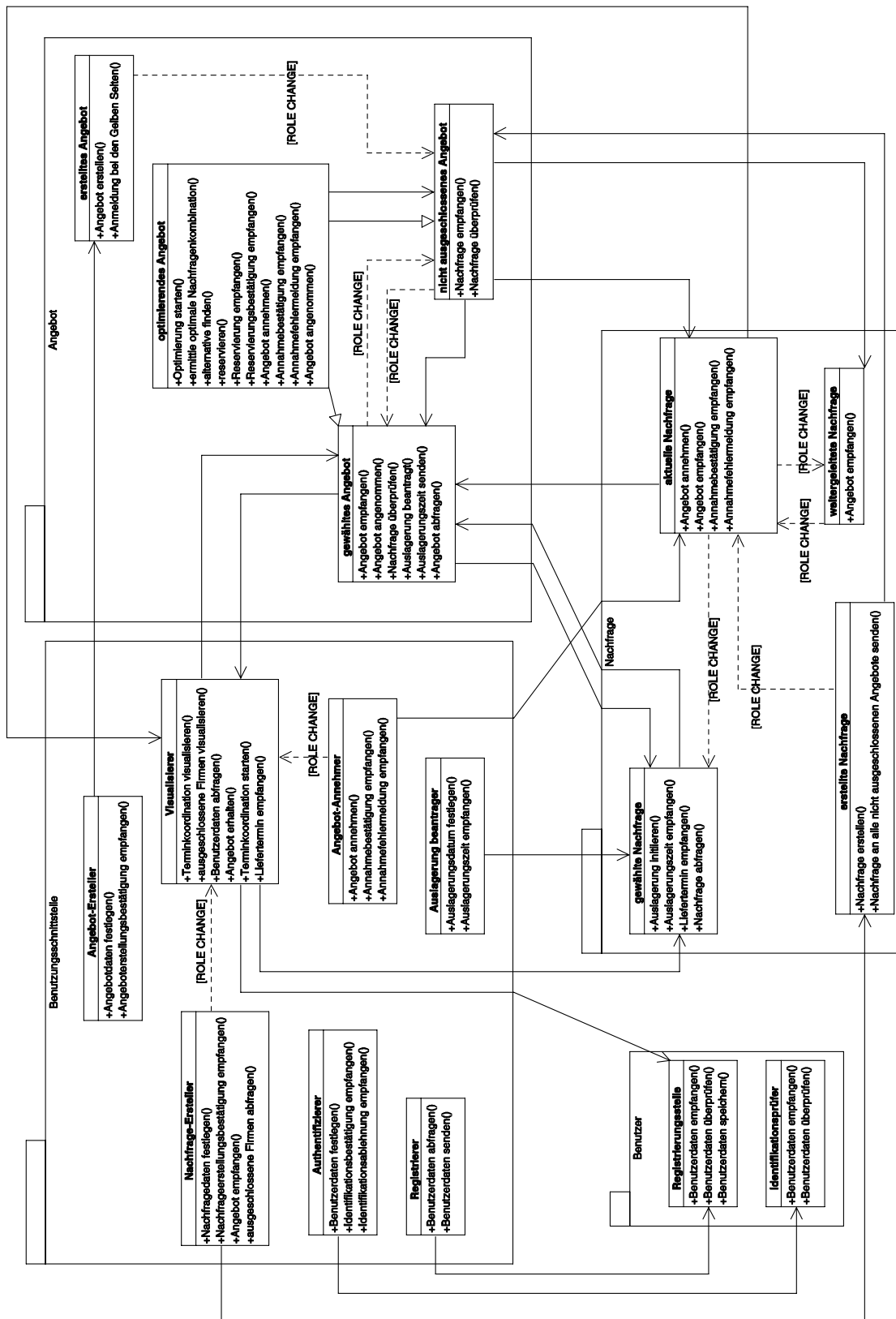


Abbildung 20: Role Description Diagram

3.7 Agent Implementation Model

Das Entwickeln von Multi-Agenten-Systemen kann in zwei logische Ebenen unterteilt werden. In der Multi-Agent Ebene wird jeder Agent als ein Element der „Agentengemeinschaft“ betrachtet und sein Verhalten innerhalb dieser Gemeinschaft beschrieben. In der Single-Agent Ebene wird detailliert die Struktur des einzelnen Agenten definiert. Diese Definition kann in einem späteren Schritt zur automatischen Quell-Code-Erzeugung genutzt werden.

Zur Erstellung des „Agent Implementation Model“ muss eine Iteration auf zwei Ebenen erfolgen. Die äußere Iterationsebene bewegt sich vom „Multi Agent Model“ zum „Single Agent Model“. In der zweiten Ebene wird die Struktur der Agenten vervollständigt („Agent Structure Definition“) und davon ausgehend ihr Verhalten beschrieben („Agent Behaviour Description“). Abbildung 23 bis Abbildung 31 zeigen die verschiedenen Verhaltens-Beschreibungen auf Multi-Agenten Ebene. Abbildung 32 stellt die daraus folgende Definition der Struktur auf Multi-Agenten Ebene dar.

Abbildung 33 bis Abbildung 35 definieren die Struktur auf Single-Agent Ebene. In der „Single-Agent Structure Definition Phase“ wird für jeden Agenten ein Klassendiagramm erstellt, welches eine Klasse für den Agenten selber enthält, sowie eine Klasse für jeden Task. In den Diagrammen wird auf die später verwendete Programmiersprache und das verwendete Framework „JADE“ (s. Kapitel 4) eingegangen. Die Klasse des Agenten erbt somit von der JADE-Klasse „Agent“ und die Tasks von den Unterklassen der JADE-Klasse Behaviour. Die Diagramme können dementsprechend später zur automatischen Quell-Code-Generierung verwendet werden.



Abbildung 21: Multi-Agent Behaviour Description - Angebot.Angebot erstellen

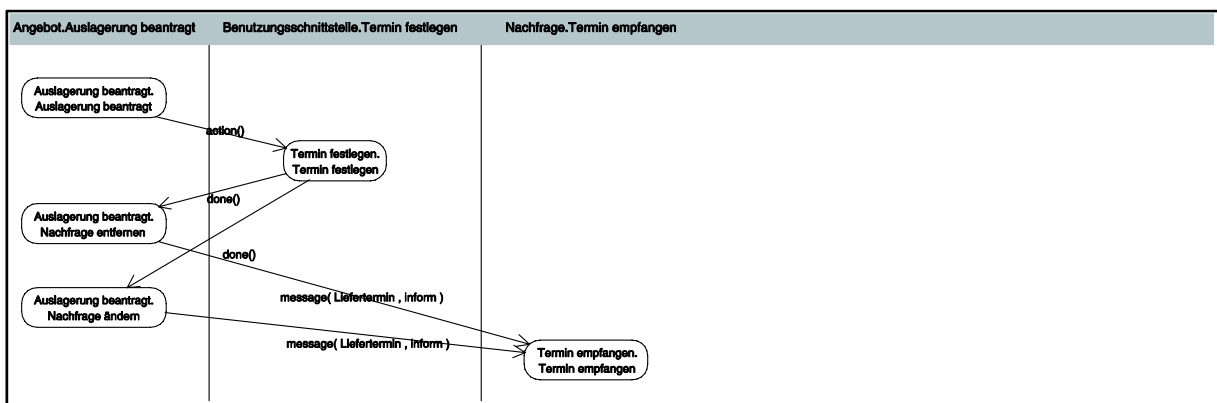


Abbildung 22: Multi-Agent Behaviour Description – Angebot.Auslagerung beantragen

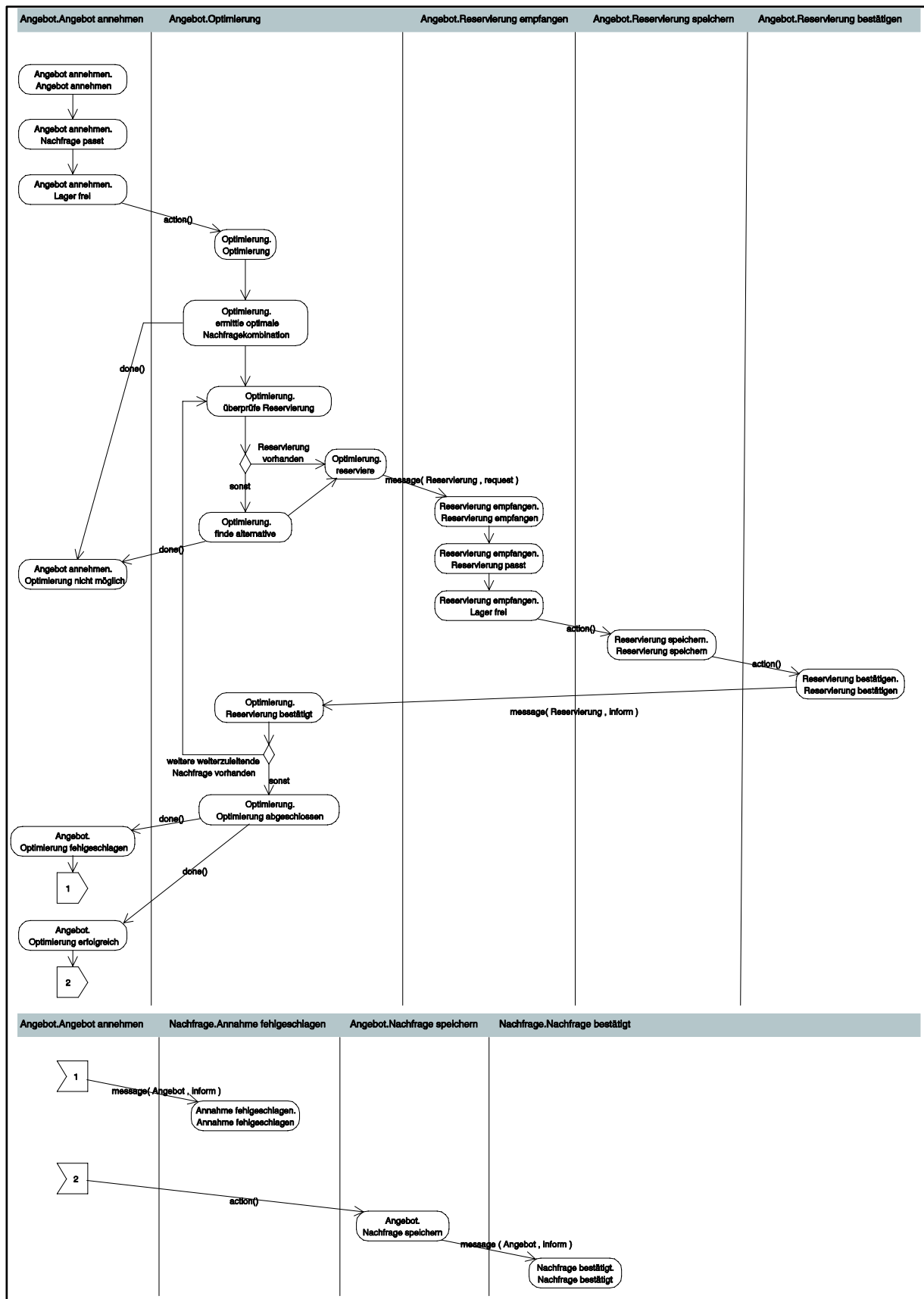


Abbildung 23: Multi-Agent Behaviour Description - Angebot.Angebot annehmen

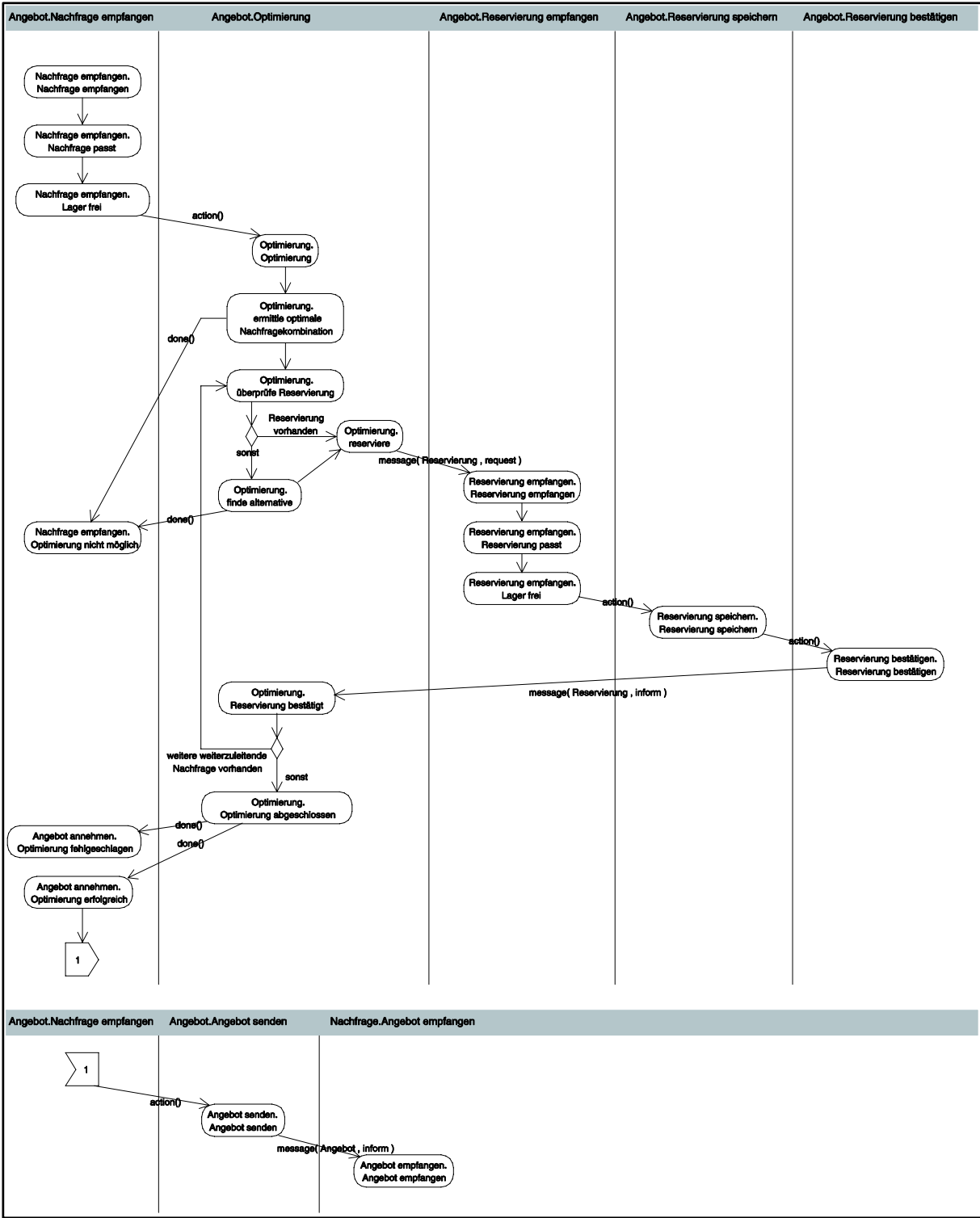


Abbildung 24: Multi-Agent Behaviour Description - Angebot.Nachfrage empfangen

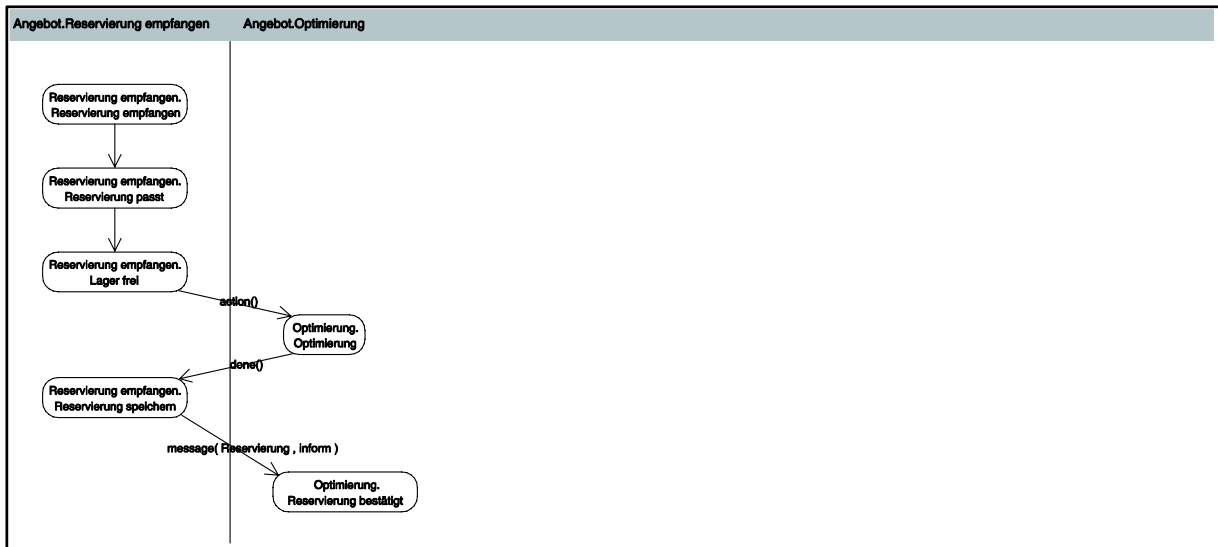


Abbildung 25: Multi-Agent Behaviour Description - Angebot.Reservierung empfangen

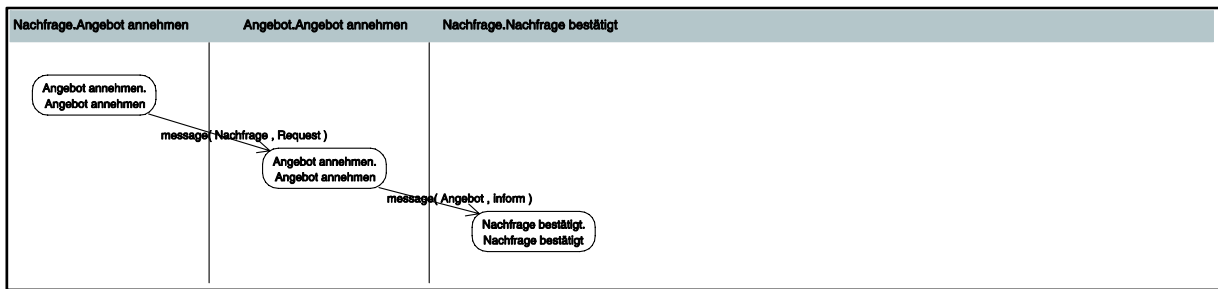


Abbildung 26: Multi-Agent Behaviour Description - Nachfrage.Angebot annehmen



Abbildung 27: Multi-Agent Behaviour Description - Nachfrage.Angebot empfangen

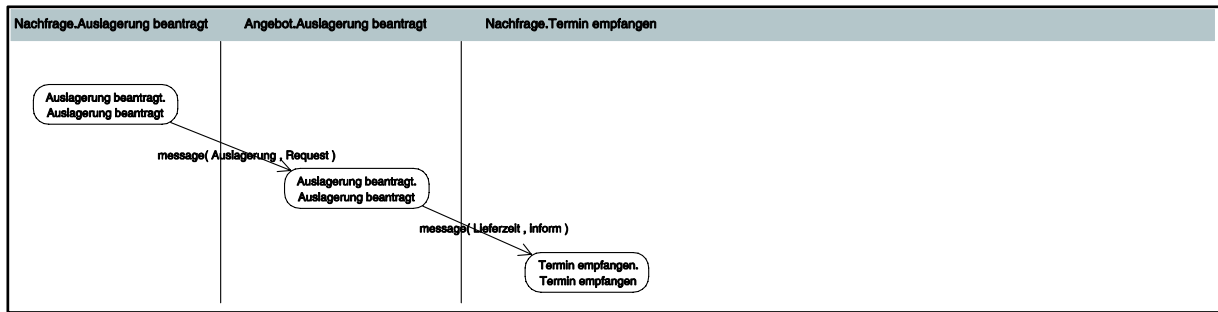


Abbildung 28: Multi-Agent Behaviour Description - Nachfrage.Auslagerung beantragt

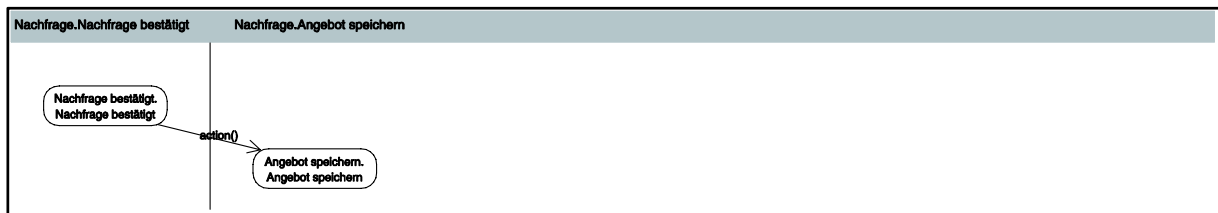


Abbildung 29: Multi-Agent Behaviour Description - Nachfrage.Nachfrage bestätigt

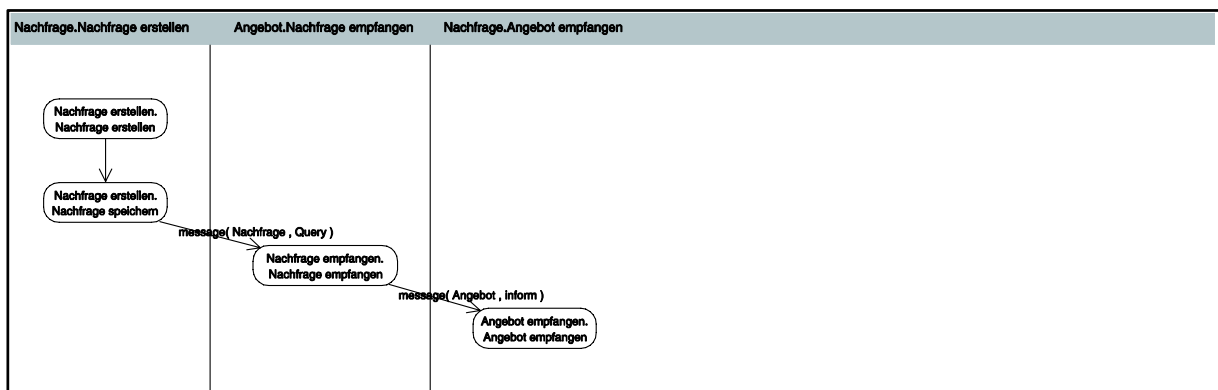


Abbildung 30: Multi-Agent Behaviour Description - Nachfrage.Nachfrage erstellen

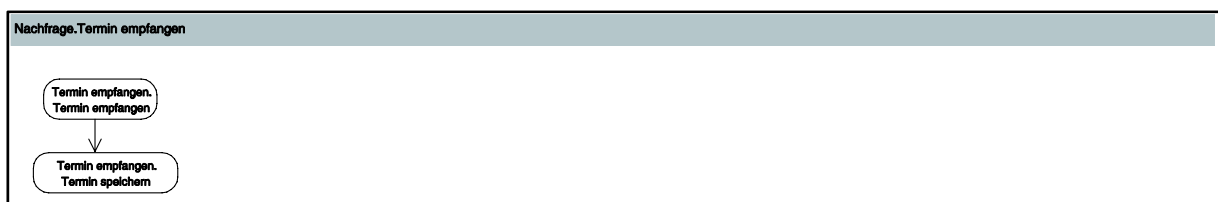


Abbildung 31: Multi-Agent Behaviour Description - Nachfrage.Termin empfangen

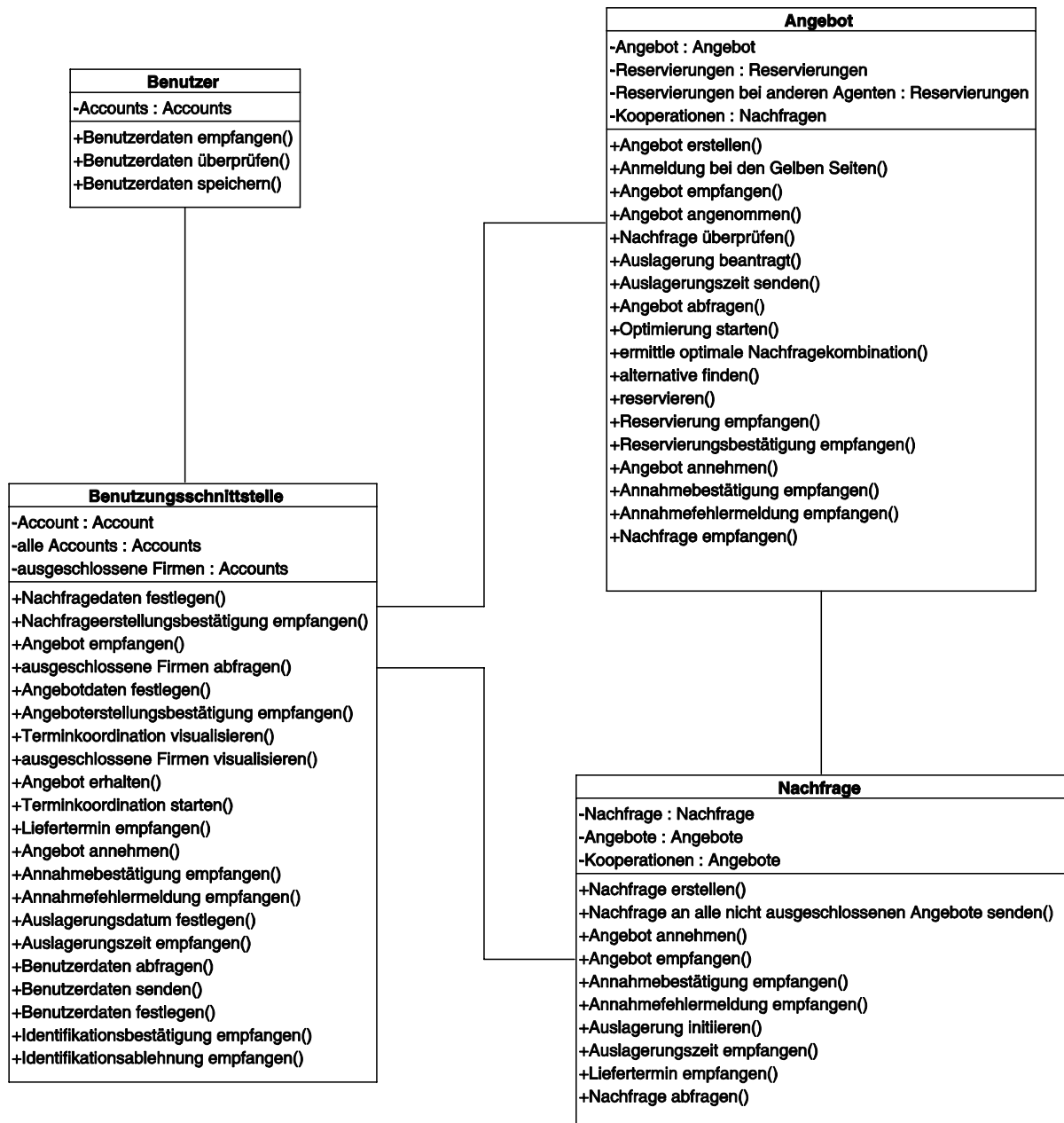


Abbildung 32: Multi-Agent Structure Definition Diagram

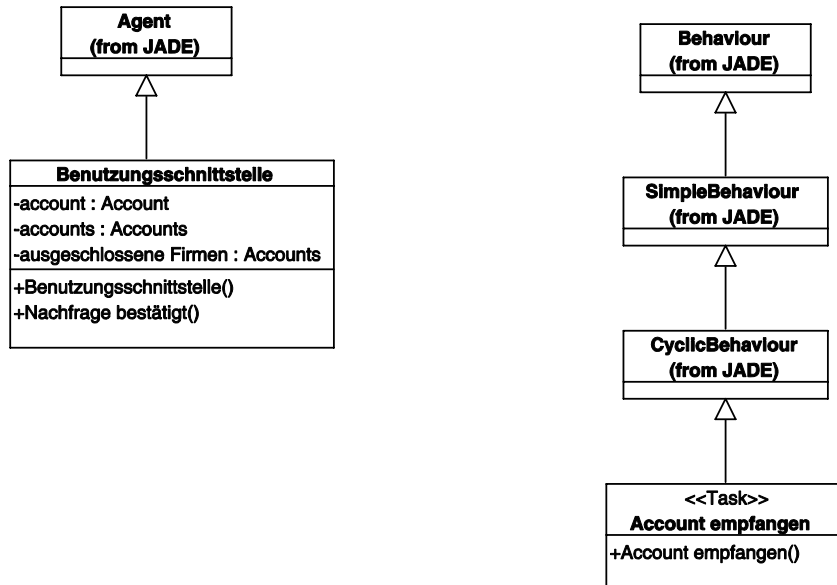


Abbildung 33: Single-Agent Structure Definition – Benutzungsschnittstelle

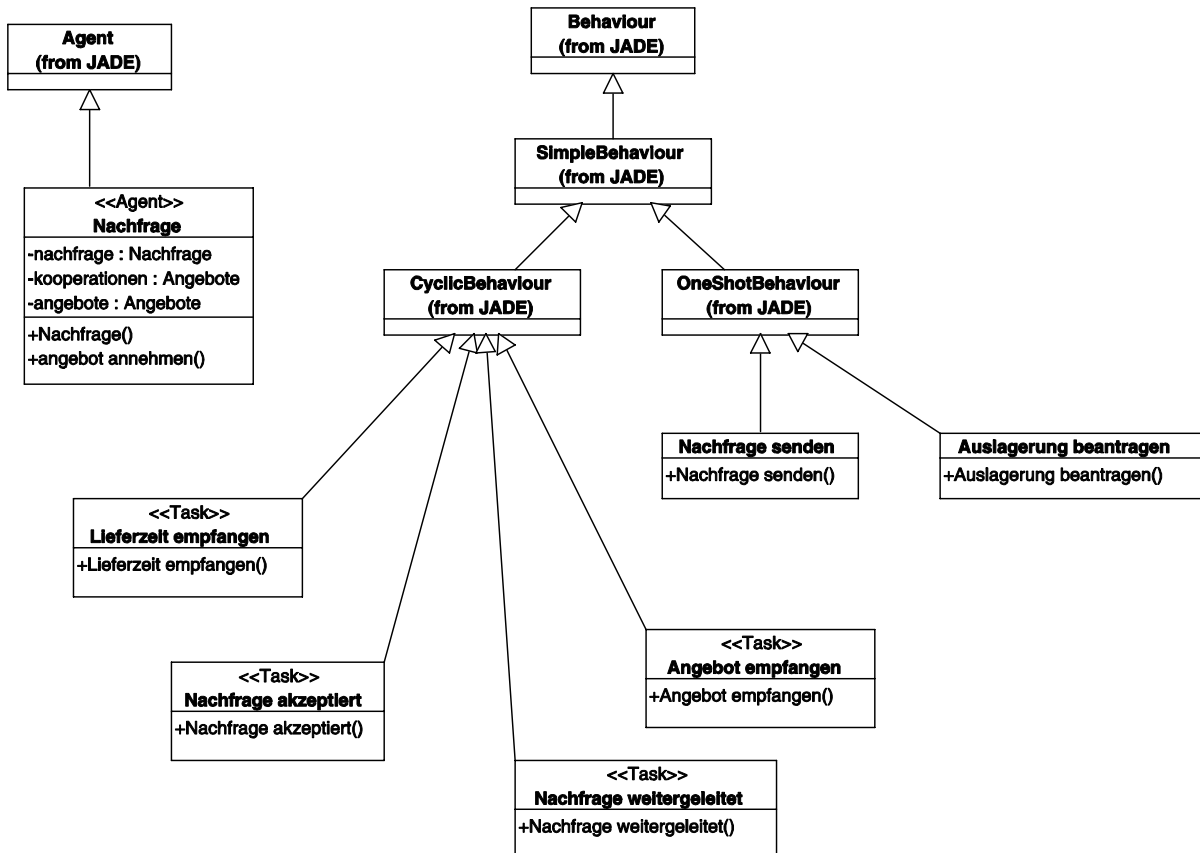


Abbildung 34: Single-Agent Structure Definition – Nachfrage

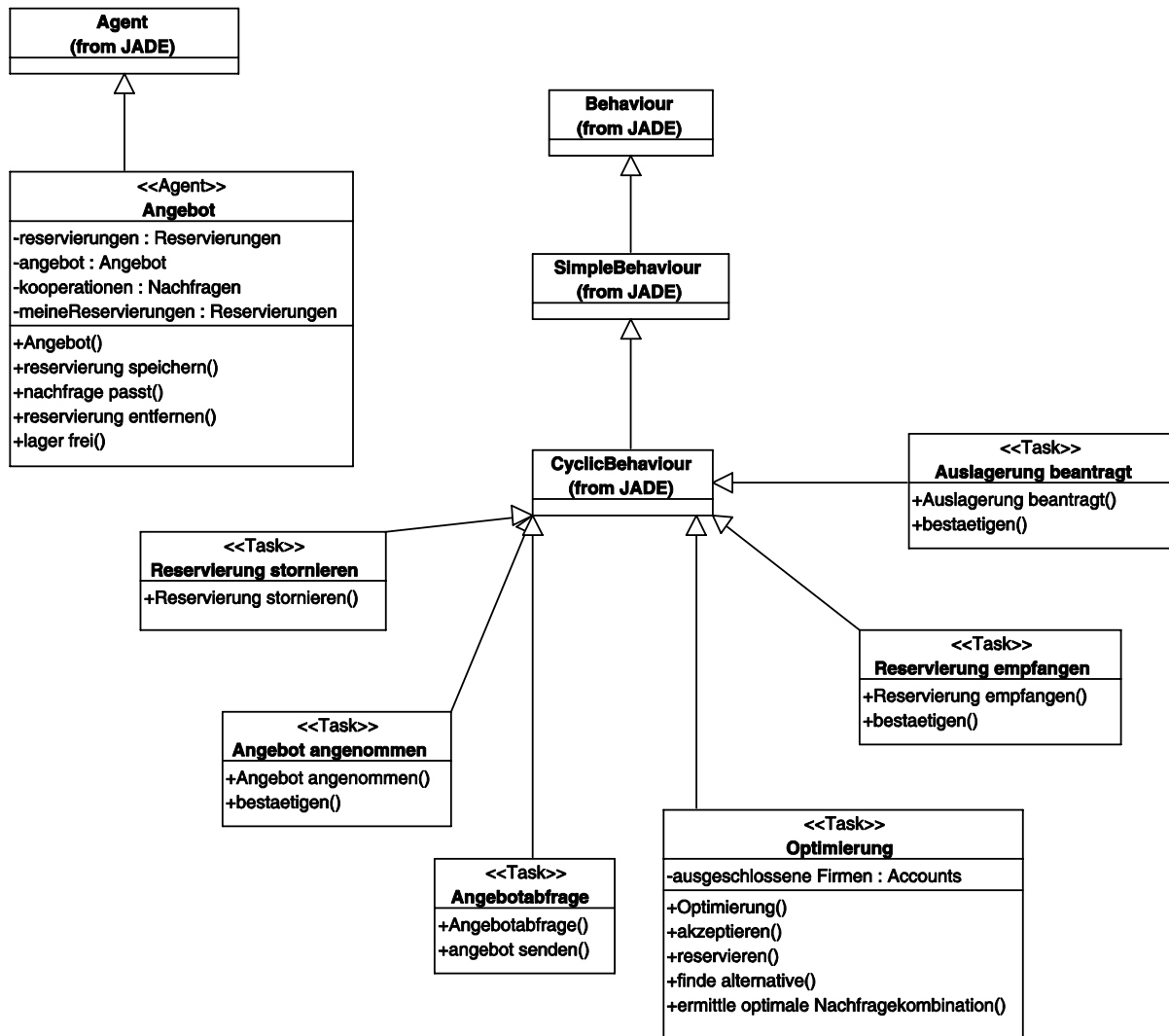


Abbildung 35: Single-Agent Structure Definition - Angebot

4 Ausblick auf die prototypische Umsetzung des Logistik-Informationssystems

In diesem Technical Report wurde die Entwicklung des Logistik-Informationssystems nach der PASSI-Methode beschrieben. Zur Implementierung des Logistik-Informationssystems als Multi-Agenten-System wird in einem weiteren Schritt das frei verfügbare Development-Framework JADE verwendet [JAD08]. Diese Laufzeitumgebung ist in der Programmiersprache Java geschrieben und stellt dem Entwickler Tools zum Programmieren und Verwenden von Agenten zur Verfügung.

Die von JADE zur Verfügung gestellte Basisklasse „Agent“ bietet mit der Agentenplattform grundlegende Basisinteraktionen, sowie Funktionen zum Versenden und Empfangen von Nachrichten. Des Weiteren unterstützt JADE die PASSI-Entwicklungsmethode für Multi-Agenten-Systeme und die Ergebnisse der PASSI-Diagramme können direkt für die Codegenerierung übernommen werden. Jeder vom Agent angebotene Service wird in einem „Behaviour“ implementiert. Die Behaviour werden von einem im Agenten vorhandenen Scheduler quasi parallel mit einem nicht-preemptiven Round-Robin-Verfahren verarbeitet. Dazu gibt es verschiedene vorgefertigte „Behaviour“ die z.B. nur einmal ausgeführt werden („OneShotBehaviour“) oder bis zum Entfernen immer wieder vom Scheduler aufgerufen werden („CyclicBehaviour“). Die quasi parallele Verarbeitung verschiedener Verhalten ist notwendig, um auf unterschiedliche äußere Bedingungen reagieren zu können. Damit der Rechner nicht durchgehend ausgelastet ist und auch andere „Behaviour“ arbeiten können, hat jede Behaviour die Möglichkeit mittels der „block()“-Funktion anstelle von einer aktiven Endlosschleife, auf eine ankommende Nachricht zu warten.

Die prototypische Umsetzung des Logistik-Informationssystems mit JADE stellt den nächsten Meilenstein bei der Modellierung eines ‚Virtuellen Lagers‘ da.

5 Literatur

- [BCo02] Burrafato, P.; Cossentino, M.: Designing a multi-agent solution for a bookstore with the PASSI methodology. Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002). 27-28 Mai 2002, Toronto (Ontario, Canada).
- [CPo02] Cossentino M.; Potts, C. - "A CASE tool supported methodology for the design of multi-agent systems" - The 2002 International Conference on Software Engineering Research and Practice (SERP'02) - June 24 - 27, 2002 - Las Vegas (NV), USA
- [JAD08] TILab: Java Agent Development Framework (JADE) <http://jade.tilab.com> online am 15.02. 2008.
- [PAS08] Cossentino, M., et. al.: PASSI Homepage and PASSI ToolKit. <http://mozart.csai.unipa.it/passi> online am 15.02.2008.
- [ZEb06] Zellerhoff, J.; Ebel, D.: Transportvergabe online. In: IT-Mittelstand (2006) 12, S. 26-28.
- [ZEb07] Zellerhoff, J.; Ebel, D.: Geteiltes Lager, voller Erfolg. In: Logistik heute (2007) 1-2, S. 46-47.
- [ZJo08] Zellerhoff, J.; Jodin, D.: Im virtuellen Lager erhöhen autonome Agenten den Service. In: Wolf-Kluthausen, H. (Hrsg.): Jahrbuch der Logistik 2008. free beratung GmbH, Korschbroich, 2008, S. 219-223.

Sonderforschungsbereich 559

Bisher erschienene Technical Reports

- 06007 Peter Kemper, Carsten Tepper: Trace Analysis – Gain Insight through Modelchecking and Cycle Reduction
- 06008 Jochen Bernhard, Dirk Jodin, Kay Hömberg, Sonja Kuhnt, Christoph Schürmann, Sigrid Wenzel: Vorgehensmodell zur Informationsgewinnung – Prozessschritte und Methodennutzung
- 06009 Doris Blutner, Stephan Cramer, Sven Krause, Tycho Mönks, Lars Nagel, Andreas Reinholz, Markus Witthaut: Ergebnisbericht der Arbeitsgruppe 5 „Assistenzsysteme für die Entscheidungsunterstützung“
- 07001 Falko Bause, Tobias Hegmanns, Stefan Pietzarka, Veye Tatah, Markus Witthaut: Ergebnisbericht der Arbeitsgruppe – Neues Problemverständnis: Ergänzung des Modellierungsparadigmas
- 07002 Arnd Bernsmann, Peter Buchholz, Stephan Kessler, Andreas Reinholz, Britta von Haaren, Markus Witthaut: Bewertungs- und Dimensionierungsmethoden im Sonderforschungsbereich 559
- 07003 Jochen Bernhard, Kay Hömberg, Lars Nagel, Iwo Riha, Christoph Schürmann, Harald Sieke, Marcus Völker: Standardisierte Modelle zur Systemlastbeschreibung
- 07004 Kay Hömberg, Jan Hustadt, Dirk Jodin, Joachim Kochsiek, Lars Nagel, Iwo Riha: Basisprozesse für die Modellierung in großen Netzen der Logistik
- 07005 Kay Hömberg, Dirk Jodin, Reineke: Bewertung und Kategorisierung der Methoden zur Datenerhebung
- 07006 Jochen Bernhard, Miroslaw Dragan: Bewertung der Informationsgüte in der Informationsgewinnung für die modellgestützte Analyse großer Netze der Logistik
- 07007 Britta von Haaren, Tatjana Malyshko: Integration of Velos-Simulation-Results into the Supply Chain Balanced Scorecard
- 07008 Britta von Haaren, Ivana Humpolcová: Ansätze zur Systematisierung des Instrumentariums zum Supply-Chain-Risikomanagement
- 08001 Jan Hombergs, Iwo Riha: Softwareauswahl für den Einsatz von Cost Benefit Sharing in Logistiknetzwerken
- 08002 Jörg Zellerhoff: Modellierung eines Informationssystems für ein ‚Virtuelles Lager‘

Alle Technical Reports können im Internet unter
<http://www.sfb559.uni-dortmund.de/>
abgerufen werden. Für eine Druckversion wenden Sie
sich bitte an die SFB-Geschäftsstelle
E-Mail: andrea.zoeller@iml.fraunhofer.de