

Bericht

Agentenbasiertes Kommunikationskonzept zur Realisierung von E-Commerce-Diensten in zellularen Mobilfunknetzen - Entwurf, Implementierung und Bewertung

von

Jens Hartmann

Matr.-Nr. 184 182

Aachen, den 29. November 2001

Betreuung durch:
o. Prof. Dr.-Ing. B. Walke

Die Arbeit ist nur zum internen Gebrauch bestimmt. Alle Urheberrechte liegen beim betreuenden Lehrstuhl. Für den Inhalt wird keine Gewähr übernommen. Vervielfältigungen und Veröffentlichungen jeglicher Art sind nur mit Genehmigung des Lehrstuhls gestattet.

Ich versichere, daß die vorliegende Arbeit - bis auf die offizielle Betreuung durch den Lehrstuhl - ohne fremde Hilfe von mir durchgeführt wurde. Die verwendete Literatur ist im Literaturverzeichnis vollständig angegeben.

Aachen, den 29. November 2001

(Jens Hartmann)

KURZFASSUNG

Die Nutzung von Internet- und E-Commerce-Diensten von Mobiltelefonen aus wird durch die einengenden Rahmenbedingungen von Mobilfunknetzen und -telefonen, wie schmalbandige Kanäle, hohe Laufzeitverzögerungen und eingeschränkte Benutzeroberfläche, erheblich erschwert.

In der vorliegenden Arbeit wird ein agentenbasiertes Kommunikationskonzept für Mobilfunknetze entworfen und bewertet, welches eine effiziente Nutzung des Internets auch von Mobiltelefonen aus ermöglicht. Dafür wird ein Dienstknoten konzipiert und implementiert, der auf einem Agentensystem basiert und verschiedene moderne Kommunikationsprotokolle unabhängig vom eingesetzten Trägerdienst unterstützt. Die Leistungsfähigkeit des neu entwickelten agentenbasierten Kommunikationskonzeptes für Mobilfunknetze wird anhand von zwei E-Commerce-Diensten analysiert, modelliert und bewertet. Während der erste Dienst mobile Agenten für die effiziente Übermittlung von HBCI-basierten Bankaufträgen vom Mobiltelefon aus einsetzt, nutzt der zweite Dienst mobile Agenten für eine schnelle und komfortable Suche nach Konsumgütern und Dienstleistungen via WAP im Internet. Die Untersuchungen zeigen, dass das Agentenparadigma eine dynamische, personalisierte und kontextabhängige Dienstangebotung in Mobilfunknetzen ermöglicht. Weiterhin wird gezeigt, dass mittels des agentenbasierten Kommunikationskonzeptes bei der Datenübertragung in GSM- und GPRS-Mobilfunknetzen eine signifikante Reduktion der Dienstausführungszeiten und Kommunikationskosten erzielt werden kann. Dies wird durch den Vergleich der agentenbasierten Dienste mit klassischen Lösungen nachgewiesen.

ABSTRACT

The use of Internet and E-Commerce services from mobile phones is limited due to the restrictions of narrow channels, high latencies and uncomfortable user interfaces within mobile radio networks and phones.

In this thesis an agent-based communication concept for mobile radio networks is developed and evaluated. For this purpose, a new service node is introduced and implemented, which is based on an agent system and supports several modern communication protocols independent of the underlying bearer services. The performance of the agent-based communication concept is evaluated based on two e-commerce services. The first service uses mobile agents for the efficient submission of banking orders via mobile phones, while the second service is a sophisticated and comfortable agent-based product search for WAP-enabled phones. The investigations show that the agent paradigm allows a dynamic, personalized and context-driven service provisioning in mobile radio networks. Moreover, the agent-based communication concept allows to significantly decrease service execution times, and communication costs for mobile data transmission over GSM and GPRS bearers. This conclusion is drawn from the comparison of the agent-based services with traditional service solutions.

INHALTSVERZEICHNIS

Kurzfassung	1
Abstract	2
1 Einleitung und Problemstellung	7
1.1 Motivation und Problemstellung	7
1.2 Vorgehensweise	8
2 Dienstanbieter in Telekommunikationsnetzen	11
2.1 Der Dienstbegriff	11
2.2 Moderne Kommunikationskonzepte	13
2.2.1 Intelligente Netze	14
2.2.2 GSM-Mobilfunknetze	15
2.2.3 CAMEL-Standard	18
2.2.4 Virtual Home Environment	19
2.2.5 WAP-Standard	21
2.3 Paketerorientierte GPRS-Kommunikation	24
2.3.1 GPRS-Mobilstation	25
2.3.2 Logische GPRS-Architektur	26
2.3.3 Protokollstapel	27
2.3.4 Datenraten	28
2.3.5 Dienstgüteparameter	29
2.4 Zusammenfassung	32
3 Die Agententechnologie	33
3.1 Historie	33
3.2 Was ist ein Agent? - Begriffsdefinitionen	34
3.2.1 Autonomie	35
3.2.2 Mobilität	35
3.2.3 Systemterminologie	37
3.2.4 Agententypen	39
3.2.5 Lebenszyklus	40
3.2.6 Sicherheitsaspekte	41
3.3 Mechanismen für die agentenbasierte Kommunikation	43
3.3.1 Nachrichtenaustausch	45
3.3.2 Eventmechanismus	46
3.3.3 Blackboard	47
3.3.4 Agentensprachen	47
3.3.5 Einsatz von XML, PHP und SQL	48
3.4 Agentensysteme	51
3.4.1 Voyager	52
3.4.2 Grasshopper	55
3.5 Standardisierung	56
3.5.1 FIPA - Foundation for Intelligent Physical Agents	56

3.5.2	OMG - Object Management Group	57
3.6	Zusammenfassung	57
4	CORBA-Unterstützung für agentenbasierte Kommunikationskonzepte	59
4.1	Grenzen von Agentensystemen	59
4.2	CORBA-Standard	60
4.3	MASIF-Standard	63
4.4	Einsatz von CORBA im Agentenkontext	65
4.4.1	Benutzerschnittstelle	66
4.4.2	IIOP für die Agentenkommunikation	67
4.4.3	Kooperation von Agentensystemen verschiedenen Typs	69
4.5	Realisierung einer Systemkommunikation nach MASIF	70
4.5.1	Die Grasshopper-Region	70
4.5.2	Die Voyager-Region	73
4.6	Zusammenfassung	74
5	Agentenbasiertes Kommunikationskonzept für M-Commerce-Dienste	77
5.1	Anforderungen an eine intelligente Agentenmigration	77
5.1.1	Zugangsnetz	78
5.1.2	Benutzermobilität	79
5.1.3	Endgerätemobilität	80
5.1.4	Netzzustandsabhängige Agentenmigration	80
5.1.5	Paketorientierte Trägerdienste	82
5.2	Agentenplattform für M-Commerce-Dienste	83
5.2.1	Konzept	83
5.2.2	Sicherheit	83
5.2.3	Architektur	84
5.2.4	Kommunikationsfluss	86
5.3	Bankgeschäfte mittels mobiler Endgeräte	87
5.3.1	Anforderungen an eine mobile HBCI-Lösung	88
5.3.2	Mobiles, agentenbasiertes HBCI-Szenario	89
5.3.3	Konzeption und Ablauf des AHB-Dienstes	91
5.4	Such- und Bestelldienste	95
5.4.1	Agentenbasierte Suche	96
5.4.2	Suchmethoden	98
5.4.3	Konzeption mobiler, agentenbasierter Such- und Bestellszenarien	99
5.4.4	Bestellung und Bezahlung	103
5.5	Zusammenfassung	103
6	Modellierung und Analyse des agentenbasierten Kommunikationskonzeptes	105
6.1	Ziele der Analyse	105
6.1.1	Datenvolumen	105
6.1.2	Kommunikationszeit	106
6.1.3	Kommunikationskosten	107
6.2	Prinzipien der verkehrstheoretischen Modellierung	107
6.2.1	Warteschlangenmodelle	107
6.2.2	Dekompositionsverfahren	110
6.2.3	Werkzeuge für die Analyse	111
6.3	Verkehrsmodelle der Netzknoten	112
6.3.1	Mobilfunknetz	113

6.3.2	WAP-Gateway	117
6.3.3	Agentenplattform für M-Commerce-Dienste	119
6.3.4	Internet	121
6.4	Mobile Agenten	122
6.4.1	Modell der Mobilität	123
6.4.2	Modell der Interaktivität	124
6.4.3	Modell der Selektivität	126
6.5	Analyse des Homebankingdienstes	128
6.5.1	Datenvolumen	129
6.5.2	Antwortzeiten	131
6.5.3	Dienstausführungszeit	132
6.5.4	Kommunikationskosten	133
6.5.5	Dienstausführungszeit bei einer GPRS-Verbindung	134
6.6	Leistungsbewertung des Such- und Bestelldienstes	135
6.6.1	Verkehrsmodell für den ASB-Dienst	136
6.6.2	WAP-Kommunikation über CSD	137
6.6.3	Datenvolumen	139
6.6.4	Selektivität	142
6.6.5	Antwortzeiten	143
6.6.6	Dienstausführungszeit	146
6.6.7	WAP-Kommunikation über GPRS	147
7	Zusammenfassung und Ausblick	151
A	Agentensystem IBM Aglets Workbench	155
B	Agentensystem Mitsubishi Concordia	157
C	HBCI-Dialog	159
D	Sicherheit von HBCI	161
	Abbildungsverzeichnis	165
	Tabellenverzeichnis	167
	Abkürzungsverzeichnis	173
	Literaturverzeichnis	181

Einleitung und Problemstellung

In nur wenigen Jahren hat sich das Kommunikationsverhalten erheblicher Teile der Weltbevölkerung spürbar verändert. Das **Internet** und die **Mobilkommunikation** haben nicht nur wesentlichen Einfluss auf die Geschäftswelt, auch der private Bereich erfährt nachhaltige Veränderungen.

Aufgrund des großen Informationsangebotes und des damit verbundenen Zeitaufwandes bei der Informationssuche, ist ein Bedarf an effizienter Software entstanden, die heterogene Kommunikationsnetze nach Informationen durchsucht, diese filtert und verwaltet. Dabei sind die Aufgaben eines modernen **Informationsmanagements** immer komplexer und unüberschaubarer geworden sind. Darüber hinaus werden skalierbare und personalisierte Lösungen für den vielversprechenden Bereich des **elektronischen Handels** (E-Commerce) gesucht. Zudem benötigt man insbesondere im Hinblick auf die kontinuierlich steigende Anzahl von mobilen Endgeräten mit Funkschnittstellen **effiziente Zugriffsmethoden** auf digitale Daten. Auch wenn moderne Funksysteme immer höhere Übertragungsraten bieten, so leidet die Funkübertragung im Vergleich zum Festnetz unter einer höheren Fehleranfälligkeit und begrenzten Funkressourcen. Weiterhin muss berücksichtigt werden, dass der Zugriff auf verschiedene Informationsquellen durch unterschiedliche Kommunikationsmechanismen erschwert wird. Ferner besteht der Wunsch nach leistungsstarken **System- und Netzmanagementfunktionen**. Neben diesen informations- und softwaretechnischen Aspekten gewinnen aus wissenschaftlicher Sicht Methoden zur Abstraktion, Modellierung und Bewertung von Kommunikationsdiensten an Bedeutung.

1.1 Motivation und Problemstellung

Softwareagenten besitzen eine Reihe von Eigenschaften (Autonomie, Intelligenz, Mobilität), die die Bereitstellung und Nutzung von innovativen, verteilten Diensten erleichtert und stellen daher für die geschilderten Probleme eine interessante Lösung dar. Daher sind Ergebnisse aus wissenschaftlichen Untersuchungen der Agententechnologie für Netzbetreiber, Dienstanbieter, Medienindustrie und Hersteller von Telekommunikationsinfrastruktur gleichermaßen von Bedeutung. Während die Benutzer von Telekommunikationsdiensten auf neue Anwendungen mit persönlicher Note warten (Personalisierung), um ihre Einkäufe, Bankkonten, privaten und beruflichen Adressbücher, Multimedia-Briefkästen und Informationsdienste zeit- und kostensparend führen und tätigen zu können, bereitet sich die Industrie auf die nächste Mobilfunkgeneration mit Multimediafähigkeiten vor. Mit steigender Mobilität erwarten die Benutzer aber auch, dass ihre persönlichen Kommunikationsdienste überall auf der Welt mit jeglichem Endgerät von unterschiedlichen Zugangnetzen aus verfügbar sind.

Die Dienstanbieterung in heutigen Mobilfunknetzen ist von ihrem Funktionsumfang her limitiert, da keine offenen Dienstausführungsumgebungen existieren und Dienste in der Regel zentral im Kommunikationsnetz erbracht werden. Die Konvergenz von Mobilnetzen mit dem Internet ermöglicht die neue Dienstumgebung *Mobiles Internet*, die jedoch weit mehr

beinhalten sollte als einen mobilen Zugang zu Webseiten. Für diese Dienstumgebung werden insbesondere dynamische und auf den Teilnehmer und sein Endgerät zugeschnittene Kommunikationskonzepte benötigt, die in Hinblick auf ihre Leistung und Skalierbarkeit eine dateneffiziente Dienstangebotung in heterogenen Kommunikationsnetzen (ISDN, GSM, WLAN, UMTS, etc.) erlauben. Dabei müssen insbesondere die mobilfunkeigenen Rahmenbedingungen wie geringe Übertragungskapazität, hohe Verzögerungen, eingeschränkte Verfügbarkeit, wechselnde Endgeräte und geografische Standorte, sowie limitierte Benutzerschnittstellen ausreichend berücksichtigt werden.

Ob mittels agentenbasierter Kommunikationskonzepte eine intelligente und ressourceneffiziente Dienstangebotung in Mobilfunknetzen realisiert werden kann, muss vor allem in Funktion, Leistung und Akzeptanz untersucht werden. Während die Funktionstüchtigkeit anhand der Konzeptionen und Prototypen verifiziert werden kann, müssen für eine Leistungsbetrachtung die verfügbaren Bandbreiten und Verarbeitungskapazitäten berücksichtigt werden, um sinnvolle Aussagen über das Auftreten von Flaschenhalseffekten tätigen zu können.

In der vorliegenden Arbeit wird eine agentenbasierte Kommunikationsplattform konzipiert und entwickelt, welche die Rahmenbedingungen moderner Mobilfunknetze berücksichtigt und die Anbietung von innovativen Diensten des mobilen E-Commerce (M-Commerce) erlaubt. Für die Funktionsprüfung der Kommunikationsplattform werden zwei agentenbasierte M-Commerce-Dienste konzipiert, implementiert und analysiert. Die Leistungsfähigkeit der Agententechnologie für Mobilfunksysteme und insbesondere der entworfenen Kommunikationsplattform wird anschließend mittels Messungen und eines mathematischen Modells untersucht und bewertet.

1.2 Vorgehensweise

Die vorliegende Arbeit besteht aus sieben Kapiteln. Zunächst werden in Kapitel 2 die Grundlagen moderner Kommunikationssysteme erläutert. Dabei werden neben dem Mobilfunkstandard GSM auch intelligente Dienstarchitekturen wie IN, CAMEL, VHE und WAP kurz beschrieben. Ferner wird in den paketorientierten GPRS-Mobilfunkstandard eingeführt, dessen Einsatz bei der Bewertung des konzipierten agentenbasierten Kommunikationskonzeptes ebenfalls berücksichtigt wird.

Im dritten Kapitel wird das wissenschaftliche Fachgebiet der Agententechnologie vorgestellt. Es werden technische Begriffsdefinitionen vorgenommen, die für das weitere Verständnis der agentenbasierten Dienstangebotung notwendig sind. Weiterhin wird untersucht, welche Typen von Softwareagenten in Mobilfunknetzen Verwendung finden und welche Mechanismen für die agentenbasierte Kommunikation eingesetzt werden können. Abschließend werden die in dieser Arbeit verwendeten Agentensysteme (Voyager, Grasshopper) und Standardisierungsaktivitäten (FIPA, OMG) im Bereich der Agententechnologie diskutiert.

Kapitel 4 beschäftigt sich damit, wie die verteilte CORBA-Architektur die agentenbasierte Dienstangebotung ergänzen kann. Hierzu werden die Probleme von existierenden Agentensystemen aufgezeigt und eine Lösung mit Hilfe eines CORBA ORBs, des OMG-Standards MASIF und zweier Agentensysteme entworfen und realisiert.

Anforderungen an eine agentenbasierte Dienstangebotung in Mobilfunknetzen, sowie die Konzeption einer agentenbasierten Kommunikationsplattform und von zwei M-Commerce-

Diensten bilden den Schwerpunkt von Kapitel 5. Hierzu wird die neue Agentenkommunikationsplattform MCAP konzipiert und implementiert. Der MCAP erlaubt Benutzer- und Endgerätemobilität, unterstützt die modernen Anwendungsprotokolle WAP, HBCI, SQL und SSL und bildet den Kern der beiden M-Commerce-Dienste. Während der agentenbasierte Homebankingdienst dabei die Vorteile mobiler Agenten zur effizienten und gebündelten Übertragung von Bankaufträgen nutzt, werden beim agentenbasierten Such- und Bestelldienst mobile Suchagenten für die Vorbereitung und Durchführung von privaten Einkäufen eingesetzt werden.

In Kapitel 6 wird zunächst ein neues analytisches Modell zur Untersuchung und Bewertung des agentenbasierten Kommunikationskonzeptes aufgestellt. Hierfür wird das Dekompositionsverfahren angewandt. Mittels des Modells kann das agentenbasierte Kommunikationskonzept mit traditionellen Dienstleistungskonzepten verglichen werden. Hierbei werden insbesondere die Kommunikationszeit mit Hilfe von Warteschlangenmodellen untersucht. Ferner werden Modelle zur Mobilität, Selektivität und Interaktivität von mobilen Agenten entwickelt. Der zweite Teil des Kapitels befasst sich mit den Ergebnissen der Analyse. Dabei werden die Antwortzeiten, die übertragenen Datenvolumina und die anfallenden Kommunikationskosten für die beiden M-Commerce-Dienste getrennt untersucht, diskutiert und bewertet. Schließlich werden die Ergebnisse für die kanalarorientierte (GSM) und paketorientierte (GPRS) Mobilkommunikation miteinander verglichen.

Kapitel 7 bildet den Abschluss der Arbeit. Hier werden die wichtigsten Erkenntnisse aus den Untersuchungen zusammengefasst. Abschließend wird ein Ausblick auf weitere Forschungsaktivitäten im Bereich der agentenbasierten Dienstleistung getätigt.

Dienstanbietung in Telekommunikationsnetzen

Trotz umfangreicher Bemühungen sind Telekommunikationsdienste in verteilten, heterogenen Kommunikationsnetzen bis heute häufig nicht global verfügbar. In diesem Kapitel wird zunächst der Dienstbegriff erläutert. Danach werden die Konzepte zur Dienstanbietung in modernen Kommunikationsnetzen, insbesondere den Mobilfunknetzen, vorgestellt.

2.1 Der Dienstbegriff

Dienste spielen eine herausragende Rolle in der Telekommunikation. So wird dem Markt für Telekommunikationsdienste ein großes Wachstumspotenzial vorausgesagt [MV00].

Die **Telekommunikationsdienste** eines Telekommunikationsnetzes sind die Informationen und Kommunikationsmöglichkeiten, die ein Netzbetreiber seinen Kunden zur Verfügung stellt.

Technisch gesehen sind Telekommunikationsdienste traditionell zentralgesteuert. Das bedeutet, dass die Erbringung von Diensten fast ausschließlich innerhalb des Kommunikationsnetzes stattfindet [Geu98]. Dies gilt insbesondere für Dienste des Intelligenzen Netzes bzw. des CAMEL¹-Mechanismus, die in den folgenden Kapiteln diskutiert werden. Im Hinblick auf das ISO/OSI-Referenzmodell [Wal97] werden Telekommunikationsdienste in zwei Klassen unterteilt, siehe auch Abb. 2.1:

Trägerdienste: Trägerdienste (*bearer services*) sind Telekommunikationsdienste, die lediglich die Fähigkeiten zur Übermittlung von Signalen zwischen verschiedenen Netzzugangspunkten (*network access points*) zur Verfügung stellen. Durch die Trägerdienste wird das physikalische Weiterleiten und Vermitteln der Kommunikationsnachrichten ermöglicht. Das Konzept der Trägerdienste wurde für ISDN²-Netze entwickelt und auf Mobilfunknetze ausgeweitet [ETS99a]. Ein GSM-Mobilfunknetz³ unterstützt verschiedene Datenübertragungsklassen, wie z.B. 1.200 *bit/s*, 2.400 *bit/s*, 4.800 *bit/s* oder 9.600 *bit/s*. Trägerdienste nutzen Funktionen der unteren drei Schichten des ISO/OSI Referenzmodells.

Teledienste: Teledienste (*teleservices*) sind Telekommunikationsdienste, die die komplette Funktionalität der ISO/OSI Schichten 1-7 nutzen. Dies schließt insbesondere Funktionen in den Endgeräten ein (z.B. Sprachwahl). Weiterhin erlauben Teledienste die Kommunikation zwischen fremdartigen Netzen, z.B. zwischen Mobilfunk- und Festnetzen. Beispiele für Teledienste sind Telefonie, Faksimile, Gruppenrufe und der SMS⁴-Kurznachrichtendienst.

¹Customized Applications for Mobile networks Enhanced Logic: wird in Kap. 2.2.3 vorgestellt

²Integrated Services Digital Network: digitales dienstintegrierendes Netz

³Global System for Mobile communication: der weltweit meist verwendete Mobilfunkstandard, eine genauere Betrachtung erfolgt in Kap. 2.2.2

⁴Short Message Service: Möglichkeit Textnachrichten zu senden und zu empfangen

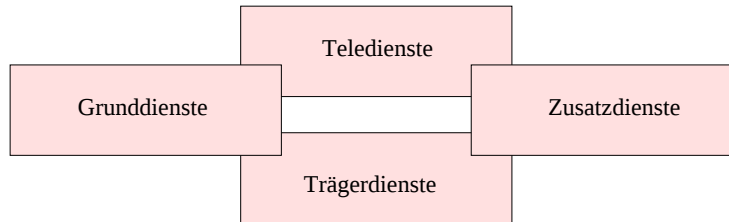


Abbildung 2.1: Telekommunikationsdienste

Grundsätzlich lassen sich Telekommunikationsdienste gemäß [ITU93] auch in *Grunddienste* und *Zusatzdienste* kategorisieren.

Grunddienste: Grunddienste (*basic services*) sind die Fähigkeiten eines Kommunikationsnetzes, die es z.B. ermöglichen, einen Anruf entgegenzunehmen und zu beenden. Im Rahmen von GSM werden als Grunddienste lediglich die Übertragung von Sprache, Daten oder Fax verstanden [Geu98].

Zusatzdienste: Zusatzdienste (*supplementary services*) ergänzen oder verändern einen Grunddienst [ETS99b]. Daher kann ein Zusatzdienst auch nicht als alleiniges Produkt einem Kunden angeboten werden, sondern wird stets zusammen oder in Verbindung mit einem Grunddienst im Rahmen eines Vertrages angeboten. Weiterhin kann ein Zusatzdienst auch für alle Teilnehmer⁵ im Kommunikationsnetz verfügbar gemacht werden. Beispiele für Zusatzdienste im GSM-Netz sind Anrufweiterleitung, Anklopfen, Mailbox und der SMS-Kurznachrichtendienst.

Dienste sind das eigentliche Ziel der Telekommunikation. Und hier sind es insbesondere die Teledienste, die Dienstleister (*service provider*) und Dienstnutzer gleichermaßen interessieren. Während der Dienstnutzer heute von seinem mobilen Endgerät nicht nur die gleichen Telekommunikationsdienste wie im Festnetz, sondern darüber hinaus spezielle multimediale und kontextabhängige (z.B. orts-, zeit- oder endgeräteabhängige) Mobilfunkdienste wünscht, haben die Dienstleister erkannt, dass sich die Wertschöpfungskette in der Telekommunikationsbranche erheblich verändert hat. Entstanden in der Vergangenheit Telekommunikationskosten fast ausschließlich durch die Trägerdienste basierend auf den verbrauchten Telefoneinheiten, so werden heute die unterschiedlichsten Telekommunikationsgebühren in Rechnung gestellt:

- *Zugangsgebühren* (access charges):
Diese fallen z.B. dann an, wenn der Benutzer einen Zugang zu einem Internetanbieter (T-Online, AOL, etc.) anwählt.
- *Dienstgebühren* (service charges):
Dienstgebühren werden berechnet, wenn der Benutzer spezielle Zusatzdienste (z.B. Premium Rate, siehe Kap. 2.2.1) nutzt. So wird auch die Nutzung des Internets häufig mit einem zusätzlichen Minutenpreis tarifiert.

⁵Teilnehmer (*subscriber*) nutzen Dienste auf Basis einer Teilnehmerkennung und stellen die Dienste einem Benutzer zur Verfügung. Teilnehmer können Personen oder Organisationen sein. Teilnehmer und Benutzer können auch ein und dieselbe Person sein [Gun96].

- *Speicherplatzgebühren* (hosting charges):
Möchte der Benutzer Webseiten ablegen, so werden ihm in der Regel Speicherplatzgebühren berechnet.
- *Spezielle Internetleistungen* (special charges):
Für die Verwaltung von E-Commerce-Diensten sind oft spezielle Sicherheits-, Datenbank- oder Abrechnungsdienste erforderlich, die gesondert in Rechnung gestellt werden.
- *Informationsgebühren* (pay per view):
Im Bereich Informationsmanagement/E-Commerce ist es inzwischen üblich, dass Informationspreise anstatt Minutenpreise berechnet werden (z.B. telefonische Auskunftsdienste, SMS, imode⁶).

Aufgrund dieser nachhaltigen Veränderungen erforschen und entwickeln Netzbetreiber und Hersteller von Kommunikationsinfrastruktur mit großem Einsatz zukünftige E-Commerce-Dienste. E-Commerce wird dabei wie folgt definiert:

E-Commerce ist derjenige Teil des E-Business, der auf die Vereinbarung und Abwicklung rechtsverbindlicher Geschäftstransaktionen (Information über Produkte und Dienstleistungen, Angebotserstellung, Vertragsaushandlung, Auftragserteilung, Lieferung und Zahlung, etc.) ausgerichtet ist.

Dabei wird unter **E-Business** die Abwicklung von Geschäftsprozessen mit Hilfe von elektronischen Informations- und Kommunikationstechnologien verstanden.

Für Entwickler und Hersteller von Telekommunikationsinfrastruktur ist es von besonderem Interesse, welchen Einfluss neue E-Commerce-Dienste auf bereits existierende Dienste haben werden, damit die angebotenen Dienste nach wie vor zuverlässig, d.h. z.B. mit vernachlässigbaren Ausfallzeiten, zur Verfügung gestellt werden können. Außerdem werden Mechanismen gesucht, um diese neuen Dienste personalisiert anbieten zu können.

Ein **personalisierter Dienst** ist ein Teledienst, der auf einen bestimmten Teilnehmer und seine Bedürfnisse, d.h. auf seine Präferenzen, das von ihm verwendete Endgerät und seinen Aufenthaltsort, zugeschnitten ist.

Personalisierte Dienste sollten auch nach ihrer Einrichtung in der Lage sein sich weiter an den Teilnehmer und seine Vorlieben anzupassen. Netzbetreiber und Handelsunternehmen können so mit Hilfe von personalisierten Diensten eine individuelle Kundenbetreuung (CRM⁷) auch über elektronische Kommunikationsmedien durchführen.

2.2 Moderne Kommunikationskonzepte

In den vergangenen Jahren wurde in den Forschungs- und Entwicklungslaboren der Telekommunikationssystemhersteller mit besonderem Nachdruck an Lösungen für die intelligente Diensterbringung gearbeitet. Das erste Produkt dieser Anstrengungen mit weltweiter Akzeptanz war der ITU-Standard Q.12xx für das Intelligente Netz (IN). Aufgrund des Erfolges der GSM-Mobilfunknetze entstanden vor Abschluss der IN-Standardisierung vergleichbare Lösungen, die insbesondere die Bedürfnisse und Rahmenbedingungen von

⁶imode ist der mobile Internetzugang von NTT DoCoMo in Japan, den über zehn Millionen Teilnehmer nutzen. Der Dienst wird nicht über einen Minutentarif, sondern über Informationsgebühren abgerechnet.

⁷Customer Relationship Management

Mobilfunkbenutzern berücksichtigen, wie CAMEL, VHE⁸ und WAP⁹, die zwar Meilensteine in der Entwicklung von Mobilfunknetzen sind, aber erst den Anfang von neuartigen Dienstleistungsvarianten darstellen.

2.2.1 Intelligente Netze

Intelligentes Netz, Mehrwertdienst (VAS¹⁰) und Virtuelles Privatnetz (VPN¹¹) sind Begriffe, die häufig im Zusammenhang mit dem liberalisierten Telekommunikationsmarkt und den damit verbundenen Vorteilen fallen. Diese Begriffen verkörpern die Idee, die Dienstlogik nicht im Übermittlungsnetz, sondern an davon unabhängigen Netzknoten, den sogenannten Dienststeuerknoten (SCP¹²) zu konzentrieren [Gun96]. Dies führt zur Einführung einer zusätzlichen, dienstorientierten Architektur, die auf der bereits vorhandenen Netzstruktur des Übermittlungsnetzes aufsetzt, aber von dieser weitgehend unabhängig ist. Tab. 2.1 zeigt, dass die Spezifikation und Umsetzung dieses Konzeptes eine lange Geschichte hat, deren Anfänge in der Mitte der 80er Jahre liegen.

Jahr	Konzept	Bedeutung
1986	IN/1	Belcore, USA, zentrale Datenbanken
1986	IN/2	Belcore, Erweiterung der IN/1 Funktionalität
1988	IN/1+	Belcore, Untermenge von IN/2
1991	AIN ¹³ Rel. 0	Belcore, Flexibilität in der physikalischen Realisierung
1992	IN-CS1	Capability Set 1, CCITT ¹⁴ -Standard
1993	TINA-C ¹⁵	Hauptziel: Integration von IN und TMN ¹⁶
1995	IN-CS2	Capability Set 2, Erweiterung von IN-CS1
1995	CTM ¹⁷	Integration von IN und DECT ¹⁸
1995	CAMEL	Integration von IN und GSM
1997	VHE	Integration von CAMEL, SAT ¹⁹ und MExE ²⁰
1998	WAP	Integration von Mobilfunk und Internet

Tabelle 2.1: Historische Entwicklung von Intelligenten Netzen

Während das IN-Konzept zunächst für öffentliche Festnetze (PSTN²¹ und ISDN) entwickelt wurde, griff die Europäische Standardisierungsbehörde ETSI²² 1994 den Gedanken im Rahmen der Entwicklung von GSM Phase 2+ auf und startete die Standardisierung von CAMEL, die in Kap. 2.2.3 vorgestellt wird.

⁸Virtual Home Environment: siehe Kap. 2.2.4

⁹Wireless Application Protocol: siehe Kap. 2.2.5

¹⁰Value Added Service

¹¹Virtual Private Network

¹²Service Control Point

¹⁴Advanced IN

¹⁵Comité Consultatif International de Télégraphique et Téléphonique, International Telegraph and Telephone Consultative Committee, heute: ITU

¹⁶Telecommunication Information Network Architecture - Consortium

¹⁷Telecommunication Management Network

¹⁸Cordless Terminal Mobility

¹⁹Digital Enhanced Cordless Telecommunication

²⁰SIM Application Toolkit

²¹Mobile station application Execution Environment

²¹Public Switched Telecommunication Network

²²European Telecommunication Standards Institute

Die Hauptziele, die aus der Sicht der Netzbetreiber mit dem Intelligenten Netz erreicht werden sollen, sind:

- Erweiterung des Dienstangebotes,
- schnelle, einfache und ökonomische Entwicklung, Prüfung und Einführung von neuen Diensten, deren Administration, sowie die Modifikation von existierenden Diensten,
- effiziente Nutzung von verfügbaren Verarbeitungstechniken und Netzressourcen,
- Wiederverwendung von Netzfunktionen,
- flexible Zuordnung von Netzfunktionen zu physikalischen Netzknoten und Unabhängigkeit vom Netzzugang,
- Einsatz von standardisierten Schnittstellen.

Aus der Sicht des Anwenders bietet das Intelligente Netz eine Vielzahl von komplexen Komfortmerkmalen. Die bekanntesten IN-Dienste in Deutschland sind Televoting (Rufnummernpräfix 0137), Hotline (0180), Premium Rate (0190), Persönliche Rufnummer (0700), Freephone (0800) und Virtuelle Privatnetze (VPN). Ferner wird IN für die sogenannte Fest- und Mobilnetzkonvergenz eingesetzt [BFGH00].

Obwohl sich das IN-Konzept technologisch und kommerziell in Teilbereichen (siehe oben) durchgesetzt hat, ist die Verwendung der IN-Technologie für zukünftige Kommunikationssysteme begrenzt, da man zum Zeitpunkt der Standardisierung den Erfolg der zellularen Mobilfunksysteme und des Internets noch nicht absehen konnte und daher beide Systeme nicht mit in die Standardisierung einbezogen wurden.

2.2.2 GSM-Mobilfunknetze

Die Standardisierung von GSM (Global System for Mobile communications) wurde 1982 von der CEPT²³ unter dem Namen *Groupe Special Mobile* begonnen. 1989 wurde die Verantwortung für den GSM-Standard an die ETSI²⁴ übergeben. Phase 1 des Standards wurde 1991, Phase 2 1994 abgeschlossen. Phase 2 führte ein umfangreicheres Angebot an Diensten, größtenteils Zusatzdienste, ein [Geu98]. 1992 wurden die ersten Netze in Betrieb genommen [Eve99], der kommerzielle Erfolg von GSM setzte schon bald nach der Installation der ersten Systeme in Europa ein. Bereits 1993 gab es 36 GSM-Netze in 22 Ländern. Diese Zahlen stiegen kontinuierlich weiter; im Mai 2000 existieren 357 Netze in über 150 Ländern der Welt [Fun00] mit rund 500 Millionen Teilnehmern. Die ständig steigende Nachfrage nach Funkkapazität in GSM-Systemen wird meist dadurch befriedigt, dass man Antennen und Sender mit geringer Leistung in kleineren Zellen einsetzt²⁵. Die wichtigsten Systemcharakteristika kann man Tab. 2.2 entnehmen.

Das GSM 1800 System²⁶ arbeitet nach dem gleichem Prinzip, allerdings auf dem Frequenzband 1710 bis 1785 MHz (*uplink*) bzw. 1805 bis 1880 MHz (*downlink*). In den USA existiert mit dem GSM 1900 System eine weitere Variante des Standards. Grundsätzlich ist der Uplink bei GSM-Systemen gegenüber dem Downlink um drei Zeitschlitzte (*time slots*) verzögert. Dies ist wichtig für den Entwurf der mobilen Endgeräte, in denen, anders als bei analogen Mobilfunksystemen, Sender und Empfänger niemals zur gleichen Zeit arbeiten.

²³Conférence Européenne des Posts et Télécommunications

²⁴European Telecommunication Standards Institute

²⁵das deutsche D2-Netz ist heute in mehr als 20.000 Zellen eingeteilt

²⁶in Deutschland: E-Netze

Frequenz Uplink	890 MHz - 915 MHz
Frequenz Downlink	935 MHz - 960 MHz
Duplexdistanz	45 MHz
Trägerabstand	200 kHz
Modulation	GMSK ²⁷
Übertragungsrate	270 kbit/s
Zugangsmethode	TDMA ²⁸
Sprachdatenrate	13 kbit/s
Frequenzen	124
Vervielfältigungsmethode	Verschachtelung ²⁹

Tabelle 2.2: Systemcharakteristika GSM 900

Mobile Kommunikationssysteme wie GSM bestehen aus einem Festnetz- und einem Mobilfunkteil, wobei die Mobilitätsverwaltung auf Funktionen im Festnetz mit Hilfe des weitverbreiteten Signalisierungsprotokolls SS7³⁰ beruht. Das GSM-Mobilfunksystem ist gut dokumentiert [Wal00], so dass nachfolgend nur die elementaren Bestandteile des Konzeptes vorgestellt werden. Abb. 2.2 zeigt die Architektur eines Mobilfunknetzes nach dem GSM-Standard.

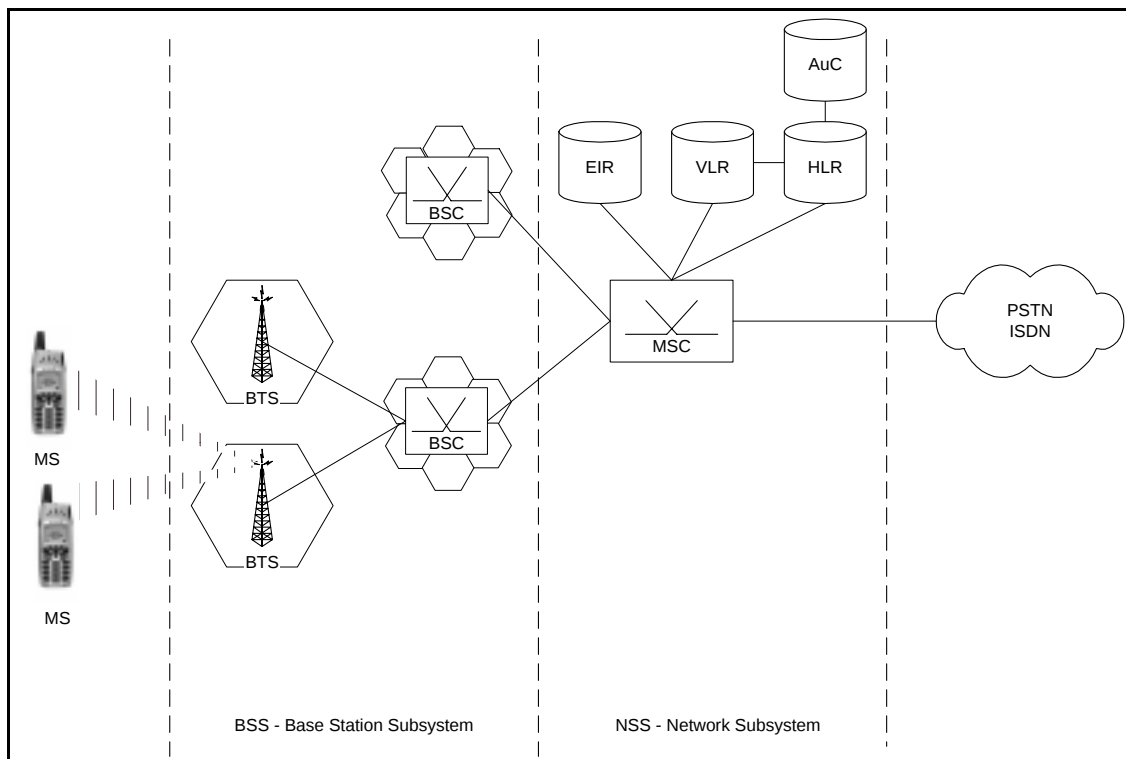


Abbildung 2.2: Die logische GSM-Architektur

²⁷ Gaussian Minimum Shift Keying

²⁸ Time Division Multiple Access

²⁹ Interleaving

³⁰ Das Signalisierungssystem Nr. 7 ist in den ITU-Standards Q.700 bis Q.795 spezifiziert und wird heute in nahezu allen öffentlichen Telekommunikationsnetzen eingesetzt.

Das GSM-Dienstgebiet ist in verschieden große Zellen eingeteilt, wobei jede Zelle durch eine Funkstation versorgt wird. Eine der wichtigsten Routinen im GSM-System ist der sogenannte *Handover*, bei dem bestehende Kommunikationsbeziehungen von einer Zelle an eine andere übergeben werden. Ein Handover kann nicht nur im Heimatnetz, das der Teilnehmer aufgrund eines Vertrages mit seinem Netzbetreiber benutzen kann, sondern auch im Fremdnetz, dem sogenannten besuchten Mobilfunknetz, stattfinden. Hierzu wurden Registrierungsmechanismen entwickelt, bei denen teilnehmerbezogene, aktuelle Daten in der Heimatdatei (HLR³¹) und in der Besucherdatei (VLR³²) verwaltet werden. Die damit erreichte Möglichkeit des *Roamings*³³ von Teilnehmern in beliebigen GSM-Netzen mit voller Dienstverfügbarkeit wird oft als Hauptgrund für den großen kommerziellen Erfolg von GSM angesehen.

Die GSM-Kommunikation basiert auf der Trennung von Nutz- und Signalisierungsinformation. Dabei wurden beim Entwurf von GSM die Prinzipien und die Protokolle von ISDN adaptiert. Zusätzliche Funktionen sind im *Mobile Application Part* (MAP) Protokoll standardisiert [ETS98b]. Für die Signalisierung auf dem Funkkanal ist das *Base Station Subsystem* (BSS) verantwortlich. Aus ökonomischen Gründen besteht es aus zwei Funktionsgruppen, nämlich den Basisstationen (BTS³⁴) und den übergeordneten Basisstationssteuerungen (BSC³⁵). Ein typisches GSM-Netz besteht dabei aus Tausenden von BTSs und mehreren Dutzend BSCs. Der Funktionsumfang einer BTS wurde daher recht gering gehalten. Die Steuerung der Nutzung von Funkkanälen und manchmal auch Vermittlungsfunktionen werden vom BSC übernommen.

Die anderen funktionellen Elemente eines GSM-Netzes werden im *Network and Switching Subsystem* (NSS) zusammengefasst. Wichtigster Bestandteil des NSS ist die mobile Vermittlungsstelle (MSC³⁶), welche Mobilfunkgespräche verbindet und die besonderen Rahmenbedingungen eines Mobilfunknetzes berücksichtigt. Jede MSC steuert und kommuniziert mit mehreren BSCs. Im Falle eines Anrufes zu einem Festnetzteilnehmer wird der Anruf an eine Vermittlungsstelle weitergeleitet, die mit einem Protokollumsetzer³⁷ (*gateway*) ausgestattet ist (GMSC³⁸). Neben HLR und VLR sind das *Authentication Centre* (AuC) und das *Equipment Identity Register* (EIR) weitere Bestandteile des NSS.

In den vergangenen Jahren wurde die Entwicklung von datenbasierten Anwendungen forciert, doch das GSM-System ist für viele Anwendungen lediglich bedingt geeignet. So leidet die mobile Datenkommunikation im Vergleich zur kabelgebundenen Datenübertragung insbesondere unter den folgenden Einschränkungen, die bei der Konzeption von zukünftigen Telediensten und Kommunikationskonzepten berücksichtigt werden müssen:

- geringe Übertragungskapazität
- hohe Verzögerung
- geringe Zuverlässigkeit
- hohe Bitfehlerrate
- eingeschränkte Verfügbarkeit

³¹Home Location Register

³²Visitor Location Register

³³wörtliche Übersetzung: Umherziehen

³⁴Base Transceiver Station

³⁵Base Station Controller

³⁶Mobile service Switching Centre

³⁷Netzknoten zur Kopplung von verschiedenen Kommunikationsnetzen

³⁸Gateway MSC

2.2.3 CAMEL-Standard

CAMEL ist kein neuer Telekommunikationsdienst, sondern eine Erweiterung der Netzfunktionalität. Mit Hilfe von CAMEL können betreiberspezifische Dienste auch im Roaming-Fall, d.h. wenn der Teilnehmer sich in einem Fremdnetz befindet, ausgeführt werden. Im CAMEL-Standard wurden keine Dienste, sondern ein Werkzeug standardisiert mit dem neue Dienste erstellt werden können. Die aus dem Festnetz bekannten IN-Dienste (Kap. 2.2.1) können mit CAMEL realisiert werden.

In Deutschland gibt es momentan neben Netzbetreibern eine Vielzahl von sogenannten Dienstanbietern wie z.B. Cellway, Debitel, Dplus, Hutchison, Mobilcom, Netztel, Talkline, Victor Vox, welche sich lediglich durch geringfügige Modifikationen in ihren Tarifmodellen unterscheiden. Um dem steigenden Bedürfnis der Dienstunterscheidung Rechnung zu tragen, wurde 1993 die Standardisierung von CAMEL zunächst unter dem Arbeitstitel *Unterstützung von netzbetreiberspezifischen Diensten im Roaming-Fall* aufgenommen. 1994 wurde der Name CAMEL gewählt.

CAMEL ergänzt die GSM-Architektur (Abb. 2.2) mit einer IN-typischen Schnittstelle zwischen Vermittlungsfunktionen (SSF³⁹) und Dienststeuerung (SCF⁴⁰) (Kap. 2.2.1). Dabei wird die SSF-Funktionalität in der mobilen Vermittlungsstelle integriert. Diese SSF-Funktion, die in CAMEL *gsmSCF* genannt wird, ist physikalisch in einem Knoten realisiert, der *CAMEL Service Environment* (CSE) heißt. Als Grundlage für die Standardisierung von CAMEL diente der ITU IN-Standard *Capability Set 1* (CS-1) [Har98].

CAMEL Subscription Information (CSI) ist der Datensatz, der jedem Teilnehmer zugeordnet wird, der CAMEL Unterstützung benötigt. Dieser Datensatz wird als Teil der GSM-Teilnehmerdaten in der Heimatdatei (HLR) und in der Besuchsdatei (VLR) gespeichert. Er enthält einen SS7 SCCP⁴¹ Global Title, um das für einen bestimmten Teilnehmer verantwortliche CSE eindeutig adressieren zu können. Weiterhin ist im CSI abgelegt, welche betreiberspezifischen Dienstlogiken angewendet werden sollen und was im Fehlerfall geschehen soll.

Die in Abb. 2.3 dargestellten funktionellen Einheiten wurden für die CAMEL-Architektur wie folgt definiert [GH97]:

- **HLR** ist verantwortlich für das Dienstprofil (CSI) des Teilnehmers; im Falle eines location updates oder wenn sich das CSI verändert hat, sendet das HLR das CSI zum VLR,
- **VLR** speichert das CSI des Teilnehmers,
- **CSE** ist die Schnittstelle zwischen dem nachfragenden und dem besuchten Netz,
- **VMSC**⁴² führt die Dienste entsprechend des CSI aus, wenn ein Teilnehmer CAMEL Unterstützung im besuchten Netz benötigt,
- **VSSF**⁴³ ist die Schnittstelle zwischen VMSC und CSE und erkennt einen IN-Dienstwunsch eines Fremdnetzteilnehmers,

³⁹Service Switching Function

⁴⁰Service Control Function

⁴¹Signalling Connection Control Part, SS7-Protokollebene 4, Protokoll der ISO/OSI-Schicht 3, ist in den ITU-Standards Q.711-714 spezifiziert.

⁴²Visited Mobile service Switching Centre

⁴³Visited Service Switching Function

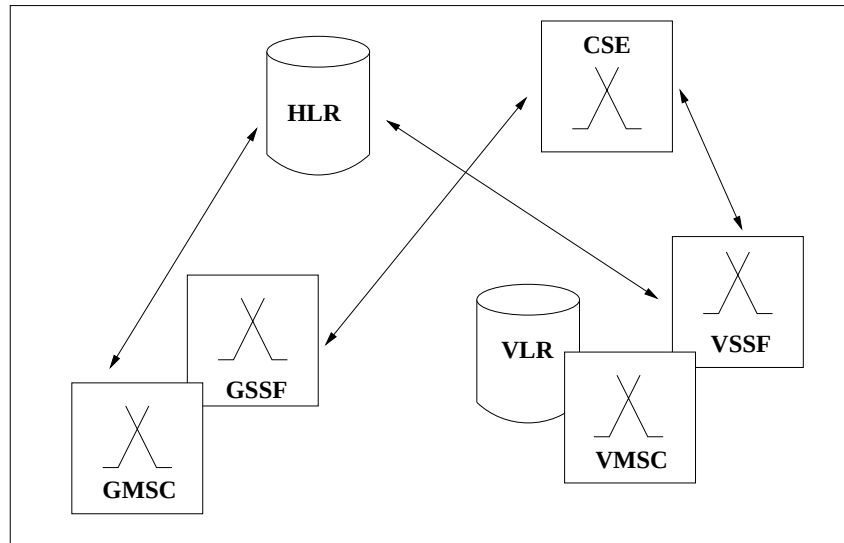


Abbildung 2.3: Die logische CAMEL-Architektur

- **GMSC** führt die Dienste entsprechend dem CSI aus, wenn ein Teilnehmer im Heimatnetz CAMEL Unterstützung benötigt,
- **GSSF**⁴⁴ ist die Schnittstelle zwischen VMSC und CSE und hilft beim Erkennen von IN-Diensten.

2.2.4 Virtual Home Environment

Das Konzept des *Virtual Home Environments* (VHE) soll Mobilfunkteilnehmern ermöglichen, immer dieselben, personalisierten, d.h. auf sie zugeschnittenen, Dienste und Dienstmerkmale nutzen zu können, egal an welchem Ort und in welchem Netz sie sich befinden und auch weitgehend unabhängig davon, welches Endgerät verwendet wird. Mit Hilfe seines persönlichen Identifikationsmoduls (z.B. Chipkarte) weist sich der Teilnehmer gegenüber Endgerät, Netz und Dienst aus. Die exakte Konfiguration der Dienste hängt dabei von den Möglichkeiten des Netzes ab, an welches der Benutzer gerade angeschlossen ist und selbstverständlich auch von den Fähigkeiten des gerade benutzten Endgerätes [BFGH00].

Das standardisierte VHE-Konzept bietet allgemeine Dienstmerkmale (*service capabilities*), mit denen spezifische Dienste und Leistungsmerkmale erstellt werden können, die über alle UMTS- oder GSM-Netze und Endgeräte genutzt werden können [3GP00]. Erste Standards wurden bereits 1997 für GSM erarbeitet und werden nun für UMTS in einem erweiterten Umfang weiterentwickelt. Diese Standardisierungsbemühungen sind bis heute noch nicht abgeschlossen. Die derzeit angedachte Lösung basiert auf den GSM-Standards CAMEL, SAT und MExE. Die Konzeptideen von VHE sind jedoch nicht auf den Mobilfunk begrenzt, sondern berücksichtigen vielmehr auch Anwendungen in Festnetzen. VHE erlaubt es damit, Dienste unabhängig vom Kern- und Zugangsnetz anzubieten.

Das VHE eines individuellen Teilnehmers ist in der Regel eine Untermenge der Dienste, welche der Dienstanbieter zur Verfügung stellt. Der Benutzer wünscht sich meist ein individuelles, netzunabhängiges Erscheinungsbild seiner Kommunikationsdienste. Daher benötigt das Netz auch keine Detailkenntnis über die mittels VHE realisierten Dienste des

⁴⁴Gateway Service Switching Function

angeschlossenen Benutzers. Teilnehmer und Dienstleister von VHE nutzen lediglich die Transport- und Signalisierungsdienste sowie die Ressourcen des Kommunikationsnetzes. Bei der Realisierung von VHE hängt es letztendlich vom Netzbetreiber, vom Dienstleister und vom benutzten Endgerät ab, ob ein entsprechender Dienst oder ein Dienstmerkmal auch wirklich verfügbar ist.

Dem Dienstleister erscheint das VHE als eine Liste von Ressourcen, Optionen und Einstellungen für einen Benutzer. Einige dieser Daten werden vom Dienstleister verwaltet, weil sie Bestandteil des Vertrages mit dem Teilnehmer sind. Andere Daten werden vom Benutzer oder Teilnehmer auf seinem Identifikationsmodul *Universal Subscriber Identity Module* (USIM) oder im Endgerät gespeichert und verwaltet. Prinzipiell besteht das VHE aus den folgenden funktionalen Einheiten:

- Benutzerschnittstelle (*user interface*),
- Dienstausführungsprogramm (*service execution program*),
- Ausführungsumgebung (*application execution environment*), die als Plattform für die Dienstausführung und den Zugriff auf Kommunikationsdienste verantwortlich ist. Diese Plattform kann mittels standardisierter Anwendungsschnittstellen (API⁴⁵s) angesprochen werden, die z.B. auf objektorientierten Technologien (Programmiersprache JAVA oder CORBA⁴⁶-Standard), beruhen können,
- individuelles Dienstprofil (*service profile*), das teilnehmer- und netzspezifische Daten enthält, die vom Programm benutzt werden,
- Verbindungssteuerung (*connection control*), die zur Steuerung von Transportdiensten und zur Mobilitätsverwaltung dient.

Daten und Programme zur Ausführung des VHE können sich im Identifikationsmodul des Benutzers (z.B. Chipkarte), im Endgerät, im Fremdnetz, im Heimatnetz und beim Dienstleister befinden. Die Programme und Daten werden bedarfsweise in das Fremdnetz, das Endgerät und/oder das Identifikationsmodul geladen, jedoch in der Heimatumgebung verwaltet. In allen Elementen ist eine Plattform vorhanden, auf der die Programme des VHE ablaufen können. Diese Plattform greift auf die Verbindungssteuerung und die Mobilitätsverwaltung der beteiligten Netze und des Endgerätes zu. Ausgehend vom o.g. Architekturmodell kann das VHE in der Heimatumgebung, im Fremdnetz, im Endgerät und im Identifikationsmodul ausgeführt werden.

Im Rahmen des Europäischen Forschungsprojektes CAMELEON⁴⁷ [CAM98] wurde eine agentenbasierte Lösung für die VHE-Spezifikation konzipiert, die in Abb. 2.4 grob skizziert ist. Diese Architektur wurde durch drei verschiedene Teledienste - einem intelligenten Kommunikationsmanager (*adaptive profile manager*), einem virtuellen Adressbuch (*virtual address book*) und einem Finanzdienst (*flexible financial service*) - verifiziert. Es hat sich gezeigt, dass mobile Agenten eine flexible Verteilung von Dienstkomponenten zwischen unterschiedlichen Dienstleistungssystemen und Endgeräten erlauben und somit das Konzept der totalen Mobilität, welches später in dieser Arbeit behandelt wird, unterstützen [HMB98, HGF98].

⁴⁵ Application Programming Interface

⁴⁶ Common Object Request Broker Architecture; Standard zur Realisierung von verteilten Anwendungen, eine genauere Betrachtung erfolgt in Kap. 4.2

⁴⁷ Communication Agents for Mobile Enhancements in a Logical Environment of Open Networks; von der Europäischen Union gefördertes ACTS-Forschungsprojekt der Firmen Ericsson Eurolab, Mannesmann Mobilfunk, Nortel Networks, Vodafone, IKV++, Gemplus, RWTH Aachen ComNets und Centre for Wireless Communication; Projektlaufzeit: März 1998 bis Februar 2000.

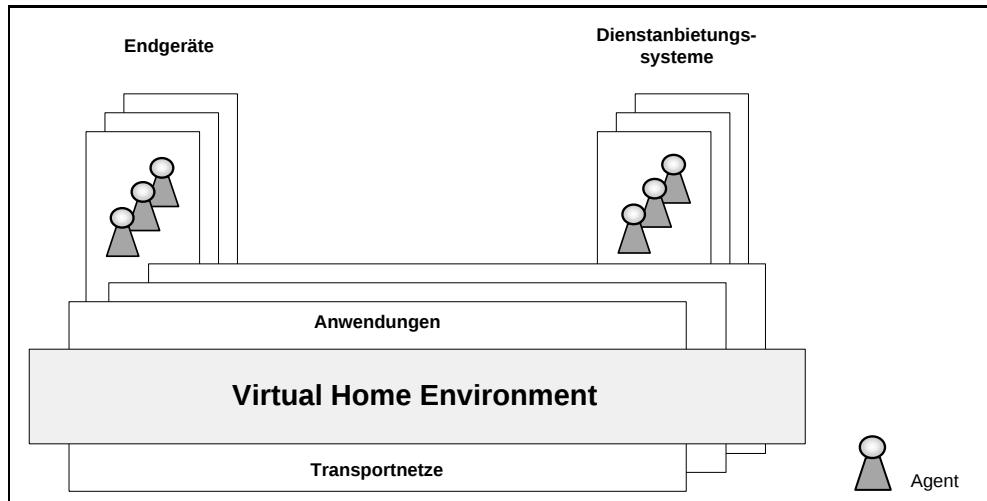


Abbildung 2.4: Agentenbasierte VHE-Architektur

2.2.5 WAP-Standard

Das *Wireless Application Protocol* (WAP) ist der erste Standard, der Beiträge leistet, um das Internet und die mobile Datenkommunikation miteinander zu verbinden. Daher wurde bei WAP besonderer Wert darauf gelegt, existierende Protokolle wieder zu verwenden und diese an die speziellen Rahmenbedingungen mobiler Endgeräte (kleine Bildschirmflächen, limitierte Eingabemöglichkeiten, wenig Speicherplatz etc.) und drahtloser Zugangsnetze (geringe Bandbreite, hohe Bitfehlerrate, erhebliche Verzögerungen etc.) anzupassen. Als Grundlage für den WAP-Standard [For98] dienten drei proprietäre Protokolle für mobile, kleine Endgeräte:

- *Intelligent Terminal Transfer Protocol* (ITTP) von Ericsson,
- *Narrow Band Sockets* (NBS) und *Tagged Text Markup Language* (TTML) von Nokia
- und *Handheld Device Markup Language* (HDML) von Unwired Planet⁴⁸.

Im Juni 1997 gründeten die Unternehmen Ericsson, Nokia, Phone.com und Motorola das WAP Forum, dem heute über 400 Unternehmen angehören. Die Mitgliedschaft im WAP Forum ist offen und die Diskussionsbeiträge aus der Mobilfunkindustrie bestimmten nachhaltig die WAP-Spezifikationen. Die WAP-Architektur wurde bereits im September 1997 veröffentlicht, der erste Entwurf eines Standards folgte im Februar 1998. Die WAP-Version 1.0 wurde schließlich im Mai 1998 verabschiedet. Es folgten die Versionen 1.1 und 1.2 im Juni bzw. Dezember 1999 und die Version 2.0 (mit WAP-Push und Multimedia Messaging) im Juli 2001.

Das Hauptziel von WAP, die Wiederverwendung von existierenden Standards aus der Internetwelt, wurde zunächst konsequent umgesetzt. Dabei läuft eine WAP-Kommunikation, wie Abb. 2.5 zeigt, stets über einen WAP-Gateway, der dabei die folgenden Funktionen übernimmt, die in Kap. 6.3.2 näher betrachtet werden:

1. **Protokollumsetzer:** Der WAP-Gateway übersetzt WAP-Anfragen in HTTP⁴⁹-Anfragen.

⁴⁸heute: Phone.com

⁴⁹Hypertext Transmission Protocol, Anwendungsschichtprotokoll für den Datenaustausch im Internet

2. **Inhaltskodierung und -dekodierung:** Um das zu übertragende Datenvolumen zu verringern, wird der Inhalt einer WAP-Nachricht kodiert. Dies geschieht derart, dass alle bekannten Kommandos (*tags*) durch ein Nachrichtenbyte substituiert werden und Bytecode (WBXML⁵⁰) übertragen wird.

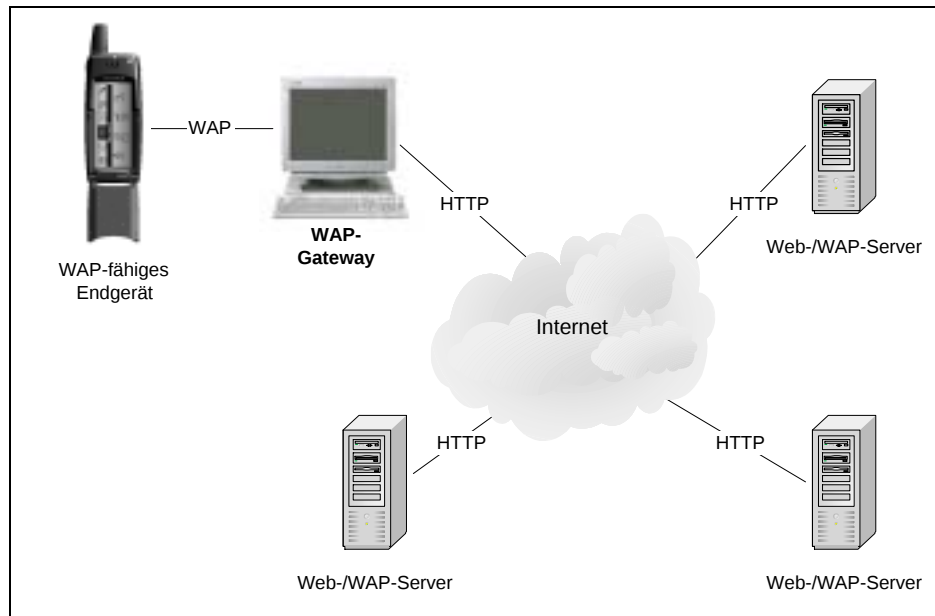


Abbildung 2.5: Funktion des WAP-Gateways

Die WAP-Architektur erlaubt den Entwurf von intelligenten Diensten für mobile Kommunikationsgeräte. Die WAP-Architektur ist in Abb. 2.6 neben der Internetarchitektur dargestellt. In den nachfolgenden Abschnitten werden kurz die einzelnen Bestandteile der WAP-Architektur vorgestellt.

Wireless Application Environment (WAE)

WAE ist eine Anwendungsumgebung für konvergierende Internet- und Mobilfunkdienste und besteht aus einem eigenen Browser⁵¹, der die folgenden Formate unterstützt:

- **Wireless Markup Language (WML):** WML ist eine von XML (Kap. 3.3.5) abgeleitete Programmiersprache, die für den Einsatz in mobilen Handgeräten optimiert wurde.
- **WML Script:** Analog zu JavaScript bei Webseiten wird bei WAP-Seiten WML Script zur Eingabesteuerung, Vorbearbeitung von Daten und als Zugang zu Telefonfunktionen eingesetzt.
- **Wireless Telephony Application (WTA):** Die Telefonfunktionen, auf die zugegriffen werden kann, werden in WTA definiert.
- **Content Format:** Verschiedene Datenformate für Abbildungen, Telefonbucheinträge und Kalendereinträge werden unterstützt.

⁵⁰WAP Binary XML

⁵¹Software, die die Navigation durch verschiedene Dokumente ermöglicht.

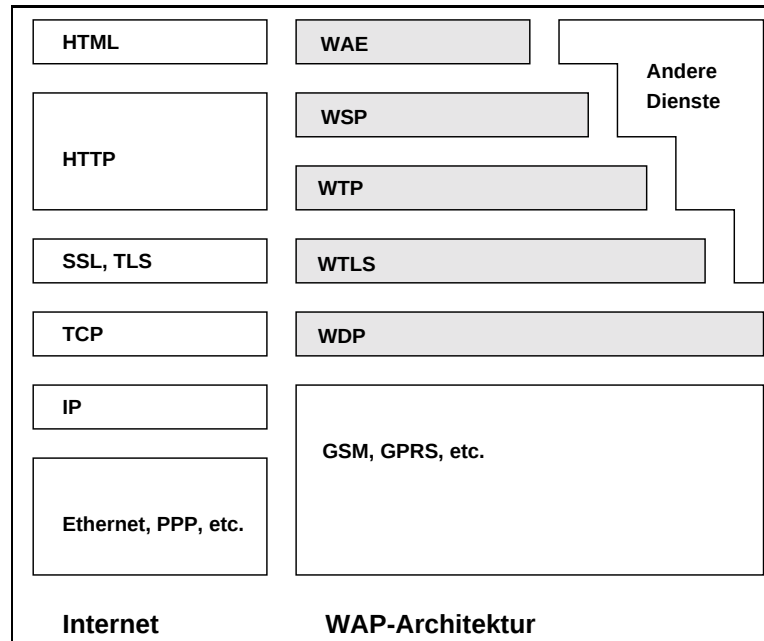


Abbildung 2.6: Internet- und WAP-Architektur

Wireless Session Protocol (WSP)

Das WSP [For00b] ist eine binäre Form von HTTP und bietet der Anwendungsschicht eine einheitliche Schnittstelle zur Sitzungsschicht, die sowohl verbindungsorientierte (WTP⁵²), wie auch verbindungslose Dienste (z.B. Datenpakete über WDP⁵³) unterstützt. Beide Dienste können durch das WTLS-Protokoll zusätzlich gesichert werden.

Wireless Transaction Protocol (WTP)

WTP ist ein transaktionsorientiertes Protokoll für mobile Endgeräte, welches drei Transaktionsklassen anbietet [Elm01]:

- **Klasse 0:** Der Sender schickt lediglich einen Aufruf. Er erhält keine Bestätigung und wird auch kein Wiederversenden des Aufrufes durchführen.
- **Klasse 1:** Klasse 1 beantwortet den Aufruf des Senders mit einer Bestätigung. Im Falle des Nichteintreffens der Bestätigung schickt der Sender den Aufruf erneut.
- **Klasse 2:** Klasse 2 unterstützt das Versenden von Ergebnissen als Antwort auf den Aufruf des Senders. Weiterhin wird das Ergebnis durch den Sender bestätigt (*3-way handshake*).

Wireless Transport Layer Security (WTLS)

WTLS ist ein Sicherheitsprotokoll, das auf dem Industriestandard *Transport Layer Security* (TLS) beruht. WTLS ist optimiert für schmalbandige Kommunikationskanäle und bietet die folgenden Sicherheitsmechanismen:

- **Authentifizierung:** WTLS ermöglicht es, die Authentizität eines Endgerätes und eines Servers zu überprüfen.

⁵²In diesem Fall (*applications over transactions*) werden verschiedene WDP-Pakete von WTP initiiert.

⁵³Hier (*applications over datagrams*) werden die WSP-Pakete direkt als Nutzdaten in WDP-Paketen übertragen.

- **Datenintegrität:** WTLS enthält Mechanismen, die sicherstellen, dass die Daten, die zwischen dem Endgerät und dem Server versendet werden, unverändert und nicht beschädigt sind.
- **Privatsphäre:** WTLS unterstützt die Verschlüsselung von Daten, so dass die versendeten Informationen nicht von zwischengeschalteten Einheiten, die den Datenstrom unterbrochen haben mögen, verstanden werden können.
- **Dienstverweigerung:** WTLS ermöglicht die Feststellung und Ablehnung von Daten, die nicht erfolgreich überprüft werden konnten und erschwert damit Angriffe auf Dienste und schützt höhere Protokollschichten.

Wireless Datagram Protocol (WDP)

WDP ist das verbindungslose WAP-Transportprotokoll, das auf die unterliegenden Trägerdienste von GSM (CSD, SMS, GPRS) und CDMA zugreift und den oberen Schichten eine einheitliche Schnittstelle anbietet, die unabhängig von dem benutzten Trägerdienst ist. Wann immer IP⁵⁴ als Routingprotokoll eingesetzt wird, wird UDP⁵⁵ als WDP-Protokoll eingesetzt.

2.3 Paketorientierte GPRS-Kommunikation

Die Trägerdienste des GSM-Mobilfunknetzes, die bisher Zugang zu paketorientierten Netzen ermöglichen, sind leitungs- bzw. kanalorientiert. Dies bedeutet, dass die Verbindung unabhängig davon, ob Daten übertragen werden oder nicht, für die gesamte Dauer belegt bleibt. Dadurch kann ein konstanter Datenfluss und eine geringe Verzögerung der Datenpakete gewährleistet werden. Nichts desto trotz ist diese Übertragungsart insbesondere bei büschelartigem Verkehr wenig effizient für den Netzbetreiber, der Lizenzgebühren für die knappen Mobilfunkfrequenzen bezahlt, und auch für die Teilnehmer, deren Kommunikationskosten sich in der Regel nach der Verbindungsdauer und nicht nach dem Datenvolumen berechnen.

Dadurch, dass man die leistungsorientierte Verbindung lediglich während des Datentransfers aufrecht erhält und in Phasen ohne Verwendung abbaut, könnte das Problem zeitnah gelöst werden. Aufgrund der Tatsache, dass der Verbindungsaufbau bedingt durch den hohen Signalisierungsaufwand zu einer signifikanten Verzögerung des Datentransfers führt, ist diese Lösung jedoch zu verwerfen. Es ist offensichtlich, dass die sogenannte zweite Mobilfunkgeneration keine effiziente Datenübertragung, insbesondere für häufige kleine Datenmengen, erlaubt⁵⁶. Daher wurde im Rahmen der Phase 2+ Standardisierung von GSM der *General Packet Radio Service* (GPRS) ins Leben gerufen. GPRS benutzt paketorientierte Mechanismen, um die Datenübertragung und Signalisierung effizient durchzuführen. Für GPRS wurden neue Funkkanäle definiert, deren Zuordnung flexibel ist. Dabei können bis zu acht Zeitschlitze (*time slots*, TS) gleichzeitig von den aktiven Teilnehmern verwendet werden. Abb. 2.7 zeigt die Ressourcennutzung bei einer leitungsorientierten

⁵⁴Internet Protocol, Protokoll der ISO/OSI-Schicht 3 (Netzschicht)

⁵⁵User Datagram Protocol; direkter Zugang zur Datenübertragung über IP. Beim UDP-Protokoll sind keine unnötigen Kontrolldaten vorgesehen, dafür kann auch keine verlustfreie Übertragung garantiert werden. Ebenso kann die Reihenfolge der Datengramme vertauscht werden. UDP wird meist dann verwendet, wenn kleine Datenmengen schnell ausgetauscht werden sollen, ohne dass vorher bekannt ist, wann die Übertragung benötigt wird.

⁵⁶Der Kurznachrichtendienst *SMS* erlaubt zwar den Transport kleiner Datenpakete bis 120 byte, bietet aber keinerlei Laufzeitgarantien.

GSM-Datenverbindung für vier Kanäle und die Ausnutzung von drei *Packet Data Channels* (PDCH) der gleichen vier Teilnehmer beim paketorientierten GPRS-Dienst, wobei der GPRS-spezifische zusätzliche Aufwand für die Paketköpfe vernachlässigt worden ist. Nachfolgend wird das GPRS-System und seine Parameter vorgestellt. In Kap. 6.3.1 werden davon ausgehend vier Verkehrsmodelle entwickelt.

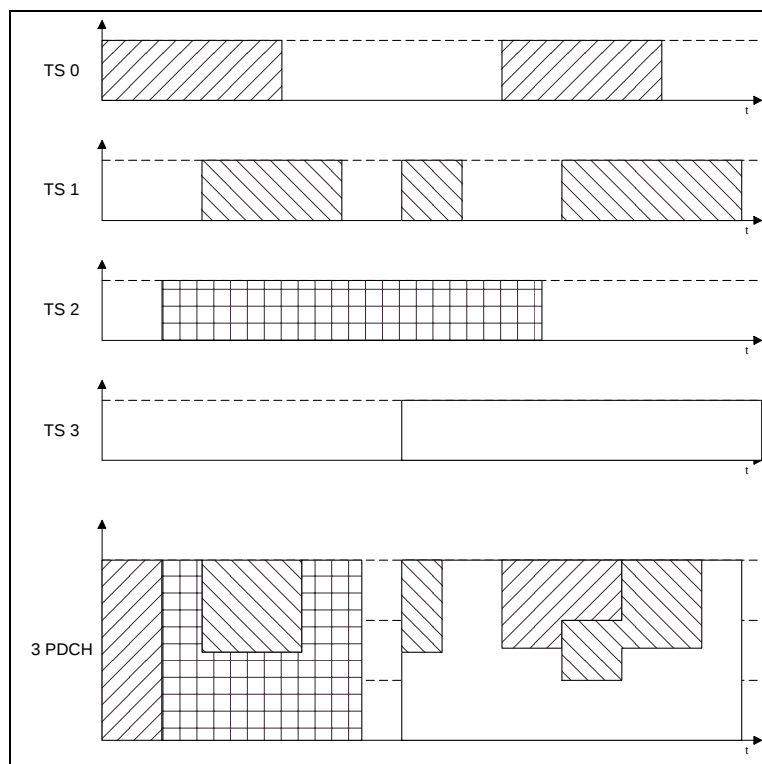


Abbildung 2.7: Ressourcennutzung von GSM CSD und GPRS

2.3.1 GPRS-Mobilstation

Die GPRS-Mobilfunkkanäle unterstützen eine parallele Dienstnutzung, d.h. während einer Sitzung können kanalvermittelte Dienste (Sprache, Daten) und GPRS-Dienste zur paketorientierten Sprach- und Datenübertragung gleichzeitig genutzt werden. Es wird daher zwischen drei Klassen von Endgeräten unterschieden [Wal00]:

- Klasse A: Gleichzeitige Benutzung aller Dienste gemäß Dienstprofil bei gleichbleibender Sprachqualität.
- Klasse B: Eingeschränkte gleichzeitige Benutzung von Diensten mit vermindertem Durchsatz bei geringerer Sprachqualität.
- Klasse C: Keine gleichzeitige Benutzung von Diensten.

Prinzipiell kann sich das GPRS-Mobilfunktelefon in drei verschiedenen Modi befinden: *Frei*, *Standby* und *Aktiv*. Das GSM-Zustandsmodell kennt lediglich die Modi *Frei* und *Aktiv*. Aufgrund des zusätzlichen Modus *Standby* ist der GPRS-Teilnehmer jederzeit in der Lage digitale Pakete zu empfangen und zu senden (*always online*). Zum Empfangen und Senden

von Daten muss das Endgerät allerdings in den aktiven Zustand wechseln. Im Standby-Modus kennt das Kommunikationssystem den Aufenthaltsort des Teilnehmers nur mit der Genauigkeit des Routinggebietes (*routing area*). Das Routinggebiet wird vom Netzbetreiber festgelegt und kann aus einer oder mehreren Zellen bestehen. Im Standby-Modus wird das Netz nicht von jedem Zellwechsel in Kenntnis gesetzt, sondern lediglich von einem Wechsel des Routinggebietes. Der Netzbetreiber kann durch eine geeignete Wahl des Routinggebietes die Anzahl von Gebietswechsellnachrichten steuern. Bevor der Teilnehmer Datenpakete empfangen kann, muss vom Netz ein Rundruf (*paging*) ausgesendet werden. Aufgrund dieses Rundrufmechanismus ist es nicht notwendig, dass das Mobiltelefon im Standby alle Datenpakete kontrollieren muss. Dadurch wird der Stromverbrauch des Telefons signifikant reduziert [Eve99]. Wenn die Mobilstation ein Paket senden möchte, benötigt sie einen Funkkanal. Da sich die GPRS-Benutzer die Funkkanäle teilen, werden die Kanäle vom BSS dynamisch zugewiesen.

2.3.2 Logische GPRS-Architektur

GPRS optimiert den Einsatz der Netz- und Funkkapazitäten. Dabei wird die strikte Trennung zwischen Netz- und Funkkapazitäten des GSM-Standards aufrechterhalten, und die Festnetzkomponenten der GSM-Netze können zusammen mit den neuen Funkübertragungsmethoden weiterverwendet werden, d.h. GPRS erfordert lediglich geringe Modifikationen an den mobilen Vermittlungsstellen (MSCs) im Hinblick auf die Signalisierung. Die GPRS-Architektur, die in Abb. 2.8 skizziert ist, ist der klassischen, leitungsorientierten GSM-Architektur (Abb. 2.2) daher sehr ähnlich.

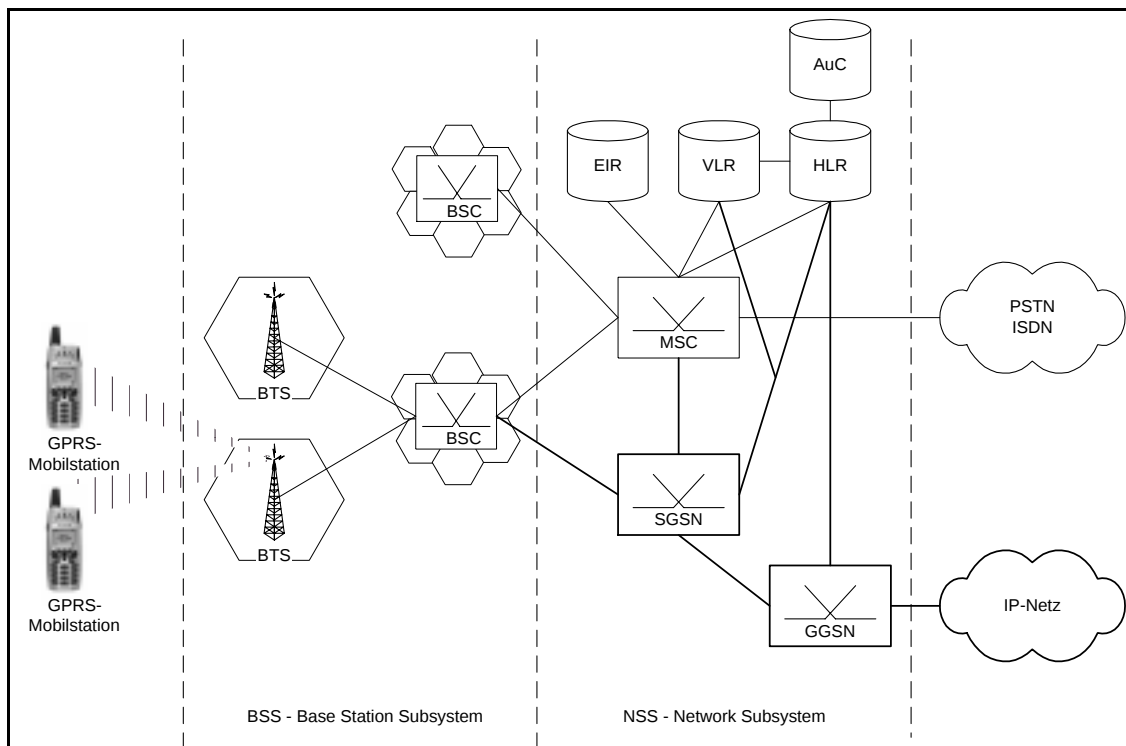


Abbildung 2.8: Logische GPRS-Architektur

Im wesentlichen wird die GSM-Architektur um zwei Netzknoten erweitert, die für die digitale Paketvermittlung verantwortlich sind:

Serving GPRS Support Node (SGSN): Der SGSN ist auf der gleichen Ebene wie die MSCs im NSS angesiedelt. Er ist für die Kommunikation zwischen dem mobilen Endgerät und dem GPRS-Netz verantwortlich, d.h. der SGSN vermittelt die Datenpakete zum Endgerät via BSC und BTS. Ferner ist der SGSN für das Sitzungs- und Mobilitätsmanagement, sowie die Verschlüsselung und Authentifizierung zuständig. Ein SGSN kann mehrere Routinggebiete versorgen und mit mehreren mobilen Vermittlungsstellen verbunden sein. Die Dimensionierung kann in Abhängigkeit von der Verkehrslast vorgenommen werden.

Gateway GPRS Support Node (GGSN): Der GGSN unterstützt die Zusammenarbeit mit externen IP- oder X.25-basierten Netzen und mit den GPRS-Netzen anderer Netzbetreiber. Er ist der erste Kontaktpunkt bei einer Kommunikation mit einem externen paketorientierten Kommunikationsnetz. Der GGSN leitet die ankommenden Pakete (PDEs) zum entsprechenden SGSN des mobilen Teilnehmers weiter. Aus der Sicht eines externen Netzes arbeitet der GGSN als Router⁵⁷ für die IP-Adressen aller GPRS-Teilnehmer. Für die Kommunikation innerhalb des Mobilfunknetzes wird ein IP-basiertes Intranet verwendet. Außerdem kann der GGSN Informationen aus dem HLR auslesen. Das HLR wird dazu um den Eintrag *GPRS Register* erweitert, der Daten über die GPRS-Teilnehmer enthält.

Die Gebührenerfassung für GPRS-Dienste erfolgt über den SGSN und den GGSN. Physikalisch können der SGSN und der GGSN jeweils eigene Netzknoten darstellen, oder als eine Einheit realisiert sein.

2.3.3 Protokollstapel

Der GPRS-Protokollstapel ist in Abb. 2.9 visualisiert [ETS98c]. Drei Protokolle steuern die Verbindung zwischen dem Endgerät und dem SGSN:

- das *Subnetwork Dependent Convergence Protocol* (SNDCP),
- das *Logical Link Control* (LLC) Protokoll,
- und das *Radio Link Control* (RLC) bzw. *Medium Access Control* (MAC) Protokoll

Als Vermittlungsschichtprotokoll in der Mobilstation (MS) bzw. dem GGSN wird in Abhängigkeit von der Festnetzumgebung auf existierende Standardprotokolle (IP, X.25) zurückgegriffen [Wal00]. Zwischen der Mobilstation und dem SGSN wird in der untersten Teilschicht der GSM-Vermittlungsschicht das SNDCP-Protokoll eingesetzt. SNDCP multiplext Pakete von verschiedenen Protokollen auf eine Verbindung, komprimiert die Daten und Protokollköpfe (*protocol header*), und zerlegt Pakete, die größer als die zulässige LLC-Paketlänge sind.

Das LLC-Protokoll baut eine logische Verbindung zwischen dem mobilen Endgerät und dem SGSN auf. LLC kann im unbestätigten Modus, d.h. ohne Rücksicht auf Paketverluste oder im bestätigten Modus, d.h. mit Verwendung von Sendungswiederholungen und Flusskontrolle zur korrekten Auslieferung der Daten, arbeiten. Das LLC-Protokoll ist an das LAPD_m-Protokoll⁵⁸ angelehnt. LLC-Pakete sind nicht größer als 1.520 byte und werden an die RLC-Schicht weitergereicht, wo sie in kleinere RLC-Blöcke zerlegt werden. Die

⁵⁷ISO/OSI-Schicht 1-3 Protokollumsetzerfunktion

⁵⁸ISO/OSI Schicht 1-Protokoll, das zur Datenübertragung zwischen den mobilen Endgeräten und den Basisstationen eingesetzt wird.

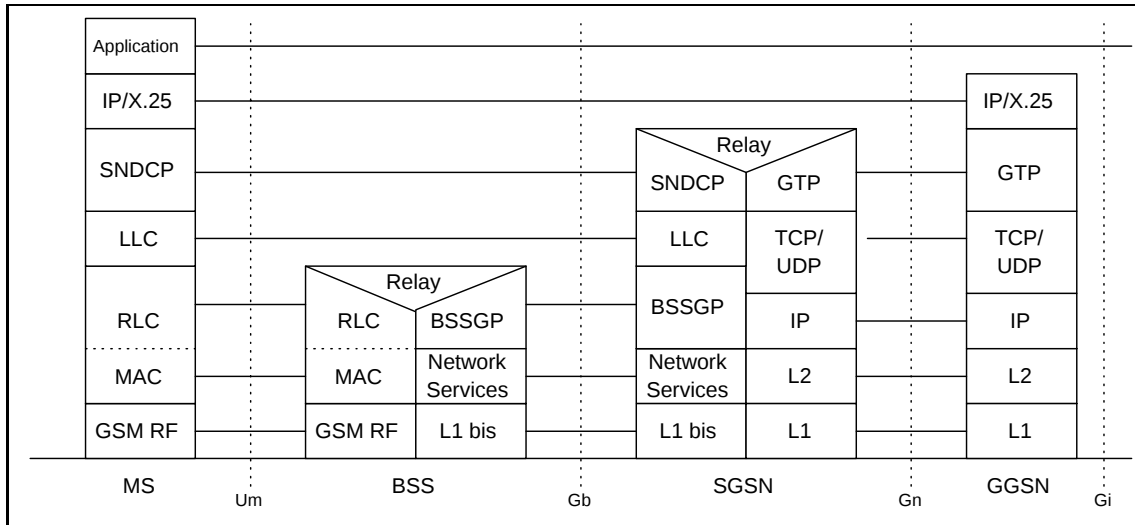


Abbildung 2.9: GPRS-Protokollstapel

Größe der Blöcke ist abhängig vom Kodierungsschema.

RLC ist ebenfalls in der Lage, im bestätigten und unbestätigten Modus zu arbeiten. Das RLC/MAC-Protokoll steuert generell den Zugriff der Paketdatenteilnehmer auf das Funkmedium durch Algorithmen zur Kollisionsauflösung, Multiplexen sowie Kanalreservierung unter Berücksichtigung der angeforderten Dienstgüte [Wal00]. Die RLC/MAC-Schicht ermöglicht einen flexibleren Zugang zu den Funkressourcen. Diese Flexibilität wird durch Einführung von logischen Paketdatenverkehrskanälen (PDTCH⁵⁹) erreicht, die auf einen Paketdatenkanal (PDCH) gemultiplext werden können. Der PDCH ist ein physikalischer Kanal, der einem Zeitschlitz im GSM TDMA-Rahmen entspricht. Bis zu acht PDTCHs können sich einen PDCH teilen. Dadurch wird erreicht, dass ein mobiles Endgerät mehrere physikalische Kanäle gleichzeitig nutzen kann, und dass ein physikalischer Kanal gleichzeitig von mehreren Mobilstationen belegt sein kann.

2.3.4 Datenraten

Die physikalische Schicht (ISO/OSI-Schicht 1) ist an die Bedürfnisse des GPRS-Standards angepasst worden. Vier verschiedene Kodierungsschemata wurden definiert, die sich insbesondere in ihrem Fehlerkorrekturverhalten und ihrem Durchsatz unterscheiden, wie Tab. 2.3 zeigt.

Kodierungsschema	Nutzdaten	Paritätsbits	Rumpfbits	Kodierbits	Punkt. Bits	Kodier-rate	Datenrate [kbit/s]
CS-1	181	40	4	456	0	1/2	9,05
CS-2	268	16	4	588	132	~ 2/3	13,4
CS-3	312	16	4	676	220	~ 3/4	15,6
CS-4	428	16	-	456	-	1	21,4

Tabelle 2.3: Kanalkodierungsschemata in GPRS

⁵⁹Packet Data Traffic Channel

Die Datenraten bei GPRS können daher zwischen 9 kbit/s und bis zu über 160 kbit/s bei der gleichzeitigen Benutzung von acht Kanälen liegen. GPRS ist derart konzipiert, dass periodische und büschelartige Datenverkehrsaufkommen genauso gut übertragen werden wie gelegentliche, kleine Datenpakete. Ferner erlaubt GPRS eine schnelle Reservierung zur Datenübertragung, so dass Paketübertragungen binnen einer halben bis einer Sekunde begonnen werden können.

2.3.5 Dienstgüteparameter

Im Unterschied zu leitungsorientierten Systemen richten sich die Nutzungsgebühren für GPRS-Teilnehmer typischerweise nicht ausschließlich nach der Dauer der Verbindung, sondern auch nach dem Volumen der übertragenen Daten. So ist der GPRS-Benutzer in der Lage, den ganzen Tag lang Daten zu senden und zu empfangen, ohne dass Telefongebühren für 24 Stunden anfallen. Weiterhin unterstützt GPRS verschiedene Dienstgüten, die es ermöglichen, den Datenpaketen der Benutzer unterschiedliche Prioritäten zuzuordnen. Die Dienstgüte (*quality of service*) hängt von den folgenden Eigenschaften ab:

1. Priorität (*precedence class*)
2. Verzögerung (*delay class*)
3. Zuverlässigkeit (*reliability class*)
4. Maximaler Durchsatz (*peak throughput class*)
5. Durchschnittlicher Durchsatz (*mean throughput class*)
6. Funkpriorität (*radio priority*)

Der Benutzer kann zwischen vier verschiedenen Dienstklassen wählen, die den Funkprioritäten entsprechen. Die gewünschte Klasse wird im HLR eingetragen. Die Funkpriorität ist daher die einzige feste Dienstgütegröße. Die restlichen Dienstgüten handeln die Netzknoten auf der Basis der Funkpriorität und der Netzlast aus. Die Umsetzung der Funkpriorität auf die anderen Dienstgüteparameter ist nicht Gegenstand der GPRS-Spezifikationen.

(1) Priorität

Unter normalen Lastbedingungen wird das GPRS-Mobilfunknetz versuchen, die Anforderungen aller Benutzer zu erfüllen. Die Prioritätenklasse legt fest, wie Datenpakete im Fall einer Überlast behandelt werden sollen. Die definierten Prioritätenklassen sind in Tab. 2.4 aufgetragen [ETS98c].

Klasse	Prioritätenname	Beschreibung
1	Hohe Priorität	Dienstzusagen sollen eingehalten werden Vorrang vor den Klassen 2 und 3
2	Normale Priorität	Dienstzusagen sollen eingehalten werden Vorrang vor der Klasse 3
3	Niedrige Priorität	Dienstzusagen sollen eingehalten werden keine Bevorzugung

Tabelle 2.4: Prioritätenklassen

(2) Verzögerung

Die in einem GPRS-Netz aufgrund der physikalischen Übertragung und der Einschränkungen des Systems auftretenden Verzögerungen sollen für bestimmte Verzögerungsklassen minimiert werden. Die Verzögerungsparameter legen die Ende-zu-Ende Verzögerung fest. Dies schließt die Verzögerung für den Zugang zum Funkkanal beim Uplink, die Verzögerung für die Zuteilung eines Funkkanals beim Downlink, die Funkausbreitung und die Verzögerung im Vermittlungsnetz ein. Die Verzögerungen in externen Netzen werden nicht berücksichtigt. In [ETS98a] wurden vier Verzögerungsklassen definiert, die in Tab. 2.5 gegenübergestellt sind.

	Maximale Verzögerung			
	Paketgröße: 128 <i>byte</i>		Paketgröße: 1024 <i>byte</i>	
	Mittl. Verz. [s]	95 % [s]	Mittl. Verz. [s]	95 % [s]
1 (vorhersagbar)	0,5	1,5	2	7
2 (vorhersagbar)	5	25	15	75
3 (vorhersagbar)	50	250	75	375
4 (bestmöglichst)	unbestimmt			

Tabelle 2.5: Verzögerungsklassen

(3) Zuverlässigkeit

Die Zuverlässigkeitsparameter legen die Übertragungscharakteristika fest, die die Anwendung benötigt. Es werden Wahrscheinlichkeiten für die folgenden Ereignisse angegeben [ETS98a]:

- Verlust eines Datenpaketes
- Übertragung eines Datenpaketes in der falschen Reihenfolge
- Doppelte Übertragung eines Datenpaketes
- Korruption eines Datenpaketes

(4) Maximaler Durchsatz

Der maximale Durchsatz spezifiziert die höchstmögliche Datenrate für eine GPRS-Datenübertragung. Der Teilnehmer erhält allerdings keine Garantie, dass der maximale Durchsatz immer erreicht wird. Vielmehr hängt die Erfüllbarkeit von den Fähigkeiten der Mobilstation und der Verfügbarkeit der Funkressourcen ab. Das Kommunikationsnetz kann den Teilnehmer unabhängig von den verfügbaren Übertragungskapazitäten auf eine vereinbarte Durchsatzrate beschränken. Der maximale Durchsatz hängt nicht von der Verzögerungsklasse ab. Es soll möglich sein, den maximalen Durchsatz auch während einer Sitzung neu zu verhandeln. Die definierten Klassen für den maximalen Durchsatz sind in Tab. 2.6 dargestellt [ETS98c].

(5) Durchschnittlicher Durchsatz

Der durchschnittliche Durchsatz wird in byte pro Stunde (*byte/h*) gemessen. Er gibt die Übertragungsrate an, die beim Datentransport durch ein GPRS-Mobilfunknetz erwartet werden kann. Das Netz kann Teilnehmer unabhängig von den verfügbaren Übertragungskapazitäten auf einen vereinbarten durchschnittlichen Durchsatz beschränken. Bei der Klasse

Klasse	Maximaler Durchsatz
1	bis 1 <i>kbyte/s</i> (8 <i>kbit/s</i>)
2	bis 2 <i>kbyte/s</i> (16 <i>kbit/s</i>)
3	bis 4 <i>kbyte/s</i> (32 <i>kbit/s</i>)
4	bis 8 <i>kbyte/s</i> (64 <i>kbit/s</i>)
5	bis 16 <i>kbyte/s</i> (128 <i>kbit/s</i>)
6	bis 32 <i>kbyte/s</i> (256 <i>kbit/s</i>)
7	bis 64 <i>kbyte/s</i> (512 <i>kbit/s</i>)
8	bis 128 <i>kbyte/s</i> (1.024 <i>kbit/s</i>)
9	bis 256 <i>kbyte/s</i> (2.048 <i>kbit/s</i>)

Tabelle 2.6: Maximaler Durchsatz

1 (*best effort*) erhält der Benutzer den Durchsatz, der zur Verfügung gestellt werden kann und benötigt wird. In diesem Fall hat der Teilnehmer keine Garantie für eine bestimmte Durchsatzrate. Das Kommunikationsnetz hingegen kann die Ressourcen effizient ausnutzen. Es soll möglich sein den durchschnittlichen Durchsatz auch während einer Sitzung neu zu verhandeln. Die definierten durchschnittlichen Durchsatzklassen sind in Tab. 2.7 angegeben [ETS98c].

Klasse	Durchschnittlicher Durchsatz
1	bestmöglichst
2	100 <i>byte/h</i> (0,22 <i>bit/s</i>)
3	200 <i>byte/h</i> (0,44 <i>bit/s</i>)
4	500 <i>byte/h</i> (1,11 <i>bit/s</i>)
5	1 <i>kbyte/h</i> (2,2 <i>bit/s</i>)
6	2 <i>kbyte/h</i> (4,4 <i>bit/s</i>)
10	50 <i>kbyte/h</i> (111 <i>bit/s</i>)
12	200 <i>kbyte/h</i> (440 <i>bit/s</i>)
15	2.000 <i>kbyte/h</i> (4,4 <i>kbit/s</i>)
19	50.000 <i>kbyte/h</i> (111 <i>kbit/s</i>)

Tabelle 2.7: Durchschnittlicher Durchsatz

(6) Funkprioritäten

Funkressourcen sind von Natur aus nur begrenzt vorhanden. Beim GPRS-Datentransport durch ein GSM-Netz wird der Flaschenhals daher die Funkschnittstelle sein. Deswegen ist es sinnvoll, Dienstgüteparameter für die Luftschnittstelle zu definieren. Bei ETSI hat man sich auf vier Funkprioritäten geeinigt. Diese Prioritäten legen die Genauigkeit und die Geschwindigkeit der Datenverarbeitung für den Up- und Downlink fest. Das bedeutet, dass die Datenübertragung abhängig von der Funkpriorität ist, die ein Benutzer hat. Bei einer hohen Funkpriorität wird automatisch auch die Dienstgüte für die anderen Netzknoten hoch sein, d.h. eine enge Verknüpfung zwischen den Funkprioritäten und den anderen Dienstgüteparametern ist erwünscht. Es macht wenig Sinn, wenn dem Benutzer eine hohe Funkpriorität eingeräumt wird, die Datenpakete vom restlichen Netz aber mit geringer Priorität übertragen werden.

2.4 Zusammenfassung

In diesem Kapitel wurde zunächst der Dienstbegriff eingeführt und die Arbeit an personalisierten E-Commerce-Diensten motiviert. Hiernach wurden mit GSM, CAMEL, WAP und GPRS die zentralen Konzepte heutiger Mobilfunksysteme erläutert. Mittels des IN-Standards wurden nicht nur Grundlagen für den CAMEL-Standard diskutiert, sondern auch dargestellt (Tab. 2.1), wie langwierig und schwierig die Einführung von zusätzlichen, intelligenten Netzknoten in Telekommunikationsnetzen ist. Das VHE-Konzept dient dabei als Paradebeispiel für die komplizierte Umsetzung einer anerkannt guten Idee. Die Standards für WAP und GPRS wurden besonders intensiv behandelt, da sie später bei der Analyse und Bewertung des neuen, agentenbasierten Kommunikationskonzeptes eine wichtige Rolle spielen.

Die Agententechnologie

Das Fachgebiet der Agententechnologie hat sich in den letzten Jahren zu einem sehr spannenden und dynamischen Forschungsgebiet innerhalb der Informationstechnologien entwickelt und verleiht der Dienstleistung in Telekommunikationsnetzen neue Impulse. In diesem Kapitel werden nach einem kurzen historischen Abriss die Charakteristika von mobilen Agenten und existierenden Systemarchitekturen vorgestellt. Am Ende des Kapitels folgt eine Behandlung der wichtigsten Standardisierungsbemühungen im Bereich der Agententechnologie.

3.1 Historie

Historisch gesehen erwuchs die Agententechnologie in den 70er Jahren als spezielles Forschungsthema aus der Verteilten Künstlichen Intelligenz¹ (VKI) [Son98]. Bereits 1977 hatte MIT²-Professor Carl Hewitt in einem wissenschaftlichen Bericht ein Modell eingeführt [Hew77], in dem er ein System aus verschiedenen Akteuren beschrieb. In diesem System ist jeder Akteur ein Softwareobjekt, das eine Mailadresse und ein ihm eigenes Verhalten hat. Die Akteure agieren parallel und kommunizieren miteinander über Nachrichten. Dieses Modell führte später zum Multiagentensystem (MAS). Das Forschungsthema MAS beschäftigt sich insbesondere mit der Kommunikation, Koordination und Kooperation zwischen Agenten.

Generell lässt sich die Entwicklung der Agententechnologie in zwei Phasen einteilen [Bra97]. In der ersten Phase konzentrierte man sich seit 1977 auf die Makroaspekte eines MAS, d.h. die Kooperation und Kommunikation der Agenten (Zerlegung und Verteilung der Aufgaben, Protokolle etc.), sowie die Koordination (Vermeidung von Konflikten und Redundanz, Synchronisation von Teilaufgaben etc.) standen im Mittelpunkt der Untersuchungen. Die rapide Entwicklung und Verbreitung von öffentlichen Computernetzen (Internet), individuellen Rechnern (PCs), Handcomputern und der Telekommunikation bringt seit Mitte der 90er Jahre neue Bewegung in dieses Forschungsgebiet und leitete somit die zweite Phase ein [KG99]. Primäres Ziel dieser Phase ist es, ein effektives MAS zu entwerfen. Dabei wurden eine Reihe neuer Anwendungsfelder offenkundig, und die Untersuchung der Agententypologie rückte mehr in den Vordergrund [WJ95b, WJ95a]. Die Agententechnologie setzt sich nun nicht nur mit den Erkenntnissen aus der Forschungsthematik der Künstlichen Intelligenz (KI) auseinander, sondern entwickelt vielmehr ihre Lösungen auch aus den wissenschaftlichen Arbeitsgebieten Objektorientierung (OO) und Verteilte Systeme (VS). In dieser zweiten Phase etablierte sich ferner auch die Differenzierung zwischen stationären (intelligenten) Agenten und mobilen Agenten, welche in Kap. 3.2.2 detaillierter beschrieben wird.

Den wissenschaftlichen Durchbruch erlangten die mobilen Agenten mit Hilfe der Agentenprogrammiersprache *Telescript* und der Agentenplattform *MagiCap*, die beide 1994 von

¹Distributed Artificial Intelligence (DAI)

²Massachusetts Institute of Technology, Cambridge, USA

General Magic herausgebracht wurden [Whi94]. Telescript ist in C++ geschrieben, rechnerunabhängig und erlaubt den Agenten sich selbständig von einer Ausführungsumgebung zu einer anderen Ausführungsumgebung zu bewegen. Dafür muss die Agentenumgebung in der Grundkonfiguration so eingerichtet sein, dass auf jedem Rechner der Agentenumgebung ein Interpreter läuft. Nach nur wenigen Jahren hat man bei General Magic die Arbeiten an Telescript wieder eingestellt und neue, Java-basierte Agentenprodukte (Odyssey, Kenya etc. [Gen00]) entwickelt.

Insbesondere die Einführung der Programmiersprache Java [Sun95] des Unternehmens Sun Microsystems hat der Entwicklung der Agententechnologie zu einem weiteren enormen Auftrieb verholfen. Dieser führte dazu, dass in den vergangenen Jahren eine Reihe von Agentenplattformen auf der Basis von Java entwickelt worden sind [HGF98, Con98]. Mit Hilfe von Java gelang es vor allem die existierenden Probleme im Hinblick auf die Übertragung, Plattformunabhängigkeit und Sicherheit von Software effizienter zu lösen.

Neben den verschiedenen Agentenplattformen, die seit 1994 konzipiert und implementiert wurden, sind seit dieser Zeit auch eine Vielzahl von Definitionen [CHK95, FG96, GHN⁺97, Bra97, FIP97, Lan98] entstanden, die beschreiben, wann ein Programm Agent genannt werden darf. Die in der vorliegenden Arbeit verwendeten Definitionen werden im folgenden Abschnitt vorgestellt.

3.2 Was ist ein Agent? - Begriffsdefinitionen

Wenngleich Softwareagenten noch lange nicht den uralten Traum der Heinzelmännchen, die einem jeden Wunsch von den Lippen ablesen können, erfüllen, so gibt es doch heute eine Reihe von nützlichen Softwareagenten, die dem Benutzer beim Umgang mit digitalen Informationen hilfreich zur Seite stehen (z.B. Internetsuchmaschinen).

Zur gleichen Zeit haben sich Agenten zum Schlagwort in den verschiedensten technischen Bereichen, z.B. Benutzerschnittstellen, Programmiersprachen, Verteilte Systeme und Robotik, entwickelt. Der Brockhaus [Bro66] kennt für einen Agenten die folgende Definition:

Agent [lat. *Handelnder*],
ein im Auftrag eines anderen in dessen Interesse Tätiger

Für Softwareagenten gilt daher in dieser Arbeit:

Softwareagenten sind ausführbare Programme, die bestimmte vordefinierte Aufgaben intelligent und autonom für einen Benutzer bzw. eine Anwendung durchführen können.

Einige Internet-Server nutzen Softwareagenten, um ein Benutzerprofil zu erstellen, das die Vorlieben und Abneigungen des Besuchers beinhaltet. So können beim Besuch einer Webseite personalisierte Dienste (Kap. 2.1) angeboten werden. Mittels des Benutzerprofils wählen Softwareagenten dann Artikel, Dienstleistungen und Werbung aus, an denen der Benutzer Interesse haben könnte. Das Wissen bzw. der Datenspeicher dieser Agenten wird theoretisch mit der Zeit immer größer. Das große Datenvolumen verringert jedoch die Mobilität der Agenten und erschwert deren Einsatz in Kommunikationsnetzen mit Funkschnittstellen. Der Mobilitätsaspekt wird in Kap. 3.2.2 näher untersucht.

3.2.1 Autonomie

Die gängigste und wichtigste Eigenschaft eines Softwareagenten ist seine Autonomie. Für diese Eigenschaft benötigen Agenten vor allem detaillierte Informationen über ihren Aufenthaltsort und ihre Umgebung, um die übertragenen Aufgaben innerhalb dieser Umgebung ausführen zu können. Während der Ausführung kann es außerdem notwendig sein, dass der Agent auf wechselnde Rahmenbedingungen bzw. konkrete Anfragen reagieren muss. Dabei sollte der Agent in der Lage sein, eine Entscheidung ohne vorherige Rücksprache mit dem Benutzer bzw. der Anwendung zu treffen. Die Kompetenzen eines Softwareagenten müssen daher im Vorfeld klar definiert und abgegrenzt werden. Der Benutzer muss dabei entscheiden, welche Rechte er den von ihm beauftragten Agenten übertragen möchte.

Analog zu der Problematik, die sich bei der Betrachtung der Interaktionsmöglichkeiten ergibt, ist festzustellen, dass Agenten hinsichtlich ihrer Bewegungsfreiheit bisher auf spezielle Plattformen beschränkt sind. Daher sind die Handlungsmöglichkeiten und die Komplexität von Agenten stets nur für diesen, im Sinne des Gesamtsystems, lokalen Gültigkeitsbereich ausgelegt. Agenten führen nach ihrer Aktivierung in der Regel eine Reihe von Handlungen aus, die zur Erfüllung von Aufgaben innerhalb ihres Gültigkeitsbereiches erforderlich sind. Dazu gehören z.B. der Bezug von Systeminformationen von zentralen Registern wie einem *Naming Service*³ oder die Interaktion mit anderen, stationären Agenten wie z.B. einem Dienstleistungsagenten (*provider agent*). Ist beispielsweise ein Agent für eine Plattform entwickelt worden, die die Existenz von Dienstleistungsagenten vorsieht, dann kann ein sinnvoller Einsatz desselben Agenten in einem Szenario, das die gleichen Dienste über einen Naming Service realisiert, nicht bzw. nur schwer erfolgen. Das Verhalten von Softwareagenten ist folglich stets nur für einen exakt zu begrenzenden Raum intelligent und autonom zu nennen.

3.2.2 Mobilität

In [MPZ96, FH00] werden zwei Kategorien von Softwareagenten definiert, siehe Abb. 3.1⁴:

Softwareagenten werden anhand ihrer Mobilität in **Intelligente Agenten** und **Mobile Agenten** unterteilt, wobei die intelligenten Agenten im Hinblick auf ihre beschränkte Mobilität häufig auch als **Stationäre Agenten** bezeichnet werden.

Die intelligenten Agenten haben ihre Wurzeln in der Künstlichen Intelligenz (KI) und sind durch ihre Urteilkraft, Planungs- und Lernfähigkeit charakterisiert⁵. Als Intelligenzstufen wird in Abb. 3.1 neben rein ausführender Software (*executable*) zwischen folgerndem (*reactive*), planendem (*proactive*) und lernendem (*adaptive*) Programmcode unterschieden. Die interessantesten wissenschaftlichen Erkenntnisse aus dem Bereich der intelligenten Agenten beschäftigen sich mit Agentensprachen wie KQML, ACL und KIF (eine Beschreibung dieser Sprachen erfolgt in Kap. 3.3.4), d.h. mit effizienten und lernenden Sprachmodellen für die Kommunikation zwischen Softwareagenten.

³Der Naming Service ist ein Dienst, der Objekten eine Kennung zuordnet und Objekte anhand dieser referenziert.

⁴RPC steht für Remote Procedure Call und meint entfernte Funktionsaufrufe. Eine genauere Betrachtung erfolgt in Kap. 3.3.

⁵In der Fachliteratur werden intelligente Agenten daher auch gerne als BDI (beliefs, desires, intentions) Agenten [Mül96] bezeichnet.

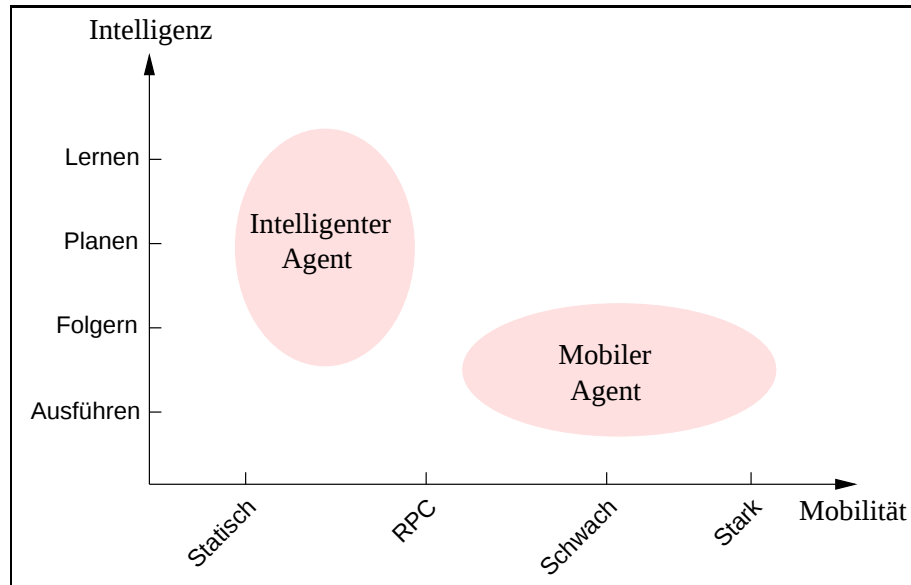


Abbildung 3.1: Klassifikation von Softwareagenten

Mobile Agenten werden in dieser Arbeit wie folgt definiert:

Mobile Agenten sind Programminstanzen, die sich eigenständig im Kommunikationsnetz bewegen und dort im Auftrag ihres Besitzers oder einer Anwendung verschiedene Dienste verrichten.

Mobile Agenten (MAs) sind typischerweise in einer Skriptsprache programmiert und können während der Ausführung von Netzknoten zu Netzknoten im Kommunikationsnetz migrieren. Als Programminstanzen bestehen sie aus Programmcode, Daten und Zustandsinformationen [Hoh01]. In der Fachliteratur [MPZ96, BHRS97] wird die Agentenmobilität häufig anhand der übertragenen Information (Daten, Code, Laufzeitinformation) in *Funktionsaufrufe* (RPC), *schwache Agentenmigration* (weak migration) und *starke Agentenmigration* (strong migration) unterteilt. Auf den besuchten Knoten des Kommunikationsnetzes können MAs mit stationären Agenten und anderen Ressourcen zur Erledigung der Aufgaben zusammenarbeiten. Der Agent entscheidet dabei, aufgrund von neu gewonnenen Informationen selbst, ob eine weitere Migration zur Erledigung der Aufgabe notwendig ist und wohin er im Netz migriert. Daher benötigt er keine dauerhafte Verbindung zu seinem Benutzer. Abb. 3.2 zeigt einen typischen Migrationsprozess eines MAs, bei dem der Agent nach vier Migrationen (1-4) wieder zu dem Ausgangsknoten, der sogenannten Heimat, zurückkehrt. Während eine Zeitlang in wissenschaftlichen Publikationen immer neue graphische Darstellungen für MAs verwendet wurden, hat sich in den letzten Jahren die Visualisierung durch eine Spielfigur etabliert, die auch konsequent in dieser Arbeit verwendet wird.

MAs haben eine Reihe von Vorteilen für verteilte Anwendungen. Durch die Migration zur Informationsquelle kann der Agent seine Anfragen asynchron, d.h. ohne regelmäßigen Datenaustausch zwischen Sender und Empfänger, lokal durchführen. Eine agentenbasierte Kommunikation zwischen Netzknoten, die lediglich eine unzuverlässige Netzverbindung besitzen, wird beim Abbruch der Verbindung nach erfolgter Agentenmigration nicht gestört, da der Agent weiter lokal agieren kann und erst nach wiederhergestellter Verbindung zurückgeschickt wird. Damit eignen sich MAs insbesondere für den Einsatz in mobilen

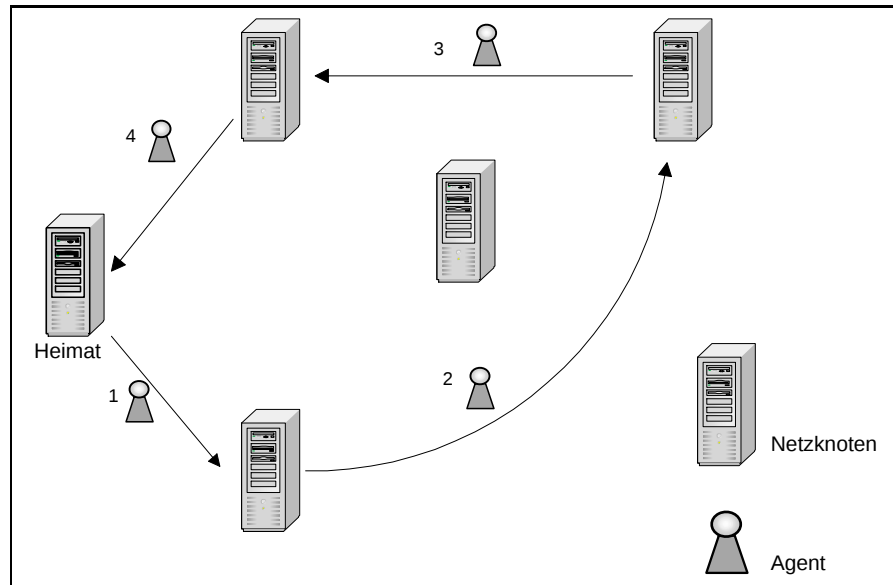


Abbildung 3.2: Mobile Agenten im Kommunikationsnetz

Endgeräten, denn oft müssen sich diese Geräte auf teure oder instabile Netzverbindungen verlassen. Das bedeutet, dass Aufgaben, die eine ständige Verbindung zwischen einem mobilen Endgerät und einem Festnetz erfordern, wirtschaftlich und/oder technisch häufig nicht durchführbar sind. Gelingt es aber die Aufgaben in MAs einzubetten, dann können die mobilen Agenten nach dem Entsenden ins Kommunikationsnetz asynchron und autonom operieren. Das mobile Endgerät kann die Verbindung später wiederherstellen und die Agenten einsammeln.

Weiterhin können die Aufgaben zwischen involvierten Netzknoten entsprechend der momentanen Auslastungen der Netzknoten dynamisch verteilt (*dynamic load balancing*) werden [Kre96]. Eine feste Zuordnung wie bei typischen Client/Server-Anwendungen existiert nicht. Außerdem können verschiedene Migrationsstrategien in Abhängigkeit von der zu erfüllenden Aufgabe und der momentanen Knoten- und Netzauslastung zum Tragen kommen. Die Summe dieser Möglichkeiten macht den Einsatz von MAs besonders interessant für Mobilfunkdienste, wo man häufig mit unzuverlässigen Verbindungen, geringem Durchsatz, leistungsschwachen Endgeräten und hohen Wartezeiten rechnen muss.

Doch die genannten Charakteristika sind zweifellos nicht revolutionär und ihre Gültigkeit ist nicht auf MAs beschränkt. Spezielle Anwendungen lassen sich auch mit traditioneller Technik effizient und robust realisieren. Die wahre Stärke von MAs liegt nicht allein darin, dass neue, verteilte Anwendungen möglich werden, sondern vielmehr in dem neuartigen Dienstleistungs-konzept, das einen Rahmen bietet, um verteilte Anwendungen einfach und effizient zu entwerfen, zu modellieren und zu implementieren.

3.2.3 Systemterminologie

Eine Agentenumgebung (*agent environment*) besteht aus Agentensystemen. Abb. 3.3 (a) stellt exemplarisch eine Agentenwelt mit zwei Agentensystemen dar.

Ein **Agentensystem** (*agent system*) ist eine Plattform, die es erlaubt, Softwareagenten zu entwerfen, zu testen, auszuführen, zu übertragen und zu beenden.

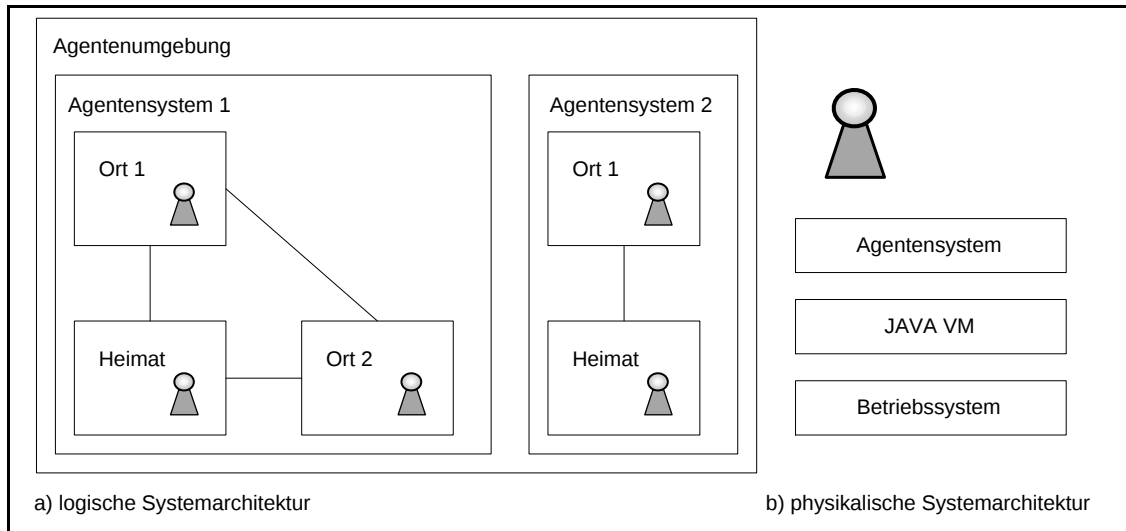


Abbildung 3.3: Systemarchitektur

Jedes Agentensystem benötigt daher einen Zugang zu einer Kommunikationsinfrastruktur, um MAs übertragen zu können. Das Agentensystem begrenzt und regelt die Fähigkeiten eines Agenten. Es wird im Allgemeinen physikalisch auf mehrere Rechner verteilt installiert sein. Die *Agentensystemtechnologie* bezeichnet den Namen und die Version der eingesetzten Systemtechnik (Agentenplattform). Moderne Agentensysteme laufen auf einer Java Virtual Machine (JVM), die ihrerseits ein Betriebssystem benötigt, das Java unterstützt, siehe Abb. 3.3 (b). Ein Agentensystem kann aus einem oder mehreren Orten bestehen.

Ein **Ort** (*place*) ist eine Laufzeitumgebung (*runtime environment*) des Agentensystems mit eigener Adresse, welche Agenten Dienste zur Verfügung stellt.

Beispiele für Orte sind die *Heimat* (Startpunkt, an dem ein Agent erzeugt wird), der *Registrierungsor*t (Ort, an dem alle Agenten, die einen speziellen Dienst anbieten, registriert sind) und der *Zielort* (Ort, zu dem der Agent migrieren muss, um seine Aufgabe zu erfüllen). Die Adresse eines Ortes besteht aus dem Namen des Ortes und der Adresse des Agentensystems. Es wird zwischen *permanente Orten*, die immer auf einem dedizierten Netzknoten vorhanden sind, und *temporären Orten*, die nur zeitweise verfügbar sind oder zwischen Netzknoten wandern, unterschieden.

Mehrere Orte, welche nicht zwangsläufig demselben Agentensystem angehören müssen, können zu einer **Region** zusammengefasst werden.

Um MAs eine autonome Bewegung im Netz zu ermöglichen, sind neben der Einrichtung von Orten, die Agenten eines bestimmten Profils beheimaten können, weitere Maßnahmen erforderlich. Dazu gehören ein Dienstregister und eine Verwaltung, die Agenten und ihre Aufenthaltsorte erfasst. Entsprechende Konzepte zur zentralen Registrierung und Verwaltung, sowie von zentralen Informationsregistern findet man sowohl in den Standards der FIPA zum Agentenmanagement [FIP98a], als auch im MASIF⁶-Entwurf der OMG [OMG98].

⁶MASIF (Mobile Agent System Interoperability Facility) wird in dieser Arbeit zur Realisierung einer Systemkommunikation zwischen verschiedenen Agentensystem eingesetzt und im Kap. 4 ausführlich diskutiert.

Die Mobilität der Agenten wird durch die Reichweite der Region bzw. der Kommunikationsinfrastruktur begrenzt. Prinzipiell ist es nicht möglich, dass Agenten eines Typs A zu einem Ort eines Agentensystems des Typs B migrieren und dort ihre Aufgaben ausführen. Die Ursachen hierfür liegen einerseits in der nicht einheitlichen Benennung von Agenten und ihren Aufenthaltsorten, die einen solchen Transport von Agenten selbst bei kompatiblen Systemarchitekturen ausschließen, wenn die verwendeten Programmiersprachen in Typ und Version übereinstimmen. Andererseits kann jedoch auch bei kompatiblen Agentensystemen der Austausch von Informationen und Agenten aufgrund der Verwendung von unterschiedlichen Java-Versionen (d.h. verschiedene JDK⁷s) blockiert werden. Eine genauere Betrachtung dieser Problematik erfolgt in Kap. 4.

3.2.4 Agententypen

In den vorangegangenen Abschnitten wurden Definitionen zu Agenten und Agentensystemen vorgenommen. Für das Arbeiten mit Agenten ist es zudem sinnvoll, Agenten nicht nur aufgrund ihrer Eigenschaften, sondern auch im Hinblick auf ihre Funktion zu kategorisieren. Gemäß Abb. 3.4 werden in der vorliegenden Arbeit Softwareagenten anhand ihrer Funktion in **Managementagenten** und **Dienstagenten** unterteilt.

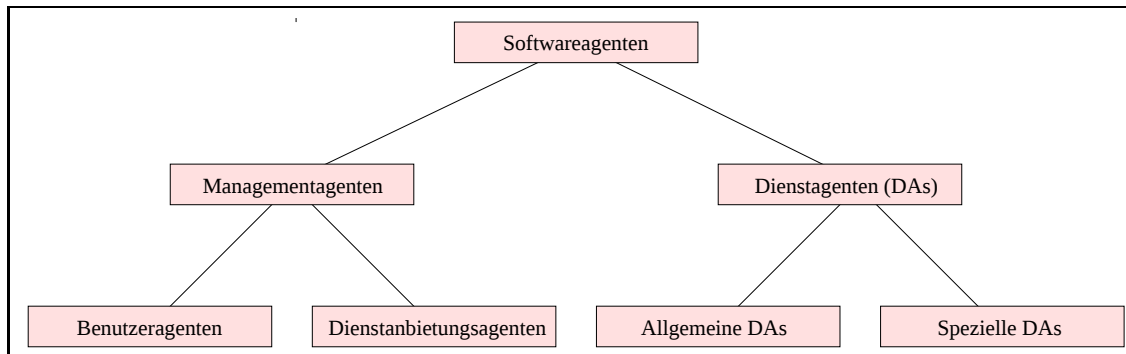


Abbildung 3.4: Agententypen

Managementagenten: Typische Vertreter von Managementagenten (management agents) sind *Dienstleistungsagenten* (provider agents) und *Benutzeragenten* (user agents). Managementagenten sind in der Regel ortsabhängig. Jeder Ort bietet zwar eine Ausführungsumgebung für Agenten, stellt aber in der Regel keinen Zugang zu lokalen Ressourcen zur Verfügung. Dies ist die Aufgabe der Dienstleistungsagenten, die permanent an einem Ort verfügbar sind [Lem99]. Dabei ist es durchaus möglich, dass an einem Ort mehrere Dienstleistungsagenten existieren. Benutzeragenten kennen die Dienste und dedizierten Vorlieben eines bestimmten Teilnehmers und speichern diese in einem Benutzerprofil ab. Managementagenten können sowohl mobil als auch stationär sein. In der Regel ist aber ihr Wissen für das Agentensystem wichtiger als ihre Mobilität.

Dienstagenten: Dienstagenten (*service agents*) sind für die Erbringung eines bestimmten Dienstes beim Benutzer verantwortlich und tragen daher insbe-

⁷Java Development Kit

sondere die vom Benutzer festgelegten Regeln und Informationen mit sich. Ein Dienstagent wird entweder im Ort lokal gestartet oder migriert zu einem Ort, um diesen zu besuchen. Dienstagenten werden nach ihrer Funktion in zwei Klassen unterteilt:

Allgemeine Dienstagenten: Allgemeine Dienstagenten⁸ (ADA) werden eingesetzt, um die Logik eines Dienstes auszuführen und mit speziellen Dienstagenten zu kommunizieren. ADAs werden immer dann zum Endgerät transferiert, wenn eine neue Version des Dienstes vorliegt⁹ oder wenn der Dienst bisher nicht auf dem Endgerät verfügbar ist. Die bekanntesten Vertreter der ADAs sind die *Endgeräteagenten* (terminal agents), welche für die Ein- und Ausgabeschnittstelle an einem bestimmten Endgerät verantwortlich sind [CAM99].

Spezielle Dienstagenten: Spezielle Dienstagenten¹⁰ (SDA) sind Agenten die einen bestimmten Dienst anbieten und ausführen. In dieser Arbeit werden hauptsächlich zwei SDAs für den mobilen Handel, *Bankagenten* (banking agents) und *Suchagenten* (search agents), eingesetzt und untersucht. Bankagenten werden z.B. dafür benötigt, um Bankaufträge eines Teilnehmers auszuführen. Suchagenten können für einen Teilnehmer gesuchte Informationen im digitalen Informationsangebot finden.

3.2.5 Lebenszyklus

Unter Lebenszyklus versteht man die Ausführungszustände eines MAs [GHN⁺97]. Wie in Abbildung 3.5 dargestellt ist, beginnt der MA im Zustand *Start*, bei dem die benötigten Ressourcen zugeteilt und der Agent als Prozess gestartet wird. Danach befindet sich der MA zunächst im *aktiven* Zustand. Beendet wird die Existenz eines MAs im Zustand *Ende*, wobei die Ressourcen wieder freigegeben und der Prozess beendet wird.

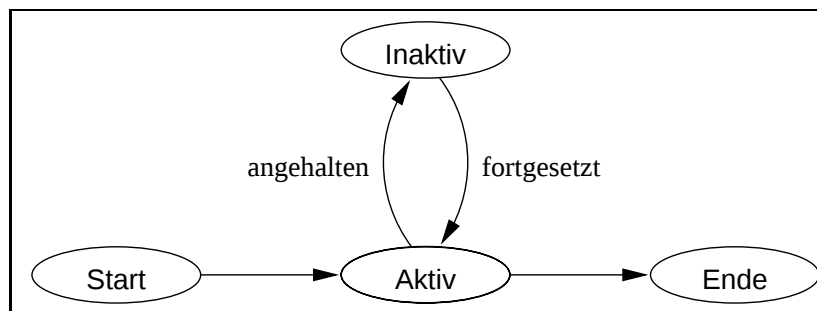


Abbildung 3.5: Lebenszyklus eines mobilen Agenten

Vor einer Migration werden die Laufzeitzustände (Programmcode und Laufzeitdaten) in ein Datenformat umgewandelt und der Agentenprozess wird angehalten. Der MA befindet sich dann im *inaktiven* Zustand. Die Daten werden anschließend zum Zielort übertragen. Dort werden die Daten wieder in ein ausführbares Format gebracht. Dem MA wird ein neuer Prozess zugeteilt und das Agentenprogramm wird fortgesetzt. Läuft alles einwandfrei,

⁸common service agents

⁹hierzu ist eine vorherige Registrierung des Dienstes bzw. Benutzer notwendig

¹⁰specific service agents

wird der Agent in der Heimat, also der Laufzeitumgebung, bei der der MA seine Migration begonnen hat, zerstört und der Prozess wird freigegeben.

3.2.6 Sicherheitsaspekte

Sicherheitsüberlegungen spielen bei der agentenbasierten Kommunikationskonzepten eine wichtige Rolle. Das Kernprinzip bei Sicherheitsüberlegungen im IT-Bereich ist die Isolation von sensitiven Daten/Codes durch Hardware- und/oder Softwarebarrieren und hierarchische Zugriffskontrolle auf diese Daten/Codes nach bestimmten Privilegkriterien [CH97]. Das Hauptziel von sicherer Kommunikation ist die Echtheit von Inhalt und Ursprung der Nachrichten zu garantieren und den Inhalt der Nachrichten vor dem Zugriff durch Unbefugte zu schützen. In der Praxis werden die Sicherheitskonzepte in folgenden fünf Funktionen umgesetzt [FB97]:

- **Authentifizierung** (Identitätssicherung)
Die Authentifizierung dient dazu, die Echtheit (Authentizität) von Person, Programm und Daten zu überprüfen und zu bestätigen.
- **Zugriffsberechtigung**
Zugriffskontrolle schützt System- und Netzressourcen vor unbefugtem Zugriff.
- **Vertraulichkeit**
Eine solche Funktion sorgt dafür, dass Informationen lediglich an Personen oder Organisationen weitergegeben werden, denen man vertraut.
- **Datenintegrität**
Die Datenintegrität sorgt dafür, dass Daten nicht geändert und gelöscht werden.
- **Verbindlichkeit**
Durch Bestätigungen, d.h. durch zusätzlichen Nachrichtenaustausch, können Daten vor der Zurückweisung oder Verweigerung durch einen Transaktionspartner geschützt werden.

Eine Agentenumgebung besteht aus Agenten und Agentensystemen (Rechner oder Netzknoten), die Agentenfunktionen (Migration, Kommunikation und Dienste) ermöglichen. Die Agentenumgebung lässt sich in mehrere Bereiche unterteilen, in denen die Sicherheitsaspekte eine unterschiedliche Rolle spielen. In Abb. 3.6 werden verschiedene Bereiche im Bezug auf Sicherheitsaspekte visualisiert, die in den folgenden Abschnitten diskutiert werden.

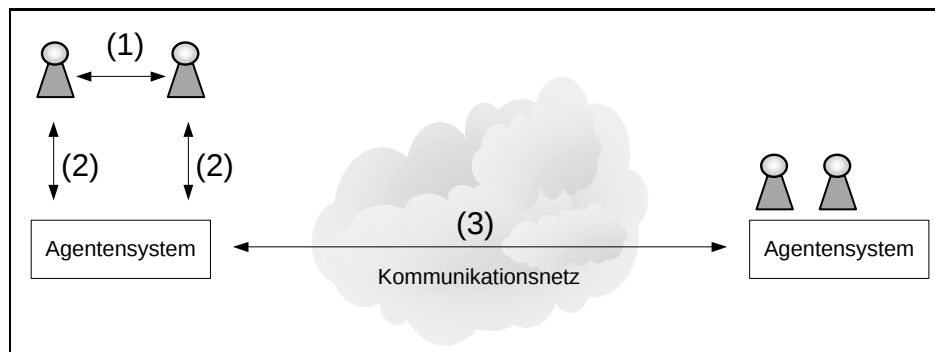


Abbildung 3.6: Sicherheitsaspekte bei Agentensystemen

(1) Sicherheit zwischen Agenten

Die Sicherheit zwischen Agenten bedeutet den Schutz eines Agenten vor anderen (böswilligen) Agenten. Mögliche Attacks sind Manipulation an Code und Daten, Maskierung (durch z.B. Fälschung der Identität), Betrug (z.B. Benutzung von Diensten ohne Bezahlung) und Dienstverweigerung¹¹ (absichtliche Verhinderung von legalen Diensten z.B. durch Erzeugen einer künstlichen Überlast). Diese Arten von Attacks gehören zu den klassischen Problemen eines verteilten Systems und werden in einer Reihe von wissenschaftlichen Berichten diskutiert [FB97]. Mögliche Gegenmaßnahmen sind:

- **die Benutzung einer sicheren Programmiersprache**
Bei der Programmiersprache Java ist z.B. ein Zugriff auf den Adressraum eines anderen Programmes nicht möglich. Dadurch wird eine physikalische Manipulation an Code und Daten ausgeschlossen.
- **die Authentifizierung von Agenten**
Das Errechnen eines genügend langen Schlüssels ist selbst mit den leistungsfähigsten Rechnern nahezu unmöglich. Durch Verwendung von öffentlichen und privaten Schlüsseln und digitalen Signaturen kann man die Authentizität eines Agenten verifizieren.
- **die Verschlüsselung der Kommunikation/Agentenübertragung**
Die Kommunikation selbst kann kodiert werden, so dass ein Fremder aus den empfangenen Daten keinen Nutzen ziehen kann. Häufig verwendete Technologien sind digitale Zertifikate (X.509), digitale Signaturen (MAC), symmetrisches (DES) und asymmetrisches (RSA) Verschlüsselungsverfahren [FB97]. Es existieren eine Reihe von sicheren Kommunikationsprotokollen, wie z.B. Secure Socket Layer (SSL) und Secure HTTP (S-HTTP), die auf diesen Technologien basieren.

(2) Sicherheit zwischen Agentensystemen und Agenten

Aufgrund der Tatsache, dass die Beziehung zwischen Agenten und Agentensystemen nicht symmetrisch ist, werden die Sicherheitsüberlegungen in zwei Abschnitte unterteilt. Während der erste Abschnitt dem Schutz des Agentensystems vor Agenten gewidmet ist, wird im zweiten Abschnitt der Schutz von Agenten vor Agentensystemen untersucht.

(2a) Schutz des Agentensystems vor böswilligen Agenten

Dieses Sicherheitsproblem ist identisch mit dem Problem des Schutzes eines Programmes vor einem Virus. Ferner greifen alle Schutzmaßnahmen aus dem Abschnitt *Sicherheit zwischen Agenten*. Wenn ein böswilliger Agent auf einer Agentenplattform ausgeführt (in meisten Fällen interpretiert) wird, hat das Agentensystem sowohl volle Kontrolle über seine eigene Laufzeitumgebung als auch über die Agentenlaufzeitumgebung. Daher sind folgende, weitere Gegenmaßnahmen möglich:

- Überprüfung von Programmcode und Laufzeitdaten
- Ressourcenkontrolle und Laufzeitrestriktionen

(2b) Schutz von Agenten vor böswilligen Agentensystemen

Dieser Bereich stellt die größten Herausforderungen in Bezug auf die Sicherheit des Systems dar, denn diesmal ist der Angreifer das System selbst. Der Agent wird in der Umgebung des Agentenservers ausgeführt (in den meisten Fällen interpretiert) und der Agentenrechner

¹¹denial-of-service

kann jedes Bit sowohl vom Programmcode, der Daten als auch die Zustände des Agenten lesen, speichern und manipulieren. Neben den Attacks, die im Abschnitt *Sicherheit zwischen Agenten* beschrieben wurden, muss ein Agent den schlimmsten Angriff *Privacy-Breaking* (Angriff auf die Vertraulichkeit) begegnen. Das angreifende Agentensystem hat genügend Zeit, um den Programmcode des Agenten zu verstehen, das Sicherheitsschild zu durchbrechen und wichtige Informationen (z.B. elektronisches Geld und private Schlüssel) dem Agenten zu entlocken.

Bis heute gibt es in diesem Bereich keine überzeugenden Lösungen. In [FGS96] wird sogar behauptet, dass es gar nicht möglich ist, einen Agenten gegenüber einem Agentensystem zu schützen. Ausgehend davon entstehen drei passive Lösungen. Die erste ist, eine Migration des Agenten zu einem nicht vertrauenswürdigen Agentensystem zu verhindern. Eine zweite Lösung ist, dass man eine Agentenwelt konzipiert, in dem nur authentifizierte Agentenserver existieren. Die dritte Lösung ist, dass man in jedem Agentensystem eine *angriffsempfindliche* Hardware einbaut und dadurch unbefugte Zugriffe verhindert. Alle diese Lösungen führen zu einem geschlossenen System und die Anwendungsmöglichkeiten der Agententechnologie werden dadurch deutlich begrenzt. Der Schwachpunkt in diesem Bereich liegt im Programmcode des Agenten, weil er der Schlüssel zu vertraulichen Informationen und in diesem Fall lesbar und manipulierbar ist. Die Frage ist, wie man verhindern kann, dass der Programmcode zum Zugang zu vertraulichen Informationen wird. In [ST98a, ST98b] wird eine neue Idee vorgestellt, in der man sowohl verschlüsselte Daten als auch verschlüsselten Programmcode für Agenten verwendet. Der Programmcode kann so in einer verschlüsselten Form ausgeführt werden und die Algorithmen (z.B. eine Signierfunktion) sind daher dem Agentensystem unbekannt und der *Privacy-Breaking* Angriff kann verhindert werden.

Eine interessante Idee wird in [Hoh01] beschrieben. Hier wird ein Agent als eine zeitbegrenzte Blackbox beschrieben, deren Inhalt (Programmcode, Daten) dem Agentenserver unbekannt ist. Diese Blackbox wird durch Verschlüsselung der Daten und *absichtliche Unordnung* des Programmcodes erreicht. Unordnung bedeutet, dass der Programmcode durch einen bestimmten Algorithmus in eine für Menschen sehr schwer zu lesende Form gebracht wird. Es ist aber wahrscheinlich, dass Hacker den Programmcode des Agenten früher oder später verstehen und manipulieren können. Deshalb ist die Blackbox zeitbegrenzt und nach einer vordefinierten Zeit nicht mehr gültig. Dies ist sicherlich eine interessante Idee, doch sie birgt weiterhin ein hohes Sicherheitsrisiko.

(3) Sicherheit zwischen Agentensystemen

Die Sicherheitsaspekte bei der Kommunikation zwischen verschiedenen Agentensystemen sind nahezu identisch zu den Überlegungen zur Sicherheit zwischen Agenten. Die möglichen Angriffe bestehen ebenfalls aus Maskierung, Betrug und Dienstverweigerung.

3.3 Mechanismen für die agentenbasierte Kommunikation

In diesem Abschnitt werden die für Softwareagenten verwendeten Kommunikationsmechanismen, Kommunikationssprachen und Programmiersprachen vorgestellt. Kommunikationspartner in Verteilten Systemen (VS) tauschen Informationen in Form von Nachrichten, sogenannten *Protokolldateneinheiten* (PDEs) [Wal00], aus. Der Nachrichtenaustausch (*message passing*) ist sowohl synchron (direkt, z.B. über einen Datenkanal) als auch asynchron (zeitversetzt, z.B. via E-Mail) möglich. Das Konzept ist generell sehr flexibel, aber auch recht komplex und damit fehleranfällig. Abb. 3.7 zeigt gängige Methoden zum Nach-

richtenaustausch. Das meistverwendete Konzept zum Nachrichtenaustausch zwischen verschiedenen Netzknoten in einem Kommunikationsnetz ist dabei der sogenannte *Remote Procedure Call* (RPC).

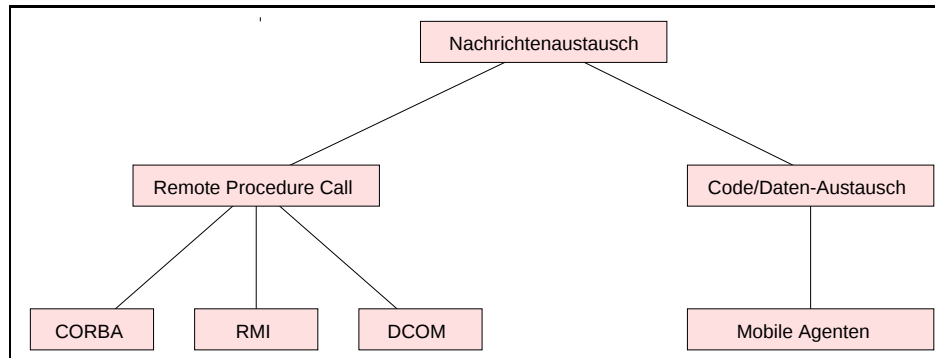


Abbildung 3.7: Kommunikationsmechanismen verteilter Systeme

Grundidee des RPCs ist die Ausweitung des Prinzips lokaler Funktionsaufrufe in verteilten Umgebungen, d.h. für einen entfernten Funktionsaufruf wird eine Nachricht konstruiert und verschickt, die Prozesse der aufrufenden Stelle werden angehalten, die Funktion wird auf dem entfernten Rechner ausgeführt, die Ergebnisse werden in einer Nachricht zurückgeschickt und der aufrufende Rechner führt seine Prozesse fort, als ob die Funktion lokal abgearbeitet worden sei.

Aufgrund seiner Ähnlichkeit zum lokalen Funktionsaufruf ist der RPC schnell zum Industriestandard für den Nachrichtenaustausch in verteilten System geworden. Das Konzept hat jedoch ein paar schwerwiegende Nachteile:

- RPCs sind zeitaufwendiger als lokale Aufrufe und können bei umfangreichen Interaktionen zwischen verteilten Rechnern zu einer hohen Kommunikationsverkehrslast erzeugen.
- Es sind keine Referenzübergaben (*call by reference*) möglich, da sich die kommunizierenden Rechner nicht notwendigerweise denselben Adressraum teilen, woraus ein erhöhter Kommunikationsbedarf und verlängerte Antwortzeiten resultieren.
- Während der Prozessausführung auf dem entfernten Rechner verbleibt der aufrufende Rechner im Wartezustand und belegt dabei Systemspeicher und Rechenleistung.
- Aufgrund der Tatsache, dass die Funktionsrümpfe¹² (*stubs*) und die Funktionsskelette¹³ (*skeletons*) zusammenpassen müssen, stellen RPCs lediglich statische Informationskanäle dar.

Bekannteste Beispiele für RPC sind der CORBA¹⁴-Standard der OMG (siehe Kap. 4.2), RMI¹⁵ von Sun Microsystems und COM¹⁶ bzw. DCOM¹⁷ von Microsoft.

¹²Code auf der Clientseite, dessen Hauptaufgabe die Übermittlung der Prozeduraufrufe sowie deren Ergebnisse ist.

¹³Server-seitige stubs

¹⁴Common Object Request Broker Architecture

¹⁵Remote Method Invocation, Java-basierter Kommunikationsmechanismus

¹⁶Common Object Model

¹⁷Distributed Common Object Model

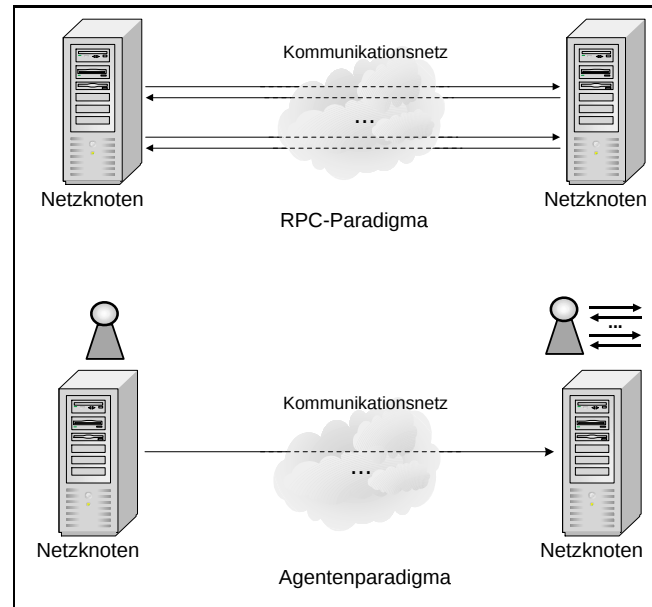


Abbildung 3.8: Vergleich RPC - Mobile Agenten

Anstelle von RPCs können Nachrichten auch mit Hilfe von MAs transportiert werden (vgl. Abb. 3.7). Im Gegensatz zum RPC wird dabei nicht nur das entfernte System gesteuert, sondern es werden vielmehr auch Programmcode, Daten und gegebenenfalls auch der Programmzustand ausgetauscht. Eine Netzverbindung wird, wie in Abb. 3.8 dargestellt, somit nur für die eigentliche Übertragung des MAs benötigt. Dafür muss jedoch auf den involvierten Rechnern ein entsprechendes Agentensystem installiert sein, das in der Lage ist, die MAs zu verarbeiten.

3.3.1 Nachrichtenaustausch

Anhand ihrer Funktionsweise lassen sich Nachrichten generell unterteilen in:

- Synchron-Nachrichten,
- Future-Nachrichten,
- One-Way-Nachrichten und
- Multicast-Nachrichten

Abb. 3.9 zeigt den charakteristischen Informationsfluss dieser Nachrichtentypen. Im Falle von *Synchron-Nachrichten* sendet zunächst Prozess A eine Nachricht an B, und blockiert dann solange, bis er eine Antwort von B erhält. B bearbeitet die Nachricht und sendet eine Antwort an A zurück. Der Prozess A wird so Zeit lange blockiert, bis eine Antwort ankommt. Bei *Future-Nachrichten* findet keine Blockierung des Prozesses A nach dem Senden der Nachricht statt, sondern die Prozesse A und B arbeiten asynchron. Allerdings ist nach wie vor eine Antwort von B notwendig. A wird zu einem späteren Zeitpunkt überprüfen, ob eine Antwort von B angekommen ist. Bei *One-Way-Nachrichten* ist eine Antwort von B nicht erforderlich. *Multicast-Nachrichten* bieten eine Möglichkeit, gruppenorientierte Nachrichten zu übermitteln. Im klassischen Szenario wird ein *Relais*¹⁸, das für die Verteilung solcher Nachrichten zuständig ist, benötigt. Eine Nachricht wird zuerst von A an den Repeater geschickt und dieser leitet die Nachricht weiter an die Ziele (B, C und D).

¹⁸Funktionen der ISO/OSI-Schichten 1-3 (je nach Zielsetzung)

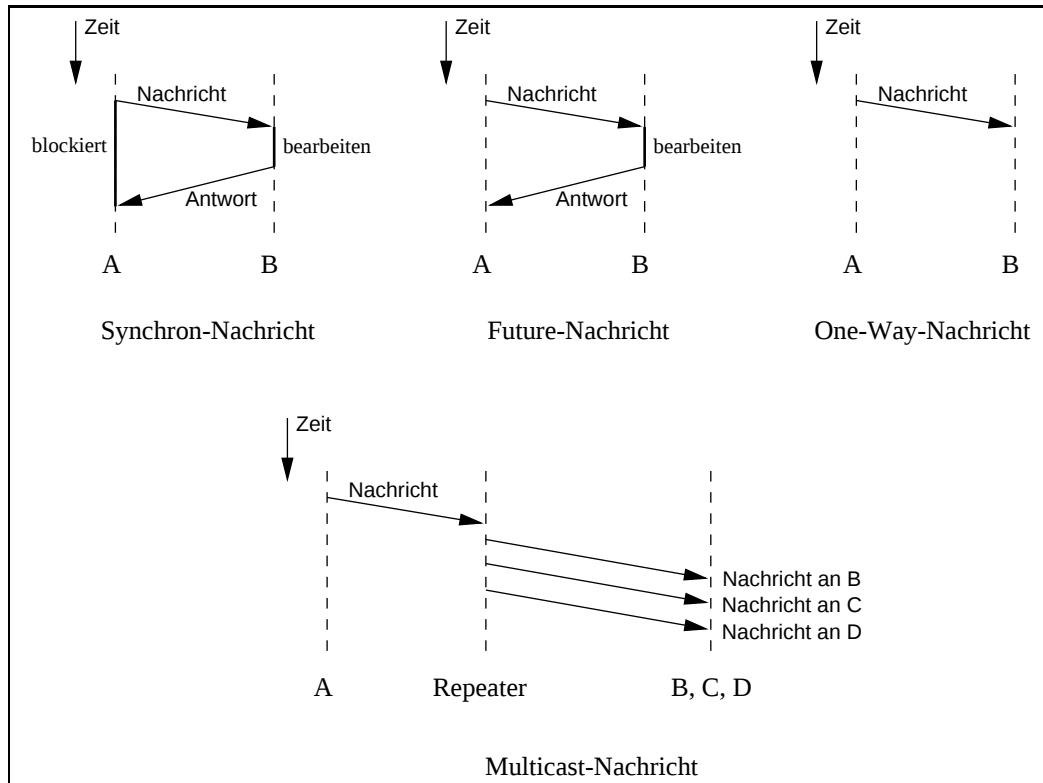


Abbildung 3.9: Kommunikation mit Hilfe von Nachrichten

3.3.2 Eventmechanismus

Benutzer, Anwendungen, Systeme und Agenten können asynchron miteinander auch über Ereignisse (*events*) kommunizieren, die von jedem dieser Partner sowohl erzeugt als auch weiterverarbeitet werden können. Eventmechanismen sind für den Einsatz in VS sehr gut geeignet, weil bei dieser Art der Kommunikation bei keinem der Partner ein Prozess blockiert wird. Zu jedem Event gehört ein Objekt, das Informationen beinhaltet. Um einen Eventmechanismus zu realisieren, benötigt man neben einem Erzeuger und einem oder mehreren Verbrauchern auch eine Instanz, bei der sich einerseits die Verbraucher von Events registrieren müssen, damit sie beim Auftreten eines solchen informiert werden können und wohin andererseits die Objekte von Events vom Erzeuger geschickt werden. Eine solche Instanz könnte

- der Eventerzeuger selber,
- ein spezielles Beobachtungsobjekt (*Event-Listener*)
- oder eine zentrale Stelle (*Event-Manager*)

sein. Während die ersten beiden Möglichkeiten für einzelne Ereignisse gedacht sind, sollte bei komplexeren Zusammenhängen und Systemen ein zentraler Event-Manager zum Einsatz kommen. Abb. 3.10 zeigt die Event-Kommunikation mittels Event-Listener und Event-Manager.

Ein Event-Listener ist dabei ein Prozess, der sich im Gegensatz zum Event-Manager auf dem gleichen Netzknoten wie der Erzeuger befindet und der lediglich für die Weiterleitung der vom Erzeuger generierten Events an die registrierten Prozesse zuständig ist. Ein Event-Manager dagegen kann sich auch auf einem anderen Netzknoten im Kommunikationsnetz

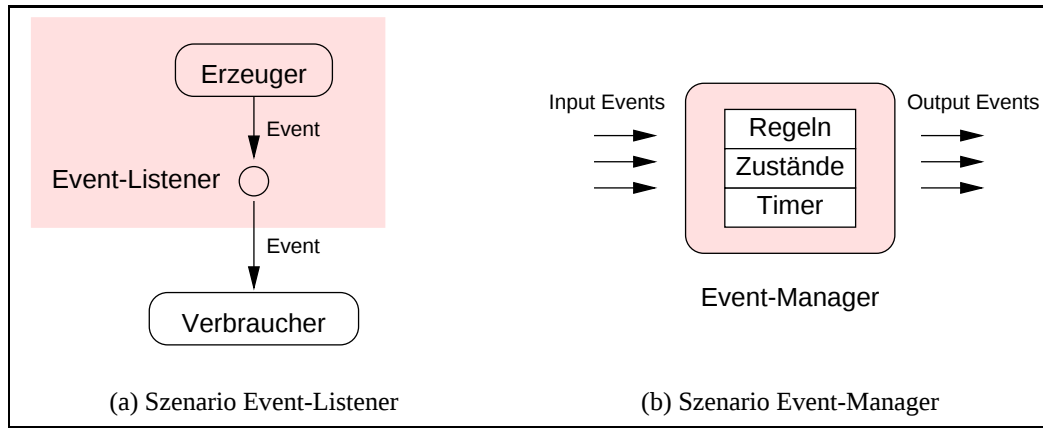


Abbildung 3.10: Kommunikation durch Events

befinden. Er ist zuständig für die Entgegennahme von auf verschiedenen Netzknoten erzeugten Events und deren Weiterleitung an die registrierten Prozesse, die ebenfalls auf anderen Knoten im Kommunikationsnetz laufen können.

3.3.3 Blackboard

In einem Blackboard-System kooperieren Agenten über eine globale Datenstruktur (Blackboard), die Aktionen und Objekte beinhaltet. Ein Blackboard-System besteht neben globalen Datenstrukturen aus Wissensquellen und Steuermechanismen [Bra97]. Agenten können Informationen (*facts*, *rules* und *scripts*) auf dem Blackboard veröffentlichen und lesen. Ein ähnliches System wurde in [LDD95] *Agent Information Space* genannt, in dem Agenten Informationen speichern und lesen können und bei dem jeder Eintrag einen Schlüssel, eine Zugriffskontrolle und einen Wert enthält. Dieses System ermöglicht die Kommunikation zwischen Agenten, Agentensystemen und Agentenbesitzern.

3.3.4 Agentensprachen

Für die Kommunikation zwischen Agenten untereinander, zwischen Agenten und Diensten und zwischen Agenten und Benutzern wurden eigene Sprachen entwickelt, die auf gemeinsamen Regeln, Bedingungen und Wissen fußen. In diesem Abschnitt werden die bekanntesten Agentensprachen, die vor allem bei intelligenten Agenten weit verbreitet sind, jedoch für mobile Agenten eine geringere Bedeutung haben, kurz vorgestellt. Während die *Knowledge Query and Manipulation Language* (KQML) und die *Agent Communication Language* (ACL) Agentenkommunikationssprachen sind, vereinfacht das *Knowledge Interchange Format* (KIF) als Inhaltssprache (*content language*) die Agentenkommunikation. Alternativ zu KIF werden aber bei der Agentenkommunikation auch häufig Inhaltssprachen bzw. einfache Programmiersprachen wie Prolog¹⁹, XML (Kap. 3.3.5), HTML²⁰ und WML (Kap. 2.2.5) eingesetzt.

KQML

KQML [MLF96] ist eine Kommunikationssprache bzw. ein Kommunikationsprotokoll zum Austausch von Informationen und Wissen. Damit sollen Kooperationen und Interopera-

¹⁹Programmiersprache der KI auf der Basis eines logischen Kalküls.

²⁰Hypertext Markup Language

tionen zwischen Agenten ermöglicht bzw. verbessert werden. Die Sprachunterstützung erfolgt durch eine auf dem Sprechakt (*speech acts*) basierende Kommunikationssprache, die eine agentenunabhängige Semantik bereitstellt [MJ00]. KQML konzentriert sich dabei auf eine erweiterbare Menge von Aktionsarten (*performatives*), die ein Agent bei der Kommunikation mit anderen Agenten und deren Wissen verwenden kann. Auf der Basis dieser Menge werden komplexe Agenteninteraktionsarten, wie z.B. Verhandlungen aufgebaut. KQML wird in einer Reihe von Prototypsystemen in den Anwendungsbereichen intelligenter Entwurf und intelligente Planung eingesetzt.

KIF

KIF [GF92] ist eine Inhaltssprache, die für den Austausch von Wissen zwischen ungleichen Softwareprogrammen an der Stanford Universität in den USA entwickelt wurde. Im Gegensatz zu KQML besitzt KIF eine deklarierende Semantik, d.h. die Bedeutung der Ausdrücke ist eineindeutig. Deshalb wird KIF gerne als gemeinsame Sprache bei der Agentenkommunikation, aber auch zwischen verschiedenen Programmiersprachen eingesetzt. Die Grundvokabeln bestehen aus Variablen, Operatoren, Objekten, Funktionen und Beziehungen. Aus diesen Vokabeln können komplexere Ausdrücke konstruiert werden.

ACL

ACL ist ähnlich wie KQML eine universelle, nachrichtenbasierte Kommunikationssprache, die auf der Sprechakttheorie beruht und dabei Informationen derart zusammenpackt, dass andere Agenten den Zweck der Kommunikation erkennen [FIP98b]. Dabei ist es der FIPA (Kap. 3.5.1) gelungen, mit einer Anzahl von nur 20 Aktionsarten (*performatives*) einen Sprachschatz zu definieren, mit dem eine flexible Kommunikation zwischen Softwareeinheiten möglich ist. Sie erlaubt u.a. die dynamische Einführung und Entfernung von Diensten [HGF98]. Nachfolgend wird ein einfaches Beispiel für eine in ACL-formulierte Suche angeführt, wobei der Inhalt der Suche in der Datenbanksprache SQL (Kap. 3.3.5) angegeben wird.

```
(query
:sender buyer
:receiver market
:ontology fipa-shopping
:language SQL
:content
  select price from market where ware='pizza romana'
:reply-with pizza-romana-price)
```

Die FIPA hat analog zu KIF mit der *Semantic Language* ebenfalls eine Inhaltssprache standardisiert [Tel00].

3.3.5 Einsatz von XML, PHP und SQL

Der Einsatz von Agenten im Internet wird durch eine Reihe von modernen Programmiersprachen ermöglicht und vereinfacht. Neben der Programmiersprache Java wurden im Rahmen dieser Arbeit zur Realisierung des flexiblen, agentenbasierten Kommunikationskonzeptes die Programmiersprachen XML, PHP und SQL verwendet, um dynamische, d.h. in Abhängigkeit vom Benutzer, dem Zeitpunkt und dem Ort des Aufrufes sich verändernde, Inhalte anbieten zu können.

XML

XML (*Extensible Markup Language*) [Sea98] ist ein einfaches, flexibles Textformat, welches von der Standard Generalised Markup Language (SGML, ISO 8879), abgeleitet ist und 1998 vom World Wide Web Consortium (W3C) eingeführt wurde. XML dient dazu, die Informationsvielfalt im Internet besser beherrschen zu können. Dabei verfolgte W3C die folgenden Ziele:

- XML soll internationale, medienunabhängige, elektronische Veröffentlichungen ermöglichen.
- XML soll Industrieunternehmen den Entwurf von plattformunabhängigen Protokollen zum Datenaustausch erleichtern.
- XML soll Daten derart verpacken, dass sie direkt weiterverarbeitet werden können.
- XML soll den Transport von Daten über das Internet/Intranet vereinfachen.
- XML soll auch ohne komplizierte, teure Software lesbar sein.
- XML soll die Informationssuche vereinfachen.

Das Resultat der Standardisierung ähnelt HTML, jedoch ist die XML-Struktur noch universeller als HTML. Im Einzelnen:

- XML ist purer ASCII-Text, der mit einem einfachen Texteditor erstellt und gelesen werden kann. Dadurch können XML-Dateien im Gegensatz zu herkömmlichen Datenformaten wie einem IP-Datenpaket jederzeit vom Benutzer nach Informationen durchsucht werden.
- XML ist wohl strukturiert, d.h. alle Kennzeichen (*tags*) werden geöffnet und geschlossen. Nachfolgend ein kurzes Beispiel für eine XML-Datei:

```
...
<Produkt>
  <Preis>15</Preis>
  <W\"ahrung>Euro</W\"ahrung>
  <Farbe>blau</Farbe>
</Produkt>
...
```

- Die XML-Grammatik kann überprüft werden, denn jedes XML-Dokument basiert auf einer *Document Type Definition* (DTD). Eine fehlerhafte oder unvollständige XML-Datei kann daher schnell durch eine Syntaxprüfung ermittelt werden.

XML ist besonders für den Einsatz in Agentensystemen geeignet, da der Agent aufgrund der DTD die Datenstruktur der XML-Datei kennt und so einfach und schnell die gesuchte Information in der Datei finden kann. Bei HTML-Dateien ist dies viel schwerer, da diese in erster Linie dafür konzipiert sind, Informationen zu visualisieren, und sich bei jeder Designveränderung auch die Position der gesuchten Daten verschiebt.

Besonders im Bereich von *Business-to-Business* (B2B) Anwendungen wird XML derzeit gerne und häufig eingesetzt, da es sich um eine standardisierte Kommunikationssprache handelt, die sich mit Hilfe einer entsprechenden DTD leicht in den verschiedensten Anwendungsfeldern einsetzen lässt. So werden im Europäischen *ParcelCall*-Projekt [PWKH00, KH01] z.B. die Daten von Transportgütern, wie Absender (*consignor*), Empfänger (*consignee*), Zielort, Herkunft, Größe, Gewicht, aktuelle Position und Temperatur, etc. durch XML-Dateien zwischen den verschiedenen Komponenten des Systems übertragen [DH01].

PHP

Die von einem Server zu einem Client übertragenen Web- und WAP-Seiten können statisch oder dynamisch generiert sein. Statisch sind die Seiten, wenn z.B. lediglich fertige HTML- bzw. WML-Dokumente vom Server geladen werden müssen.

Dynamische Web- und WAP-Seiten werden erst nach der Anfrage des Clients erstellt.

Dynamische Methoden werden häufig bevorzugt, weil diese den Anbietern von Web- und WAP-Inhalten mehr Flexibilität bieten und weniger aufwendig zu warten sind. Anstatt stetig neue HTML- bzw. WML-Seiten zu programmieren, können z.B. aktuelle Informationen mittels MAs in einer Datenbank abgelegt werden und dann per Skript dynamisch in ein entsprechendes HTML- bzw. WML-Dokument eingelesen werden.

In der Vergangenheit wurden zum Erstellen von dynamischen Seiten häufig CGI²¹-Skripte eingesetzt. Wird die Unterstützung durch ein CGI-Skript benötigt, so startet der Server das Skript, das meist in PERL verfasst ist, und schreibt die Ausgabeinformation in die Kommandozeile bzw. in eine Datei, welche dann zum Webbrowser (Client) gesendet wird.

PHP (*PHP Hypertext Preprocessor*²²) ist eine leicht zu lernende Programmiersprache, deren Syntax an C angelehnt ist [PHP01]. PHP kann problemlos auf jedem Apache-Webserver²³ installiert werden. Für andere Webserver wie den engsten Wettbewerber von Apache, dem Microsoft Internet Information Server, gibt es PHP-ähnliche Produkte (z.B. Microsoft ASP). Aufgrund der Tatsache, dass PHP für die verschiedensten Betriebssysteme ohne Gebühr verfügbar ist, hat PHP schnell eine hohe Akzeptanz in der Internetgemeinschaft gefunden. Im Vergleich mit CGI hat PHP erhebliche Vorteile:

- CGI-Skripte erzeugen gewöhnlich eine komplette Web-/WAP-Seite, d.h. auch wenn nur eine Zeile des Dokumentes dynamisch generiert werden muss, wird die komplette Seite durch das Skript erstellt. PHP arbeitet somit effizienter, denn die PHP-Anweisungen stehen direkt im Programmcode des Web-/WAP-Dokumentes und zwar nur an den Stellen, die wirklich dynamisch generiert werden müssen.
- Der PHP-Parser ist im Web-/WAP-Server integriert. Folglich ist der gesamte Vorgang erheblich schneller und leistungsfähiger als das Starten eines PERL-Skriptes.
- PHP ist speziell für die Bearbeitung von Web-/WAP-Seiten geschrieben worden. Viele Sicherheitsaspekte wie Ausführungs- und Zugriffsbeschränkungen werden bereits im Vorfeld zufriedenstellend gelöst.
- PHP arbeitet gut mit existierenden Datenbanken zusammen. So werden rund 20 verschiedene Schnittstellen zu SQL-Datenbanken und auch eine LDAP²⁴-Schnittstelle vom PHP-Parser unterstützt.

Das nachfolgende Beispiel zeigt ein HTML-Dokuments mit integriertem PHP-Skript:

```
<HTML>
<HEAD></HEAD>
Dies ist ein PHP-Beispiel<BR>
<?php echo 'Datum und Uhrzeit:'.date(''); ?>
</HTML>
```

²¹Common Gateway Interface

²²rekursive Abkürzung, ähnlich GNU: GNU is Not Unix

²³mehr als die Hälfte aller Webserver laufen heute mit der Apache-Software [Net00]

²⁴Lightweight Directory Access Protocol

Innerhalb eines Dokuments können mehrere PHP-Blöcke, die mit dem Tag `<?php` eingeleitet werden, integriert sein. Der PHP-Parser des Web-/WAP-Servers ersetzt diese Blöcke jeweils mit den Ergebnissen der aufgerufenen Funktionen. Damit eignet sich PHP hervorragend für die Programmierung von dynamischen Web-/WAP-Seiten und zur Integration von Softwareagenten. Beispielsweise können der Start und die Ergebnisse eines Suchagenten (Kap. 3.2.4) mittels PHP einfach in eine dynamische Web-/WAP-Seite integriert werden.

SQL

SQL (Structured Query Language) ist ein Standardprotokoll zur Kommunikation mit einer *relationalen Datenbank*²⁵, das von IBM entwickelt wurde und heute von marktführenden Datenbankherstellern wie Oracle und Sybase verwendet wird. Generell werden Datenbanken hauptsächlich zum Speichern von Informationen verschiedener Formate, die von ein paar Zahlen bis zu großen Textmengen variieren können, verwendet. Der entscheidende Vorteil einer Datenbank ist, dass innerhalb kürzester Zeit ein gesuchter Eintrag aus einer großen Datenmenge herausgesucht werden kann.

Eine typische SQL-Anfrage sieht wie folgt aus:

```
SELECT Name, Vorname FROM Kunden WHERE Alter < 30
```

Diese Anfrage gibt als Ergebnis alle Einträge in den Feldern *Name* und *Vorname* zurück, wo gleichzeitig das Feld *Alter* mit einem Wert kleiner als 30 belegt ist.

Ein wesentlicher Vorteil von SQL-Datenbanken ist ihre Fähigkeit zur Kommunikation über öffentliche Kommunikationsnetze wie dem Internet. So kann man eine SQL-Datenbank, die mit dem Internet verbunden ist, von jedem Rechner aus abfragen. Gewöhnlich authentifiziert sich der Benutzer gegenüber der Datenbank mit einem Benutzernamen und einem Passwort. Da Softwareagenten bei informationsreichen Suchvorgängen häufig mit einer SQL-Datenbank in Kontakt treten, findet man in zahlreichen Agentenanwendungen eine entsprechende Schnittstelle.

3.4 Agentensysteme

Die rapide Entwicklung der Agententechnologie ist auch ein Resultat der schnellen Verbreitung der Programmiersprache Java. Vor der Einführung von Java gab es lediglich einige wenige Agentenplattformen, von denen nur *Telescript* von General Magic als kommerzielles Produkt interpretiert werden kann. Heute gibt es rund ein Dutzend erfolgreiche Agentenplattformen, die fast ausschließlich Java-Programmcode einsetzen, so dass eine Unterscheidung der Agentenplattformen bezüglich der verwendeten Programmiersprache nicht mehr zeitgemäß ist.

In diesem Kapitel werden zwei relativ ausgereifte Agentenplattformen - *Voyager* von ObjectSpace und *Grasshopper* von IKV++ - vorgestellt, die im Rahmen dieser Arbeit eingesetzt werden. Im Anhang der Arbeit werden zusätzlich die Agentensysteme *Aglets Workbench* von IBM und *Concordia* von Mitsubishi diskutiert. Grundsätzlich konnten bei den vier untersuchten Agentenplattformen (Voyager, Grasshopper, Aglets und Concordia) die folgenden Systemcharakteristika festgestellt werden:

²⁵Eine relationale Datenbank besteht aus verschiedenen, zueinander in Beziehung stehenden bzw. verknüpften Tabellen.

- Agentensysteme unterstützen die Grundfunktionen zum Arbeiten mit MAs, wie Erzeugen, Verstehen, Übertragen, Empfangen, Ausführen und Beenden von Agenten.
- Agentensysteme sind ausschließlich in Java implementiert und benötigen daher eine JVM und ein JDK in der Version 1.1 oder höher.
- Mobile Agenten werden als Java-Bytecode übertragen.

3.4.1 Voyager

Das Produkt *Voyager* der Firma ObjectSpace²⁶ aus Dallas, USA, ist eine der bekanntesten und meist verwendeten Agentenplattformen, obwohl Voyager eigentlich eine generelle Java-basierte Entwicklungsumgebung für verteilte Systeme darstellt [HGF98]. Voyager wird seit 1997 kontinuierlich weiterentwickelt. Für die Untersuchungen und Entwicklungen in dieser Arbeit wurde zuletzt die Version 3.2 eingesetzt, die 1999 erstmals erschienen ist. Die herausragende Stärke von Voyager ist, dass es gelungen ist, die Komplexität der Programmierung verteilter Systeme mit Hilfe der Programmiersprache Java zu verstecken. Für den Programmierer stehen eine Reihe von Funktionen zur Verfügung, wie

- die Mobilität von Programmcode durch dynamisches Laden von Programmklassen und Zugriff auf entfernte Klassen durch Proxy-Objekte,
- die Möglichkeit mit CORBA, RMI und DCOM-Programmen zu kommunizieren,
- ein universeller Naming Service²⁷,
- die Bereitstellung und Unterstützung von verschiedenen Nachrichtentypen
- und die automatische Reaktivierung von gespeicherten Objekten.

Hierfür müssen lediglich die entsprechenden Java-Klassen von Voyager neben einer JVM auf dem Rechner installiert werden. Ein weiterer Vorteil von Voyager ist, dass es auch auf Endgeräten mit sehr begrenztem Speicherplatz installiert werden kann. Weiterhin wird für eine sichere Kommunikation das SSL²⁸-Protokoll unterstützt.

Das Herz eines Voyager-Agentenumgebung ist der *Voyager Application Server*, der auf einer JVM aufsetzt (vgl. Abb. 3.3) und ein Ort des Agentensystems ist. Der Start eines Servers ist die Voraussetzung für alle Agentenaktivitäten. In einem verteilten System muss auf allen beteiligten Rechnern ein Voyager Server, der die jeweilige Heimat darstellt, laufen. Der Server kann durch einen Kommandozeilenbefehl oder ein Java-Programm mit der `startup()` Methode wie folgt gestartet werden:

```
Voyager.startup("8000");
```

Ein Aufruf eines Voyager Servers von einem entfernten Rechner aus ist nicht möglich. Als Parameter der `startup()` Methode wird die Portnummer für den Voyager Server übergeben. Jeder auf einem Rechner gestartete Voyager Server stellt einen Ort des Agentensystems dar, der über den dediziert vorzugebenden Kommunikationsport des IP-Netzes mit anderen Orten über Socket-Verbindungen²⁹ kommuniziert.

Für die Agentenkommunikation stellt Voyager spezielle Funktionen wie z.B. `moveTo()` zur Verfügung. Zudem existieren Mechanismen, die eine Agentenmigration absichern. Dazu gehört u.a., dass die zuvor unterbrochene Nachrichten-Verbreitung erneut aufgenommen

²⁶<http://www.objectspace.com>

²⁷Dienst, der den Namensraum verwaltet und dabei Namen und Daten einander zuordnet

²⁸Secure Socket Layer

²⁹Ein Socket ist ein Dienstzugangspunkt, an den auf der einen Seite eine Netzverbindung und auf der anderen Seite ein Anwendungsprozess gebunden werden kann. Sockets stellen dem Voyager-Agentensystem daher eine API für die Verwendung des IP-Protokolls zur Verfügung.

wird und beim Auftreten eines Fehlers der Originalzustand wieder hergestellt wird.

Ähnlich wie CORBA (siehe Kap. 4.2) benötigt Voyager zum Aufruf einer Klassenmethode in einer entfernten JVM eine Instanz der Rumpfklasse in der lokalen JVM [Obj97]. Eine Rumpf-Klasse dient als eine Schnittstelle zwischen dem aufrufenden und aufgerufenen Objekt, und in dieser Klasse sind alle *public* Methoden der aufgerufenen Klasse definiert. Eine Instanz der Rumpf-Klasse nimmt einen Aufruf entgegen, wandelt ihn in ein Datenformat um und transferiert diese Daten zum Zielobjekt. Abb. 3.11 zeigt diesen Vorgang schematisch für die Rumpf-Klasse `MyClass_proxy`. Rumpf-Klassen werden in Voyager auch als *Proxy-Klassen* bezeichnet, und eine Referenz einer Instanz einer Rumpf-Klasse heißt *Virtuelle Referenz*. In CORBA wird auf der Zielobjektseite eine Instanz der sogenannten Skelett-Klasse benötigt, um die Daten wieder in einen normalen Aufruf zurückzuwandeln und die Methode des Zielobjekts aufzurufen. Rumpf- und Skelett-Klassen kommunizieren miteinander über ein Kommunikationsprotokoll, wie z.B. IIOP³⁰. Voyager hingegen benutzt einen Agenten, um die Daten für den Aufruf zum Zielobjekt zu bringen. Ein Agent ist dabei in Voyager ein spezielles Objekt, das sowohl Mobilität als auch Autonomie besitzt, d.h. ein MA kann nach dem Starten ohne Kontakt mit seinem Besitzer durch das Kommunikationsnetz übertragen werden und seine Aufgaben selbständig erledigen. Mit Agenten hat Voyager einen eigenen Mechanismus (keine Verwandtschaft mit RMI und IIOP) zur Übertragung der Aufrufdaten entwickelt, wobei auf der Zielobjektseite keine Instanz der Skelett-Klasse notwendig ist. Wie in Abb. 3.11 dargestellt ist, werden die Ergebnisse des Aufrufs ebenfalls von einem MA zum aufrufenden System zurückgebracht.

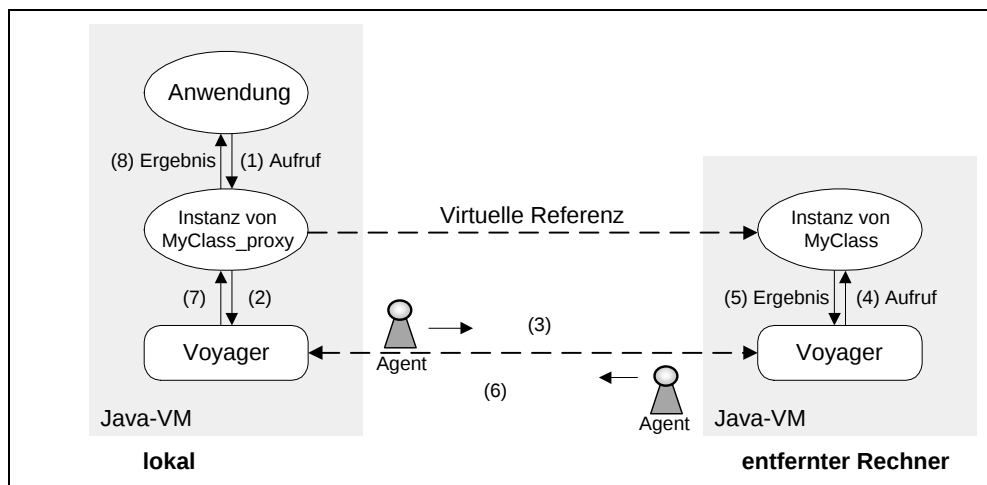


Abbildung 3.11: Erzeugen eines entfernten Objektes in Voyager

Voyager unterstützt folgende reguläre Nachrichtentypen (vgl. Kap. 3.3.1):

- Synchron-Nachrichten
- Future-Nachrichten
- One-Way-Nachrichten
- Multicast-Nachrichten

Die ersten drei Nachrichtentypen werden in Voyager durch einen sogenannten **Messenger** realisiert. Ein **Messenger** ist ein Mini-Agent, der den Inhalt der Nachricht zum Zielobjekt

³⁰Internet Inter-ORB Protocol, siehe auch Kap. 4.2

3.4.2 Grasshopper

Die Agentenplattform *Grasshopper* ist ein Produkt der Firma IKV++ aus Berlin [IKV01]. In dieser Arbeit wurde Grasshopper in der Version 1.2 für die Betriebssysteme Microsoft Windows und SUN Solaris untersucht. Ein wesentlicher Vorteil von Grasshopper ist, dass neben den für Agentensysteme üblichen APIs auch Werkzeuge zur Visualisierung von Laufzeitumgebungen und eine Regionsverwaltung mitgeliefert werden.

Abweichend von Kap. 3.2.3 wird im Grasshopper-System ein Ort lediglich als ein logischer Aufenthaltsort innerhalb der Grasshopper Laufzeitumgebung, einer sogenannten *Agentur* (agency) definiert, d.h. eine Agentur besteht aus einem oder mehreren Orten. Die genaue Adresse eines Ortes setzt sich bei Grasshopper folglich aus der Adresse der Laufzeitumgebung (Agentur) und dem Namen des Ortes zusammen. Außerdem wird in Grasshopper der Heimatort als *Information Desk* und das Agentensystem als *Distributed Agent Environment* (DAE) bezeichnet.

Wie Abb. 3.13 zeigt, besteht ein Agentensystem in Grasshopper aus Agenten, Orten und Agenturen. Dabei werden die Funktionen für Agenten von der Agentur bzw. durch die in ihr implementierten Dienste zur Verfügung gestellt. Die Agentur kann in Verbindung mit einer graphischen Oberfläche zur Verwaltung von Agenten und zur Konfiguration der Agentur gestartet werden. Jede Agentur besitzt mindestens einen Kommunikationskanal, d.h. beim Start werden ein oder mehrere *Ports* geöffnet, über die die Agentur mit anderen Agenturen kommunizieren kann. Grasshopper unterstützt HTTP, RMI und IIOP. Während die Kommunikation über Sockets mit SSL gesichert werden kann, können RMI und IIOP nicht durch Methoden von Grasshopper geschützt werden.

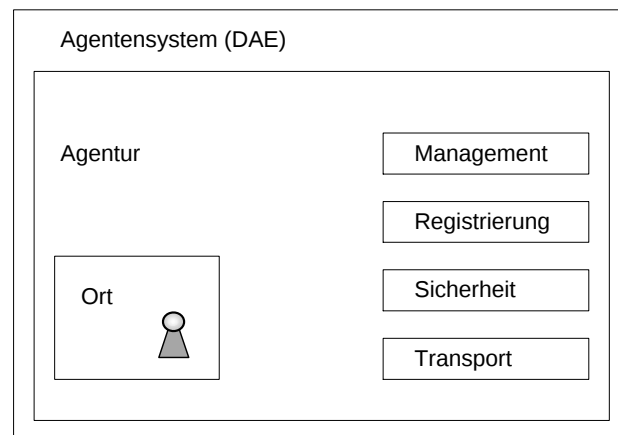


Abbildung 3.13: Grasshopper Agentensystem

Eine Region wird in Grasshopper durch die Registrierung von Agenturen bei einer speziellen Agentur, der *Region Registry*, aufgespannt. Hat ein Benutzer die dazu erforderlichen Rechte, kann er auf alle innerhalb dieser Region definierten Dienste zugreifen. Ändert sich der Zustand einer Agentur, werden Agenten gestartet oder beendet, so wird dies unmittelbar der Region Registry mitgeteilt. Jede Agentur ist aber auch in der Lage, ohne eine zentrale Verwaltung zu arbeiten. Allerdings stehen dann auch keine Suchfunktionen zu anderen Agenturen zur Verfügung. Über explizite Adressierung kann aber auch in diesem Fall eine Interoperabilität zwischen Agenten in entfernten Orten erzielt werden.

Durch Starten einer Region Registry wird ein Naming Service für die Elemente einer Region verfügbar und Suchfunktionen nach Agenten oder Agenturen können genutzt werden. Ist das Agentensystem für die Verwendung von CORBA-Klassen eingerichtet, wird u.a. der Einsatz von IIOP³² für die Kommunikation zwischen verschiedenen Agenturen und der Region Registry erlaubt. Die Region Registry kann dann sowohl innerhalb der durch sie aufgespannten Region als auch nach außen zur Speicherung, Pflege und Darstellung von Systeminformationen eingesetzt werden.

Um Agenten zu starten, muss deren Quelle explizit angegeben werden. Ein Codeserver zum automatischen Laden von Klassen ist bisher nicht vorhanden. Zusätzlich können Agenten, die lokal verfügbar sind und für die Einträge in einem Agentenkatalog erstellt wurden, ausgewählt und gestartet werden.

3.5 Standardisierung

Im vorangegangenen Abschnitt wurden zwei Agentensysteme mit ihren Eigenschaften vorgestellt. Nach intensiver Untersuchung verschiedenster Agentensysteme (siehe auch Anhang A und B) kann festgehalten werden, dass es schwer fällt, mehrere allgemeingültige Grundsätzen zu Agentensystemen unterschiedlicher Hersteller zu formulieren, da die Konzepte der existierenden Systeme sehr stark variieren. Betrachtet man die Gesamtheit der Agentensysteme, so sind gerade die Unterschiede in der verwendeten Programmiersprache, der Kommunikationsinfrastruktur, der Verteilung von Wissen zwischen stationären und mobilen Elementen, und den Schnittstellen zum Betriebssystem und den verwendeten Agenten auffällig. Gerade aber die Verfahren zum Austausch von Nachrichten und zur Übertragung von Programmcode bzw. Klassen, sowie die speziellen Betriebssystemunterstützungen, sind für die Konzeption von Agentensystem von entscheidender Bedeutung. Erschwerend wirkt sich auch häufig die Bindung an spezielle Systemumgebungen oder Programmiersprachen aus, da diese sehr oft kurzen Entwicklungszyklen unterliegen. Um die Anstrengungen für Softwareagenten nicht in einer Reihe proprietärer Insellösungen münden zu lassen und zur weiteren Verbreitung von Agentensystemen, bedarf es eines breiten Konsens in der Softwareindustrie. Vielerorts entstand daher der Wunsch nach entsprechenden Standardisierungsaktivitäten, die sich mit einer weitgehend implementierungsunabhängigen Konzeption von Agentensystemen und deren Systemcharakteristika beschäftigen und damit auch das Zusammenspiel verschiedener Agentenlösungen vereinfachen. Nachfolgend werden die entsprechenden Bemühungen der *Foundation for Intelligent Physical Agents* (FIPA) und der *Object Management Group* (OMG) kurz vorgestellt.

3.5.1 FIPA - Foundation for Intelligent Physical Agents

FIPA ist eine nicht kommerzielle Vereinigung mit Sitz in Genf, Schweiz. Die offizielle Gründung erfolgte am 1996. Heute hat die FIPA 59 Mitglieder aus 15 Ländern³³, darunter namhafte Unternehmen wie British Telecom, France Telecom, Fujitsu, Hewlett Packard, IBM, Lockheed Martin, Lucent, Mitsubishi, Motorola, Nortel Networks, Siemens, Sun, Telia und Toshiba. Das Ziel der FIPA ist es, durch rechtzeitige, international anerkannte Spezifikationen die Zusammenarbeit von agentenbasierten Anwendungen, Diensten und Geräten zu gewährleisten [FIP97].

³²Standardprotokoll für die Kommunikation zwischen CORBA-Produkten verschiedener Hersteller

³³Stand: Juli 2001

Die Grundlage aller FIPA-Standards ist die FIPA 97-Spezifikation. FIPA 97 besteht aus sieben Teilen, von denen die ersten drei Teile die Agententechnologie spezifizieren (Agentenmanagement, Agentenkommunikationssprache ACL, Agentensoftwareintegration), während sich die anderen vier Teile mit speziellen Anwendungen beschäftigen (Reiseassistent³⁴, Sekretär³⁵, Multimediaunterhaltung³⁶, sowie Netzmanagement³⁷). FIPA 98 und FIPA 99 erweiterten diesen Grundstamm, so dass das Gesamtwerk der FIPA-Spezifikationen heute 18 Teile umfasst.

3.5.2 OMG - Object Management Group

OMG ist ein offener und nicht kommerzieller Zusammenschluss von Hardware- und Softwareherstellern, Netzbetreibern sowie gewerblichen Nutzern von Computersoftware. Unter den über 800 Mitgliedern findet man neben Forschungseinrichtungen und Lehrstühlen fast alle namhaften Firmen der Informations- und der Telekommunikationsindustrie mit Ausnahme von Microsoft, die mit ihrer Eigenentwicklung COM/DCOM in direkter Konkurrenz zur OMG steht. Seit der Gründung der OMG im Mai 1989 ist aus dem Bestreben herstellernabhängige Spezifikationen zu entwickeln vor allem der CORBA-Standard (Kap. 4.2) entstanden. Als Basis für die Standardisierung dienen reale Lösungen. Dadurch bleibt bei der Konstruktion umfassender Modelle der Kontakt zu realen Szenarien gewahrt.

Die *Mobile Agent System Interoperability Facilities Spezifikation* (MASIF) ist ein Standard der OMG, der sich mit mobile Agenten und Agentensysteme beschäftigt [OMG98]. MASIF integriert die Agententechnologie in CORBA, damit Agentensysteme unterschiedlicher Hersteller über einen OMG ORB zusammenarbeiten können. Der MASIF-Standard besteht aus drei Teilen. Im ersten Teil werden Agenten, Agentensysteme, Agentenübertragung etc. definiert. Dies dient als Grundlage und Ausgangspunkt für die nachfolgenden Abschnitte. Im zweiten Teil werden die agentenrelevanten CORBA-Dienste, u.a. der *Naming Service* und der *Security Service* diskutiert. Der letzte Teil des Standards ist eine Sammlung von IDL-Definitionen (vgl. Kap. 4.2) und Schnittstellen, die eine Zusammenarbeit zwischen verschiedenen Agenten und Agentensystemen ermöglichen. Eine genauere Betrachtung der Standards CORBA und MASIF erfolgt im nächsten Kapitel.

3.6 Zusammenfassung

In diesem Kapitel wurde in das informationstechnische Gebiet der Agententechnologie eingeführt. Dafür wurden die Aufgaben und die Funktionsweise eines Softwareagenten erläutert und die dieser Arbeit zugrundeliegenden Definitionen für agentenbasierte Systemarchitekturen vorgenommen. Mobile Agenten werden dabei als eine aus Programmcode, Datensätzen und Laufzeitinformationen bestehende Einheit beschrieben, die zwischen Netzknoten mit unterschiedlichen Betriebssystemen über verschiedene Trägerdiensten übertragen werden kann. Die gängigen Kommunikationsmechanismen, Kommunikationssprachen und Programmiersprachen der Agententechnologie wurden in diesem Abschnitt ebenso beschrieben, wie die Agentensysteme Voyager und Grasshopper, die für die prototypische Realisierung des neuen, agentenbasierten Kommunikationskonzeptes eingesetzt worden sind. Abschließend wurden die Standardisierungsbemühungen der FIPA und OMG im Bereich der Agententechnologie kurz erläutert.

³⁴Personal Travel Assistance

³⁵Personal Assistant

³⁶Audio/Video Entertainment & Boardcasting

³⁷Network Management & Provisioning

CORBA-Unterstützung für agentenbasierte Kommunikationskonzepte

Die Möglichkeiten, Agenten zur Dienstleistung einzusetzen, sind bei heutigen Agentensystemen auf den Gültigkeitsbereich des Agentensystems beschränkt, vgl. Abb. 3.3. Innerhalb dieses Bereiches stehen z.B. eine Registrierung für Dienste und Agenten, sowie eine Liste der verfügbaren Orte zur Verfügung. Wie Abb. 4.1 zeigt, ist über die Systemgrenzen hinaus in der Regel keine Zusammenarbeit zwischen verschiedenen Agentensystemen möglich. In diesem Kapitel werden mit Hilfe des CORBA-Standards Konzepte erörtert, wie eine Dienstleistung auch außerhalb der Grenzen eines Agentensystems ermöglicht werden kann.

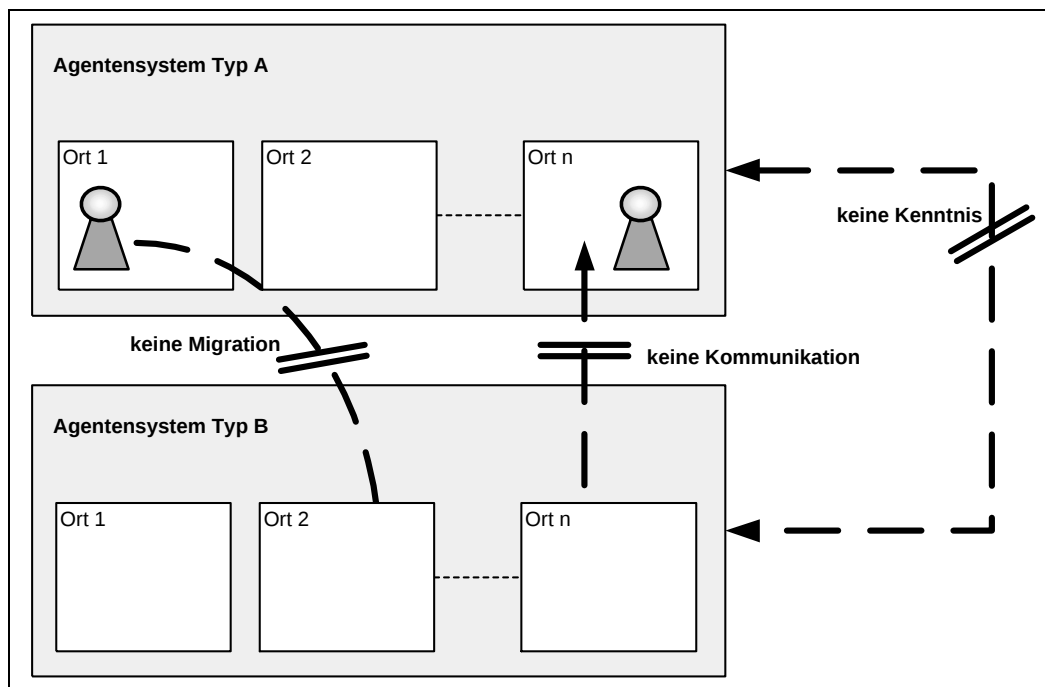


Abbildung 4.1: Grenzen für die Zusammenarbeit zwischen Agentensystemen

4.1 Grenzen von Agentensystemen

Faßt man die Agentenwelt als die Summe von Orten, Systemdiensten und gemeinsamen Kommunikations- oder Verwaltungsmechanismen auf, so existieren die folgenden sechs Konfliktpotenziale, die erhebliche Konsequenzen für die Interoperabilität von unterschiedlichen Agentensystemen haben:

Betriebssystem: Orte und Agentensystem liegen logisch und funktionell zwischen den Agenten und einem Betriebssystem, vgl. Abb. 3.3. Obwohl kein direkter Kontakt

besteht, benötigen Agenten häufig Funktionen des Betriebssystems, um z.B. von einem Netzknoten zu einem anderen migrieren zu können.

Programmiersprache: Auch wenn alle modernen Prototypen von Agentensystemen in JAVA implementiert wurden, so kommen mit TCL/TK und C++ auch weitere Sprachen für die Programmierung von MAs in Frage. Ein Ort kann jedoch ausschließlich Agenten des zu ihm passenden Typs verarbeiten, so daß die Wahl der Programmiersprache die Einsatzmöglichkeiten beschränkt. Diese Begrenzung schließt teilweise sogar verschiedene Versionen einer Sprache voneinander aus.

Systemarchitektur: Betrachtet man Agentensysteme getrennt voneinander, werden die Begriffe *Agentensystem*, *Region* und *Ort* häufig unterschiedlich voneinander definiert und interpretiert. Diese unterschiedliche Zuweisung von Bedeutungen zu den Begriffen hat zur Folge, daß die Verarbeitung von Ortsangaben des einen Systems von einem zweiten Agentensystem nicht ohne zusätzliche Angaben möglich ist.

Agentensprachen: Für Agentensysteme wurden unterschiedliche Kommunikationssprachen entworfen (vgl. Kap. 3.3.4), die leider nur unzureichend aufeinander abgebildet werden können.

Administration: Die Verwaltung von existierenden Agentensystemen ist häufig proprietär organisiert. Weder die Verfahren zur Speicherung von Daten zu verfügbaren Orten, noch von deren Eigenschaften und der jeweiligen Agenten sind einheitlich. Agentensystemtyp A ist daher in der Regel nicht in der Lage, einen neuen Ort im Agentensystem vom Typ B zu registrieren.

Datenhaltung: Softwareagenten werden Informationen durch den Programmierer, den Anwender oder durch andere Agenten über die Systemumgebung vermittelt. Existiert innerhalb eines Systems A ein Naming Service, der von Agenten zur Erkundung des Systems und der Umgebung eingesetzt werden kann, können Agenten dieses Systems meist ähnliche Informationen von einem Dienstleistungsagenten in einem System B [Döl98] nicht erfragen.

Bei Überlegungen zu Gemeinsamkeiten und zur Interoperabilität von Agentensystemen sind diese Begrenzungen stets zu berücksichtigen. Aufgrund der recht großen Variationsbreite heutiger Agentensysteme, erscheint ein für alle Architekturen geltender umfassender Standard schwierig zu realisieren. Die OMG hat sich in den vergangenen Jahren bemüht, die Agententechnologie in die *Common Object Request Broker Architecture* (CORBA) zu integrieren, um die Interoperabilität von unterschiedlichen Agentensystemen zu ermöglichen. In den folgenden Abschnitten werden einige grundlegende Informationen zum Verständnis von CORBA und der *Mobile Agent System Interoperability Facility* (MASIF) der OMG gegeben.

4.2 CORBA-Standard

CORBA ermöglicht unterschiedlichen, verteilten Anwendungen miteinander zu kommunizieren, ohne dass diese voneinander wissen wie sie konzipiert und wo sie physikalisch abgelegt sind.

CORBA 1.1 wurde 1991 von der OMG herausgegeben und definiert die Schnittstellenbeschreibungssprache *Interface Definition Language* (IDL) und das Zusammenspiel zwischen Client- und Serverobjekten mit Hilfe eines *Object Request Brokers* (ORB). In der im Dezember 1994 veröffentlichten Version 2.0 des CORBA-Standards wurde erstmals spezifiziert, wie ORBs verschiedener Hersteller zusammenarbeiten sollen. Den Kern von CORBA bildet dabei die Definition eines funktionalen Modells. Eine Anwendung besteht hiernach aus drei verschiedenen Komponenten:

- Client (-objekt)
- Server (-objekt)
- Object Request Broker (ORB)

Dabei ist der ORB der zentrale Kern eines CORBA-Systems.

Ein **ORB** vermittelt als *Middleware* Anfragen zwischen Client- und Serverobjekten.

Dabei ist es nicht notwendig, dass Client und Server in derselben Programmiersprache und für das gleiche Betriebssystem implementiert sind. Weiterhin ist auch eine örtliche Verteilung von einem oder mehreren Serverobjekten und einem oder mehreren Clientobjekten möglich. Die Aufgaben eines CORBA ORBs sind also:

- Umsetzung zwischen verschiedenen Programmiersprachen
- Auffinden von gesuchten Objekten
- Transport von Nachrichten zwischen Client und Server

Beim Entwurf von Client und Server ist daher als zusätzliche Aufgabe die Definition der Schnittstelle zwischen diesen beiden zu berücksichtigen. Das Ergebnis eines so erweiterten Entwurfsprozesses ist eine dreischichtige Softwarearchitektur, deren grundsätzlicher Aufbau in Abb. 4.2 dargestellt ist [Lül99].

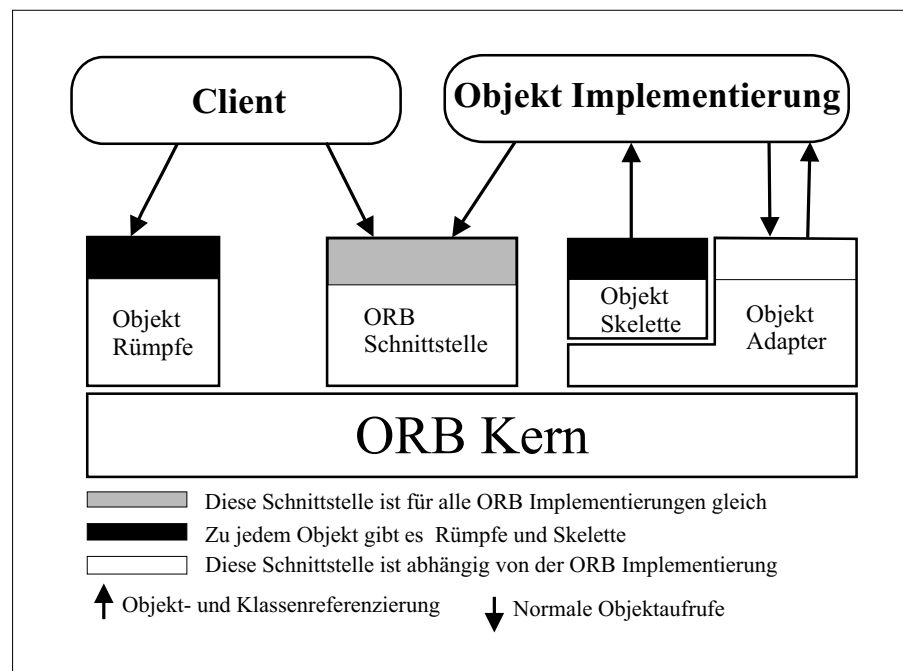


Abbildung 4.2: Schematische Darstellung der Funktion eines CORBA ORBs

Dem Kern von CORBA zuzuordnen ist auch IDL, denn die logische Aufteilung von Anwendungen in Client und Server erfordert eine weitgehend genaue Betrachtung der zwischen diesen Komponenten auszutauschenden Informationen. IDL wird in diesem Zusammenhang für die von speziellen Spracheigenschaften unabhängige Beschreibung der Schnittstellen verwendet. Die Beschreibung erfolgt derart, dass die Datentypen und die Funktionen, die in einer Schnittstelle abgebildet werden, in einer C++ ähnlichen Syntax definiert werden. Ähnlich zur Vererbung in C++ können in IDL spezifizierte Schnittstellen erweitert werden. Damit wird ein hohes Maß an Auf- bzw. Abwärtskompatibilität gewährleistet.

Mehr als ein Drittel des CORBA-Standards beschäftigt sich mit der Übersetzung von IDL-Beschreibungen in verschiedene Programmiersprachen, wie z.B. C, C++, Cobol, Java und Smalltalk. Es wird festgelegt, in welche Sprachidiome ein Ausdruck in IDL in der jeweiligen Zielsprache umgesetzt wird. Sprachspezifische IDL-zu-Sprache Übersetzer wie *IDL2java* oder *IDLtocpp* sorgen dann für die richtige Übersetzung in die jeweilige Programmiersprache. Soll ein Serverobjekt in C++ programmiert werden, erzeugt ein IDL-zu-C++ Übersetzer Codeskelette, die in den Serverprogrammcode integriert werden können. Diese Codegerüste sind sowohl sprach- als auch ORB-spezifisch. Skelette, die mit einem Compiler vom ORB A erzeugt werden, können in der Regel nicht zum Aufbau eines Servers vom Typ B verwendet werden. Sprachunabhängig sind lediglich die Nachrichten, die von den ORB-Klassen auf Serverseite nach Umsetzung durch den Skelettcode versendet werden. Aus Sicht eines Programmierers entsprechen die Codeskelette (auf der Serverseite) bzw. -rümpfe (auf der Clientseite) Schnittstellen, die lokal verfügbar sind. Eine lokale Anwendung kann daher nicht feststellen, ob die eingebundenen Funktionen in lokal gespeicherten Bibliotheken oder in fernen Objekten realisiert sind. Server bzw. Client arbeiten deshalb nur mit lokalen Nachrichten.

Die IDL-Definitionen schreiben nicht vor, wie Funktionen implementiert werden sollen. Eine Zuweisung der Aufgaben von Client und Server muss folglich parallel zum IDL-Entwurfsprozess geschehen. Am Ende des Designs steht die Umsetzung von IDL-Code in die jeweilige Sprache von Client und Server. Der auf diese Weise erzeugte Code wird ebenso wie CORBA-Klassen auf der Client- und auf der Serverseite eingebunden. Die CORBA-Klassen beinhalten dabei das, was aus der Sicht von Client oder Server als *ORB* bezeichnet wird.

Für die ortstransparente Kommunikation definiert der CORBA-Standard das *General Inter ORB Protocol* (GIOP) und als Spezialisierung dieses Protokolls das *Internet Inter ORB Protokoll* (IIOP), das auf TCP/IP aufsetzt. GIOP ist für den Einsatz mit verbindungsorientiert arbeitenden Protokollen vorgesehen. Für die CORBA-Konformität sind sowohl GIOP als auch IIOP gefordert. IIOP stellt heute das Standardprotokoll zur Kommunikation zwischen CORBA-Produkten verschiedener Hersteller dar.

Über das Konzept der reinen Vermittlung von Objektreferenzen und ihrer sprachlichen Umsetzung hinaus definiert der CORBA-Standard *Common Object Services* (COS). In COS sind insgesamt 15 Dienste beschrieben, die neben der Kernfunktionalität unterstützend zur Entwicklung und zum Einsatz verteilter Anwendungen eingesetzt werden können. Der wichtigste von ihnen ist der *Naming Service*. Er dient dazu, Objektkennungen, die in CORBA zur Benennung und Identifizierung von Objekten verwendet werden, zentral zu speichern und die Anfragen nach Objekten und Methoden diesen Kennungen zuzuordnen.

Für den Einsatz in Agentenplattformen bieten sich vor allem die folgenden drei Bestandteile des CORBA-Standards an:

- IDL für die sprachneutrale, implementierungsunabhängige Definition von Schnittstellen und die genaue Zuweisung von Aufgaben in verteilten Systemen,
- IIOP als einfaches Protokoll zur Kommunikation zwischen Orten, Agentensystemen und Verwaltungselementen und
- COS Naming für die Benennung und Identifikation von Agenten.

Ferner kann durch CORBA eine elegante Integration lokal verfügbarer Funktionen in die Laufzeitumgebung von Agenten ermöglicht werden. Agentenplattformen wie Voyager und Grasshopper erlauben daher schon heute eine Kommunikation mit CORBA-Objekten.

4.3 MASIF-Standard

Die *Mobile Agent System Interoperability Facility* (MASIF) der OMG [OMG98] basiert auf dem CORBA-Standard.

MASIF definiert einen logischen Rahmen für den Entwurf von Agentensystemen, mit dem eine Zusammenarbeit von verschiedenen Agentensystemen vereinfacht möglich wird.

Gegenstand von MASIF sind vor allem die folgenden fünf Bereiche:

- **Agentenmanagement:** Damit Agenten auf verschiedenen Plattformen mit Standardmethoden erzeugt, beendet und transferiert werden können, soll ein systemübergreifendes Agentenmanagement erreicht werden. Ziel der Standardisierung ist es daher gemeinsame Funktionen zu definieren, die zum Management von unterschiedlichen Systemen eingesetzt werden können.
- **Agentenmigration:** Die Lokalität des auszuführenden Codes ist ein wesentlicher Vorteil von Agentensystemen. Dadurch lassen sich in der Regel die Anzahl der Transaktionen im Netz und die Antwortzeiten minimieren. Durch eine standardisierte Agentenmigration wird es für Systeme verschiedenen Typs möglich, Agenten eines jeweils anderen Typs aufzunehmen oder auszuführen.
- **Agenten- und Agentensystem-Namen:** Ein grundsätzlicher Punkt ist die Verwendung gleicher Namen für Agentensysteme. Um eine reibungslose Zusammenarbeit zu erreichen, sollten Agentensysteme zur Benennung ihrer Objekte die gleiche Systematik verwenden. Ferner sollte die Kennung eines Agenten auch über die Grenze eines Systems hinaus Gültigkeit haben und eindeutig zuzuordnen sein.
- **Agentensystemtypen:** Eine Klassifizierung möglicher Agentensystemtypen ist wichtig, um durch eine einfache Typprüfung feststellen zu können, ob der Transfer eines Agenten des Typs A zu einem System des Typs B überhaupt möglich ist. Welche Eigenschaften einen Systemtyp definieren, und wie diese zu organisieren sind, ist Bestandteil von MASIF.
- **Syntax zur Ortsbeschreibung:** Für die Syntax zur Beschreibung von Quell- und Zieladressen sind verschiedene Formate vorstellbar. Die Festlegung einer verbindlichen Form ermöglicht systemübergreifende Aktionen. Analog zu den Systemnamen und der Festlegung von Systemtypen ist zu definieren, was Bestandteil einer Adresse sein soll und wie diese zu strukturieren ist. Zusätzlich ist der Einsatz einer zentralen Registrierung für Namen in Agentensystemen angedacht.

Explizit ausgeklammert werden die sprachlichen Aspekte der Agentenkommunikation, die unter anderem von der FIPA (Kap. 3.5.1) umfassend diskutiert werden. Außerdem werden in MASIF nur Szenarien betrachtet, in denen Agenten direkt von einer Region in eine andere wechseln, oder in diesem Sinne direkt miteinander kommunizieren. Die Problemstellung des Multi-Hop¹-Agententransfers ist ebenfalls nicht Bestandteil des MASIF-Standards. Insgesamt sind damit die Möglichkeiten, mit Hilfe des MASIF-Standards eine flexible, systemübergreifende agentenbasierte Dienstleistung zu realisieren schon deutlich eingeschränkt.

Im Kern besteht der MASIF-Standard aus zwei Schnittstellen, mit denen die systemübergreifende Verwaltung von Agentensystemen und MAs ermöglicht werden soll:

- *MAFFinder* und
- *MAFAgentSystem*

Abb. 4.3 zeigt, wie ein Client eines Agentensystems über einen ORB auf diese beiden Schnittstellen zugreift. Dabei ist es zweitrangig, ob es sich bei diesem Client um ein zweites System, eine Anwendung oder einen Agenten einer fernen Plattform handelt. Bei der im Rahmen dieser Arbeit entstandenen MASIF-konformen Verbindung zwischen der Grasshopper- und Voyager-Agentenplattform wurde eine Interoperabilität nach MASIF mit dem neu konzipierten und implementierten *MASIF-Port* beim MAF-Client realisiert. Die Aufgabe und Funktionsweise des MASIF-Ports wird in Kap. 4.5 beschrieben.

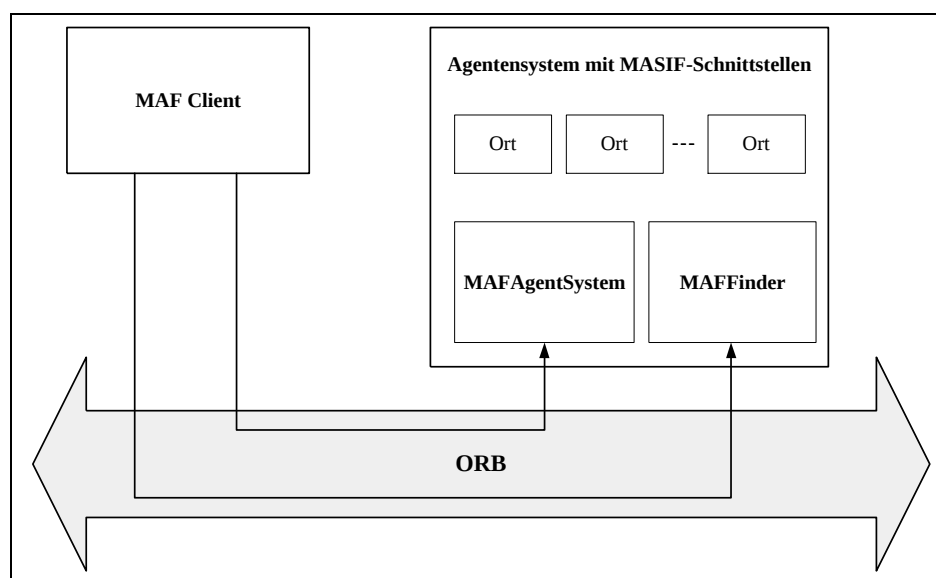


Abbildung 4.3: Zugriff auf die Schnittstellen eines MASIF-Systems

MAFFinder

Der MAFFinder ist ein Naming Service, der eine Liste registrierter Agentensysteme führt und die Zugriffsmethoden und Suchfunktionen der Agentensysteme in einer Schnittstelle definiert. Jede Region eines MASIF-basierten Agentensystems besitzt einen MAFFinder. Funktionen des MAFFinders sind beispielsweise **register**, **lookup** und **unregister**. Bei

¹Multi-Hop: Zum Erreichen des Ziels ist mehr als ein Übergang von Region zu Region notwendig

diesen Funktionen wird der Name eines Agenten, als ein strukturierter Datentyp, der aus den folgenden drei Komponenten besteht, übergeben:

- `authority`
- `identity`
- `agent_system_type`

Während die ersten beiden Komponenten den Agenten des Systems beschreiben, wird durch den Bestandteil `agent_system_type` die Herkunft des Agenten angegeben. Unter der Voraussetzung, dass Agentensysteme gleichen Typs keine Kennung zweimal vergeben, sind damit Namen auf jeden Fall eindeutig. Sollten zwei Systeme unterschiedlichen Typs zufällig die gleiche Kennung für Agenten generieren, dann können diese anhand des Systemtyps voneinander unterschieden werden.

MAFAgentSystem

Während der MAFFinder eine Regionsverwaltung darstellt, ist die MAFAgentSystem-Schnittstelle für die agentenorientierte Kommunikation zwischen verschiedenen Agentensystemen gedacht. Ein Agentensystem besteht aus einer Laufzeitumgebung mit einem oder mehreren Orten (siehe auch Kap. 3.2.3). Die Migration von Agenten zwischen Orten von verschiedenen Agentensystemen kann physikalisch als Migration von MAs zwischen zwei Netzknoten oder zwischen zwei Agentensystemen auf einem Netzknoten interpretiert werden. Orte, die dem selben Netzknoten zuzuordnen sind, müssen folglich nicht zwingend demselben Agentensystem angehören. OMG definiert im MASIF-Standard die folgenden 14 Funktionen zur Verwaltung von Agenten und Agentensystemen in einer heterogenen Agentenwelt:

- `create_agent ()`
- `fetch_class ()`
- `find_nearby_agent_system_of_profile ()`
- `get_agent_status ()`
- `get_agent_system_info()`
- `get_authinfo ()`
- `get_MAFFinder ()`
- `list_all_agents()`
- `list_all_agents_of_authority ()`
- `list_all_places()`
- `receive_agent ()`
- `resume_agent ()`
- `suspend_agent ()`
- `terminate_agent ()`

Die teilweise erforderlichen Parameter wurden hier aus Gründen besserer Darstellung weggelassen. Für die vollständige Erklärung, die Parameterlisten und die Rückgabewerte sei hier auf den Standard [OMG98] verwiesen.

4.4 Einsatz von CORBA im Agentenkontext

CORBA ist ein genereller Ansatz zur objektorientierten Programmierung, bei dem die Speicherstelle eines Objektes, ebenso wie die verwendete Programmiersprache, für den Anwender und den Programmierer eine untergeordnete Rolle spielt. Die Agententechnologie und der CORBA-Standard verfolgen damit im Hinblick auf ihre Systemunabhängigkeit

ähnliche Ziele. Bei Agentensystemen ist die Verfügbarkeit von Programmcode (Klassen) jedoch in der Regel auf das geschlossene System begrenzt, so kann man zwar Code von einem entfernten Ort laden, um ihn lokal auszuführen, doch der Ort muss innerhalb der Systemgrenzen liegen. Funktionen, die nicht innerhalb eines Agentensystems verfügbar sind, können nur durch eine Erweiterungen des Systemdesigns eingebaut werden. So gibt es z.B. im Agentensystem Voyager (Kap. 3.4.1) eine CORBA-Schnittstelle. Die Verbindung von Objekten innerhalb von Voyager zu externen CORBA-Objekten wird dabei in einem CORBA-ORB realisiert.

Dem Konzept des Transportes von Klassen setzt CORBA den Transport von Nachrichten und eine Umsetzung auf die jeweils verwendete Programmiersprache entgegen. Dies entspricht einem universellem RPC. Neben RPCs können CORBA-Objekte, ähnlich wie Agenten, auch über asynchrone Nachrichten oder über Events (Kap. 3.3.2) miteinander kommunizieren. Eine Kommunikation über ein Nachrichtendepot, wie es in einigen Agentensystemen mit sogenannten *Blackboards* (Kap. 3.3.3) realisiert werden kann, gibt es in CORBA allerdings nicht.

In den folgenden Abschnitten wird beschrieben, wie der CORBA-Standard das im Rahmen der Arbeit neu entwickelte agentenbasierte Kommunikationskonzept bei der Dienstleistung in heterogenen Kommunikationsnetzen unterstützen kann.

4.4.1 Benutzerschnittstelle

Ein möglicher Synergieeffekt bei der Kombination des Agentenparadigmas mit CORBA liegt in der Konzeption der Benutzerschnittstelle. Da es heute und (wahrscheinlich) auch in Zukunft eine Vielzahl von verschiedenen Endgeräten mit unterschiedlichsten Leistungsmerkmalen gibt bzw. geben wird, kann nicht von standardisierten Schnittstellen auf den Endgeräten ausgegangen werden. Interpretiert man die Benutzerschnittstelle (Dienstdarsteller) als einen eigenen Netzknoten, so sind beim entworfenen, agentenbasierten Kommunikationskonzept mindestens drei Knoten mit gegebenenfalls unterschiedlichen Systemplattformen bei der Dienstleistung involviert, siehe Abb. 4.4.

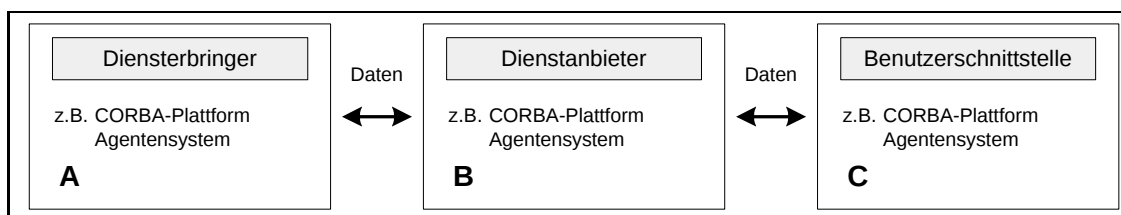


Abbildung 4.4: Netzknoten bei der agentenbasierten Dienstleistung

Nach einer genauen Analyse des Datenflusses zwischen den einzelnen Komponenten kann die Benutzerschnittstelle als Client realisiert werden, dessen Dienste als Serverobjekte ausgeführt werden. Grundsätzlich besteht keine Veranlassung, dabei nur von Diensten eines System- oder Architekturtyps auszugehen. Eine Kombination von Agentendiensten, CORBA-Diensten und sonstigen Netzdiensten ist in diesem aufgesplitteten Szenario vorstellbar. So ist möglich, daß die spezifischen, zu einem Dienst gehörenden Informationen in IDL beschrieben werden, und die Umwandlung in eine dem Benutzer und dem Endgerät verständliche Form erst im Knoten B oder C erfolgt.

Bei heutigen Agentenplattformen besteht meist keine klare Trennung zwischen der Benutzerschnittstelle und der Dienstleistung. Dies kann an einem typischen Startprozess eines agentenbasierten Dienstes, wie er auch beim implementierten AHB-Dienst (Kap. 5.3) verwendet wird, verdeutlicht werden:

1. Das Agentensystem wird gestartet. Dieser Prozess kann in der Regel nicht von einem entfernten Ort aus geschehen.
2. Der Benutzer meldet sich beim System an.
3. Der authentifizierte Benutzer (Teilnehmer) erhält sein zugeschnittenes, persönliches (personalisiertes) Dienstangebot, d.h. er kann nur die Dienste nutzen, für die er Zugangsrechte besitzt. Aus diesem personalisierten Dienstangebot kann der Benutzer nun einen oder mehrere Dienste auswählen und starten.
4. Die Dienste werden genutzt.

Die Interaktion mit Benutzern sowie die Darstellung von Information gehört nicht zu den Kernfunktionen und neuen Kommunikationskonzepten von Agentensystemen. Agentensysteme führen Aufträge ihrer Benutzer aus und erhalten dazu notwendige Informationen vom Benutzer, zentralen Registern und Diensten der jeweiligen Plattform. Der einzige zusätzliche Kontakt eines Benutzers mit einem Agentensystem sollte deshalb die Anmeldung sein. Diese Anforderung wurde beim implementierten ASB-Dienst (Kap. 5.4) berücksichtigt.

Trennt man in einem System die Datenein- bzw. -ausgabe von der Dienstauführung, so nähert man sich dem eigentlichen CORBA-Ansatz, bei dem getrennt voneinander entwickelte Komponenten zusammengeführt werden sollen. Um die Personalisierung eines Dienstes zu erreichen, bieten sich Benutzeragenten (Kap. 3.2.4) zur Verwaltung eines Benutzerprofils an, in dem persönliche Einstellungen wie Aus- und Eingabepreferenzen, aber auch persönliche Daten (Interessen, Präferenzen, Gewohnheiten, etc.) und eine Dienstliste abgelegt sein können. Schließlich können auch Dienste, die auf Basis verschiedener Agentenplattformen entwickelt wurden, eingebunden werden, wenn diese in IDL spezifiziert sind und eine CORBA-Schnittstelle existiert. Abb. 4.5 zeigt, wie verschiedene agentenbasierte Dienste mit Hilfe eines ORBs auf unterschiedlichen Endgeräten angeboten werden können.

Exemplarisch könnte so ein *Persönlicher Informationsdienst* konzipiert werden, dessen Benutzerschnittstelle in einer zum Endgerät passenden Sprache, z.B. WML für ein WAP-fähiges Mobiltelefon, programmiert wird. Für den Benutzer ist die Trennung zwischen der Informationsdarstellung und der -verarbeitung nicht sichtbar. Ob für das Nachschlagen in elektronisch verfügbaren Daten ein agentenbasierter Ansatz oder RPCs verwendet werden, ist ebenfalls für den Benutzer nicht zu erkennen. Der Vorteil dieses Ansatzes liegt in der Austauschbarkeit von Implementierung und Infrastruktur. Verschiedene Clients stellen so die gleichen Funktionen zur Verfügung und die Realisierung dieser Funktionen über unterschiedliche Kommunikationsnetze (LAN, WAN, GSM etc.) sind denkbar. Lediglich die unterschiedlichen Dienstaufführungszeiten würden von den Benutzern wahrgenommen.

4.4.2 IIOP für die Agentenkommunikation

Der CORBA-Standard sieht zur Kommunikation zwischen Objekten eine Zwischenschicht vor, die Details der Implementierung und deren Ort abstrahiert. Um einen Zugriff auf ferne Objekte zu ermöglichen, muß ein Kommunikationsmechanismus zwischen verschiedenen Orten definiert werden. Dies gilt sowohl für den Start einer verteilten Umgebung, als auch für deren Betrieb. Bei einfachen CORBA-Implementierungen ist ein Zugriff auf ferne Objekte oft nur dann möglich, wenn ein Client genaue Informationen besitzt, wo die

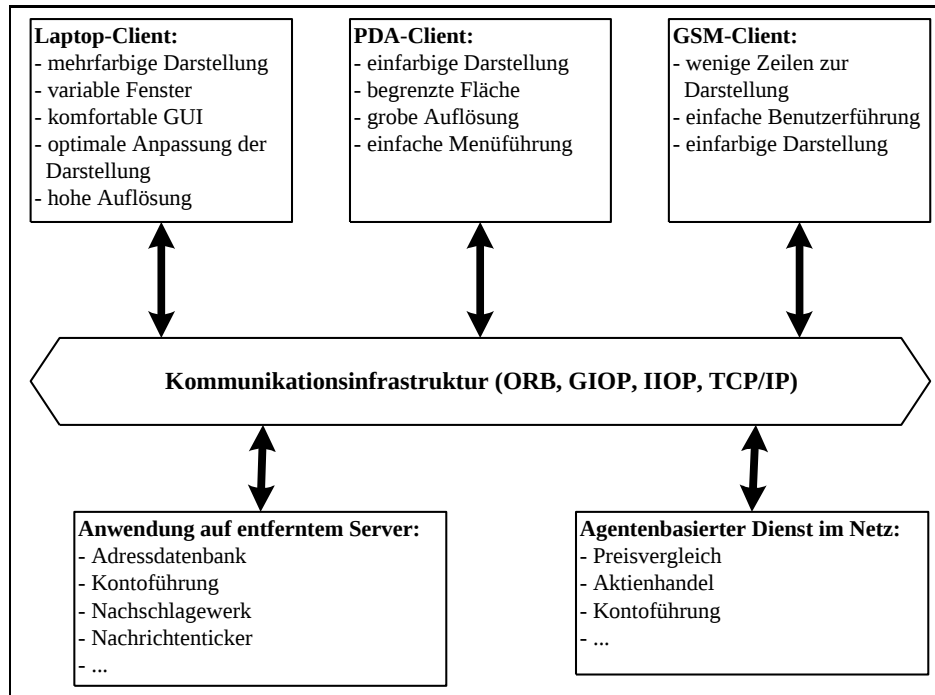


Abbildung 4.5: Verschiedene Endgeräte nutzen die gleichen Dienste

gewünschten Funktionen verfügbar sind.

In Agentensystemen ist zwischen einzelnen Orten einer Region ebenfalls eine Kommunikationsinfrastruktur vorzusehen. Über diese müssen die Abstimmung des Namenskontexts, die ortstransparente Agentenkommunikation und schließlich auch der Transport von Laufzeiteinformationen und Klassen abgewickelt werden. Der Austausch dieser Information wird von den existierenden Agentensystemen unterschiedlich gelöst. Daher können Orte verschiedener Agentensysteme in der Regel auch nicht miteinander kommunizieren. Um hier Abhilfe zu schaffen, wurde der Einsatz von IIOP als Kommunikationsprotokoll untersucht.

IIOP definiert ein Nachrichtenformat für ein weites Spektrum von Anforderungen. Es bietet sich vor allem wegen seiner Schlichtheit an und stellt eine geeignete Basis für eine allgemeine *Ort-zu-Ort-Kommunikation* dar. Eine derartige Realisierung bietet die Agentenplattform Grasshopper (Kap. 3.4.2). Neben RMI und Sockets wird dort auch IIOP als Kommunikationsprotokoll zwischen verschiedenen Orten eingesetzt.

Eine weitere interessante Perspektive für den Einsatz von IIOP für die *Ort-zu-Ort-Kommunikation* ist die Möglichkeit zum Tunneln von Firewalls². So ist es verhältnismäßig einfach, eine firmeninterne Region aufzubauen, die im Firmennetz durch Firewallssysteme geschützt wird. Soll nun ein externer Ort Kontakt zum Firmensystem erhalten, so ist es mit IIOP-Gateways möglich, die z.B. IIOP auf HTTP umsetzen, Anfragen durch die Firewallssysteme hindurch zu vermitteln. Für alternative Mechanismen wie RMI und Sockets, die bei gängigen Agentensystemen häufig Verwendung finden, bedeutet dies ein erheblich größerer Aufwand.

²Firewalls schützen bestimmte zugeordnete Domänen von Kommunikationsnetzen durch ihre Sicherheitsfunktionen vor unerwünschtem Zugriff (z.B. aus dem Internet).

4.4.3 Kooperation von Agentensystemen verschiedenen Typs

Die Kooperation und Kommunikation zwischen verschiedenen Agentensystemen, sowie das Management über System- oder Regionsgrenzen hinweg erfordern weitgehende Überlegungen zu vorhandenen Gemeinsamkeiten und möglichen Schnittstellen der Agentensysteme. Je unterschiedlicher zwei Agentensysteme sind, desto geringer wird die erzielbare Interoperabilität sein. Zunächst ist festzulegen, welche Informationen über die Grenzen eines Agentensystems hinaus ausgetauscht werden sollen. Ergebnisse solcher Analyseprozesse stellen sowohl die Spezifikationen der FIPA in [FIP98a] und als auch die OMG in [OMG98] vor.

Die Schwerpunkte dieser Betrachtungen liegen in unterschiedlichen Bereichen. Während die FIPA einen Schwerpunkt auf die Definition eines Protokolls zwischen Orten³ legt, wird im MASIF-Standard der Versuch unternommen, eine Region in einer Schnittstelle nach außen abzubilden. Weiterhin werden im FIPA-Standard weitreichende Annahmen über den inneren Aufbau eines Agentensystems gemacht. So muss ein FIPA-konformes Agentensystem für jeden Agenten eine eindeutige Identität⁴ erzeugen. Eine derartige Vorgabe existiert nach MASIF nicht. Die Wahl der eindeutigen Benennung ist zunächst eine Frage der Eindeutigkeit innerhalb eines Agentensystems, d.h. verschiedene Agentensysteme können gleiche Bezeichnungen für ihre Agenten verwenden. Erst wenn es zu regionsüberschreitenden Kontakten zwischen Orten oder Agenten kommt, müssen die Bezeichnungen allgemein eindeutig sein. Der eventuell geringere Aufwand zur Erzeugung von Kennungen, die nur innerhalb einer Region eindeutig sind, wird so spätestens beim Design eines Kontaktpunktes zu anderen Regionen zu einem erhöhten Aufwand führen. Beim Entwurf eines Systems zur Vergabe weltweit eindeutiger Kennungen für Agenten sollte einerseits der Aufwand vertretbar sein und andererseits sollte die Aufgabe derart zufriedenstellend gelöst werden, dass bei der Weiterentwicklung einer Agentenumgebung keine unnötigen Beschränkungen vorliegen.

Auch in einem in sich geschlossenen Agentensystem muss das Problem der Zuweisung von eindeutigen Kennungen gelöst werden. Dabei ist jedoch eine deutlich geringere Komplexität anzusetzen. In einem System, das dem MASIF-Ansatz folgt, kann ein reduzierter Namensraum innerhalb einer Region verwendet werden. Der Kontaktpunkt der Region zur Außenwelt muss dann aber in der Lage sein, die internen Kennungen derart zu erweitern, dass die so konstruierte Bezeichnung global eindeutig ist. In jedem Fall muss ein Minimum an Systeminformationen über eine Region und die darin enthaltenen aktiven Agenten in der Schnittstelle einer Region abgebildet werden. Hat ein System dies realisiert und die Schnittstelle nach außen definiert, ist es prinzipiell möglich, von einem zweiten Agentensystem aus auf die in der Schnittstelle abgebildeten Informationen zuzugreifen.

Agentensysteme arbeiten mit verschiedenen Sprachen und Architekturen und sind für unterschiedlichste Aufgaben konzipiert. Das macht, wie in den vorherigen Abschnitt erläutert, eine Kommunikation zwischen Agentensystemen und den Austausch von Diensten untereinander unmöglich. Um dieses grundlegende Problem zu lösen wird daher vorgeschlagen, CORBA aufgrund seiner Universalität als Basis für die Kommunikation zwischen Agentensystemen einsetzen. Statt jedes einzelne Agentensystem mit einem spezifischen Zugangspunkten für ein oder mehrere andere Agentensysteme oder statische Anwendungen auszustatten (MASIF), wird in dem konzipierten und entworfenen Kommunikationskon-

³werden innerhalb des FIPA-Standards als *agent platform* bezeichnet

⁴wird im FIPA-Standard *Global Unique Identifier* (GUID) genannt

zept genau einmal eine definierte Schnittstelle realisiert. Dies läßt sich durch eine für alle Systeme verbindliche Schnittstellenbeschreibung in IDL erreichen. Der Funktionsumfang eines Systems wird dabei nicht verändert. Je nachdem, ob ein Zugriff von außen erfolgen soll oder eigene Elemente angeboten werden sollen, wird jeweils eine Schnittstelle in die bestehende Architektur integriert.

Der MASIF-Standard schlägt die Implementierung einer Systemmanagementschnittstelle vor. In CORBA werden die Aufgaben für Client und Server eindeutig zugewiesen. Um CORBA nun für die Kommunikation einsetzen zu können, bedeutet dies für den Entwurf einer Systemmanagementschnittstelle, dass eine vollständige Implementierung beide Teile vorzusehen hat. So wird einerseits ein Zugriff auf fremde Systemdaten ermöglicht und andererseits kann von außen auf die Informationen des eigenen Systems, wie verfügbare Agenten etc., MASIF-konform zugegriffen werden. Erhält ein Agentensystem so Kenntnis von den in einem fremden System realisierten Funktionen, so kann dies als eine rudimentäre Dienstanündigung interpretiert werden. Rudimentär deshalb, weil die Information über die Kennung eines Agenten, seinen Aufenthaltsort und seinen Zustand noch keinen Zugriff auf seine Funktionen erlaubt. Unabhängig von einer späteren Realisierung eines Systemgrenzen überschreitenden Objektzugriffes, ist der Austausch von Systemmanagementdaten der erste notwendige Schritt zur Erlangung von Interoperabilität zwischen heterogenen Agentensystemen.

4.5 Realisierung einer Systemkommunikation nach MASIF

Für die im Rahmen dieser Arbeit durchgeführte Umsetzung von MASIF-Konzepten in einem realen Szenario wurden die Agentenplattformen Voyager in der Version 2.0.0 und Grasshopper in der Version 1.2, sowie der CORBA ORB von IONA (OrbixWeb 3.1) ausgewählt. Dabei sollen agentenbasierte Dienste, die auf der Agentenplattform Grasshopper entwickelt wurden und dort verfügbar sind, von einem Voyager-Agentensystem aus erkannt und gestartet werden können. Nachfolgend wird beschrieben, welche Maßnahmen erforderlich sind, um ein entsprechendes Dienstangebotssystem mit zwei verschiedenen Agentensystemen zu realisieren.

4.5.1 Die Grasshopper-Region

In Abb. 4.6 ist die dafür verwendete Logik auf der Seite des Agentensystems Grasshopper in einem Flußdiagramm dargestellt.

Damit Grasshopper den MASIF-Standard unterstützt, müssen zunächst die entsprechenden CORBA und CfMAF-Klassen⁵ geladen werden. Außerdem sollte ein Naming Service im Netz verfügbar sein. Prinzipiell reicht dafür ein `orbxdj` Daemon-Prozeß⁶ auf einem beliebigen Netzknoten im Kommunikationsnetz. Beim Starten einer Region Registry oder einer Agentur (Kap. 3.4.2) im Grasshopper-System wird daher zuerst lokal geprüft, ob ein `orbxdj` auf der selben Maschine aktiviert wurde. Wird kein aktiver Prozess entdeckt, so wird ein UDP-Broadcast in das lokale Netz gesendet, auf den OrbixWeb Daemon-Prozesse antworten. Stehen gleich mehrere Dienste im gleichen Netz zur Verfügung, wird die erste Antwort verwertet und die folgenden verworfen.

⁵Die CfMAF-Klasse realisiert die IDL-Spezifikation des MASIF-Standards.

⁶Naming Service, der von OrbixWeb zur Verfügung gestellt wird.

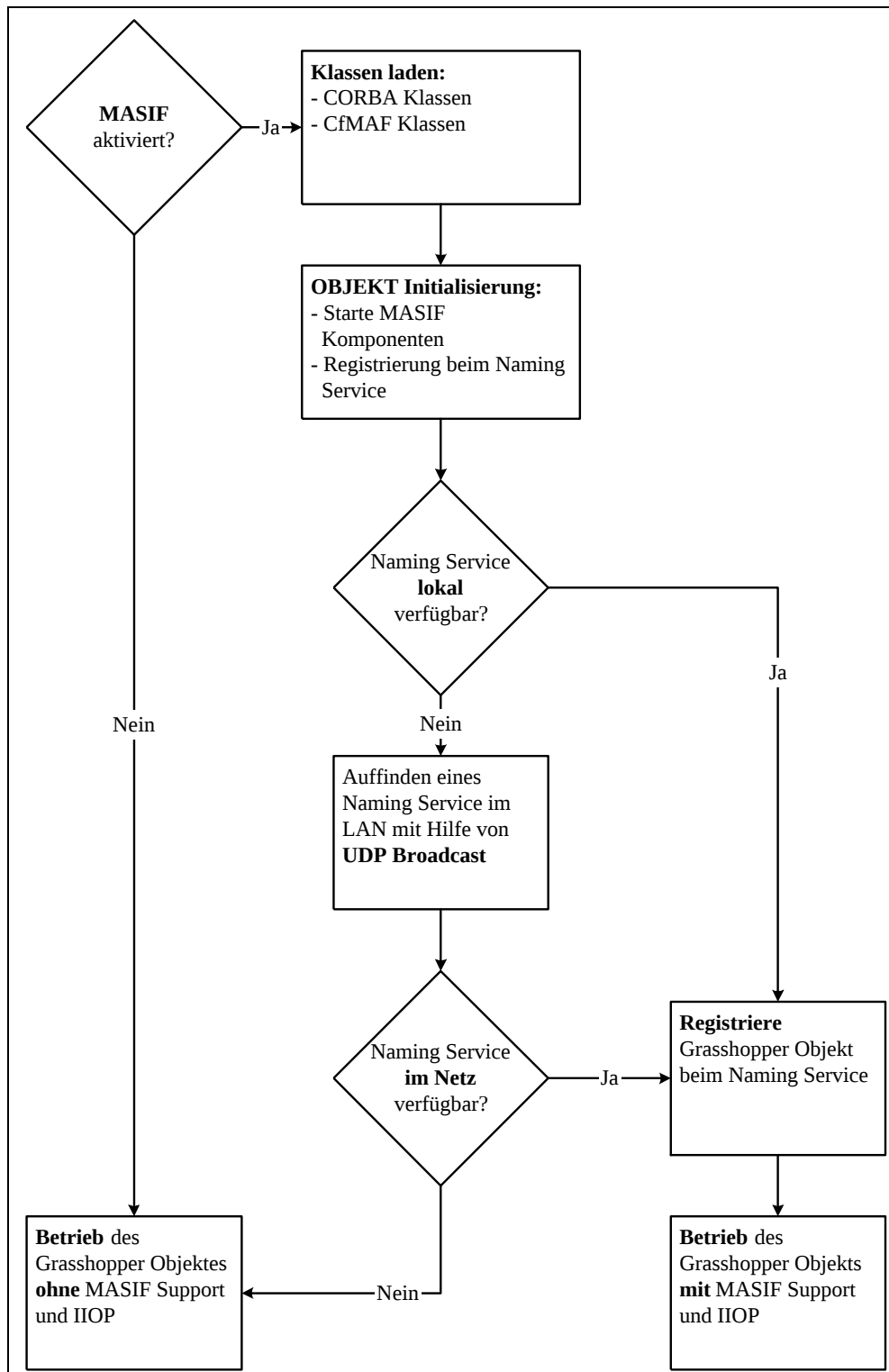


Abbildung 4.6: Einbinden des OrbixWeb Naming Service in Grasshopper

Agentensysteme können prinzipiell als Einzelsystem oder als Teil einer Region gestartet werden. Für den Start einer Grasshopper-Region muß zunächst die zentrale Verwaltung von Grasshopper, die Region Registry, gestartet werden. Wird ein Name für die Registry vorgegeben, dann wird dieser als Objektname im CORBA Naming Service eingetragen. Im vorliegenden Szenario wurde „**MCAP**“ als Objektschlüssel für die Region Registry gewählt. Weitere Parameter der Startdatei, die nach dem ersten Start erzeugt wird, sind die Port-Adressen der einzelnen Protokolle und der Einsatz der Sicherheitsdienste von Grasshopper:

```
Auszug aus der Datei "MCAP.INI"
...
[communication]
servers=iiop:1571, socket:7000, rmi:12000
useSecurity=false
...
```

Um eine Grasshopper Agentur an die Region Registry anzubinden, muß der Agentur beim Start das Protokoll, die IP-Adresse, der Kommunikationsport und der Name der Region Registry übergeben werden:

```
iiop://164.48.194.149:1571/MCAP
```

Beim Starten versucht sich das Grasshopper System mit Hilfe des vorgegebenen Protokolls bei einer Region Registry einzutragen. Schlägt dieser Versuch fehl, startet das System ohne zentrale Anbindung. Das Starten, Beenden und Verschieben von MAs ist nach wie vor allerdings auch ohne zentrale Anbindung möglich. Es stehen dann jedoch keine Such- und Verwaltungsfunktionen zur Verfügung.

Der Einsatz von CORBA erfordert eine klare Einteilung in Server- und Client-Komponenten. Die MASIF-Integration von Grasshopper erlaubt es, Grasshopper als Client oder als Server in Verbindung mit anderen Systemen einzusetzen. Bei der realisierten Systemkommunikation wird Grasshopper als Server betrieben, was bedeutet, dass Systeminformationen aus der Grasshopper-Umgebung vom Agentensystem Voyager aus gelesen werden können. Die Aufgabe, einen MAFFinder zu implementieren, ist nicht trivial, da dem MASIF-Standard kaum Vorgaben zur Realisierung zu entnehmen sind. Die realisierte heterogene, verteilte Umgebung für MAs kann jedoch ohne Einschränkung für die Bewertung der Praxistauglichkeit der MASIF-Konzepte herangezogen werden.

In dem Versuchsaufbau werden zwei logisch und konzeptionell getrennte Regionen (eine Grasshopper- und eine Voyager-Region) parallel zueinander betrieben. Die Voyager-Region stellt dabei einen MAF-Client zur Verfügung. Abb. 4.7 zeigt die Grasshopper-Region, die aus drei Agenturen besteht.

Dabei ist die Agentur mit der Heimat, *InformationDesk*, gleichzeitig auch der Zugangspunkt der Region. Festgelegt ist der Einsatz von IIOP zum Zugriff auf die MAFAgent-System-Schnittstelle. Da das Ergebnis der Methode `get_MAFFinder()` eine Referenz auf Routinen in der Region Registry zurückgibt, muss die Grasshopper Region Registry auch IIOP unterstützen. Die Registrierung des Zugangspunktes bei der Region Registry erfolgt ebenfalls unter Verwendung von IIOP. Dadurch ist u.a. das Protokoll zwischen der Agentur des Zugangspunktes und der Region Registry festgelegt. Der Zugangspunkt ist damit nichts anderes als eine normale Agentur. Für die Kommunikation zwischen verschiedenen Orten der Grasshopper-Region kann weiterhin das HTTP-Protokoll verwendet werden.

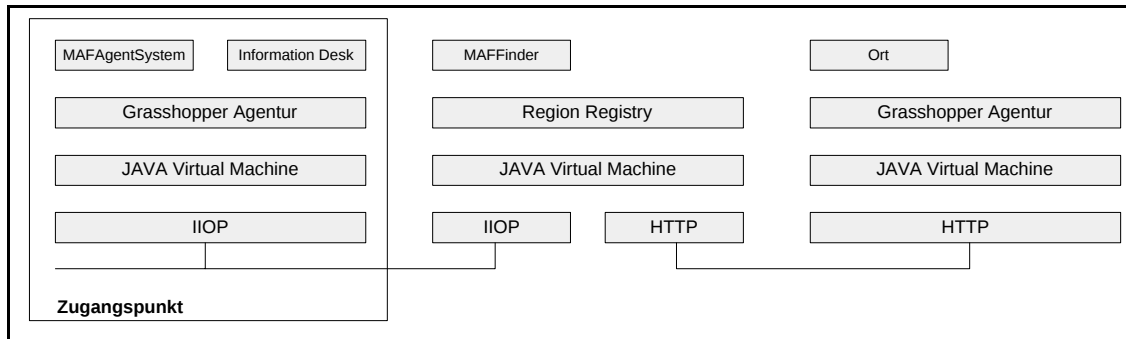


Abbildung 4.7: Grasshopper-Region mit MASIF-Schnittstellen

4.5.2 Die Voyager-Region

Innerhalb des Voyager-Agentensystems ist es möglich, Orte, also Laufzeitumgebungen für Agenten, an verschiedenen Stellen im Netz einzurichten. Einerseits kann so eine Lastverteilung, andererseits aber auch eine flexible Zuordnung von Orten im Netz zu Funktionen bzw. Diensten erfolgen. Denn für Voyager-Agenten ist es unerheblich, wo ein Dienst gestartet wird, da zunächst bei jeder Dienstanfrage geklärt wird, wo der Dienst verfügbar ist. Ob ein oder mehrere Orte an einem Kontakt zu einer anderen Region beteiligt sind, ist ebenfalls unerheblich. Für die Kommunikation als MAF-Client (Abb. 4.3) mit einem anderen MASIF-konformen Agentensystem ist neben einem Naming Service ein MASIF-Port erforderlich. Der Zugriff auf die MASIF-Schnittstellen ist für Agenten innerhalb von Voyager transparent. Die Objektreferenzen, die z.B. Grasshopper intern verwendet, werden über die CORBA-Schnittstellen an den MASIF-Port vermittelt. Innerhalb der Voyager-Umgebung erfolgt die Aktivierung der entfernten Methoden in Grasshopper im MASIF-Port. Den generellen Aufbau der Voyager-Region kann man Abb. 4.8 entnehmen.

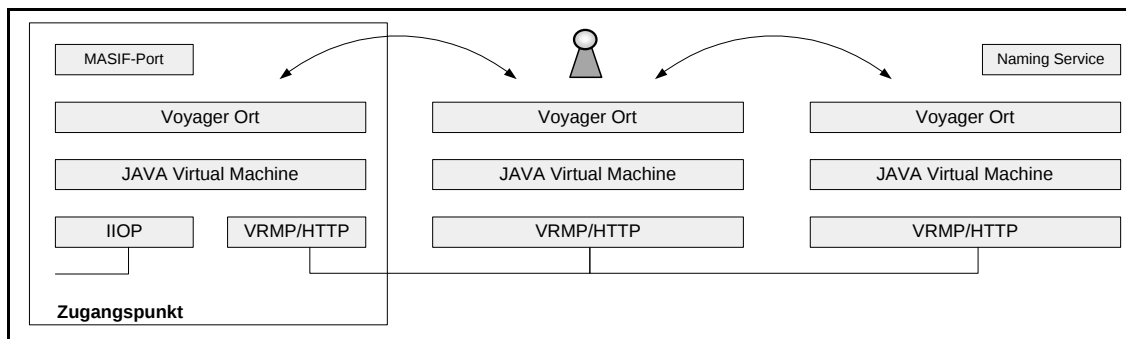


Abbildung 4.8: Voyager-Region mit MASIF-Port

Für die Systemkommunikation zwischen den Agentensystemen Voyager und Grasshopper nach MASIF wurde eine Dienstankündigung realisiert. Die Aufgabe des MASIF-Port ist es dabei, in kurzen Zeitabständen nach einem Naming Service zu suchen und eine Liste von verfügbaren MAFFinder- und MAFAgentSystem-Schnittstellen zu führen. Der MASIF-Port ist also ein Dienst, der einen Teil der Aufgaben des Naming Service übernimmt. Das Voyager-System sieht für alle wesentlichen Systemkomponenten Einträge im Naming Service vor. Der erste Schritt besteht folglich aus dem Eintrag des MASIF-Ports in der Datenbank des Naming Services. Dieser Eintrag beinhaltet den Namen und den Ort, an

dem der Dienst verfügbar ist. Während der MASIF-Port jedoch ein Systemdienst ist, der die Funktionalität des Voyager Systems erweitert, können durch die Referenzen zu den Speziellen Dienstagenten (SDA) des fremden Agentensystems nur Verbindungen zu externen Diensten hergestellt werden, d.h. der MASIF-Port vermittelt lediglich Referenzen auf externe Agentensysteme und Orte. Detailkenntnisse über die externen Dienste müssen sich die Agenten aktiv durch Nachfragen erwerben. Der Startprozess des MASIF-Ports ist in einem Flußdiagramm in Abb. 4.9 dargestellt.

Die Ortsliste des Naming Service wird in kurzen Zeitabständen durch einen mobilen Agenten um die Einträge aus dem MAFFinder ergänzt. Die Liste aller existierender Agenten, d.h. aller Management- und Dienstagenten, wird im Naming Service von Voyager jedoch nicht automatisch aktualisiert. Eine exakte Synchronisation beider Systeme erscheint nicht sinnvoll, da dies einen zu großen Eingriff in die Strukturen der Systeme erfordert. Der MASIF-Port stellt somit nur Kontaktpunkte, d.h. spezielle Dienste, außerhalb von Voyager in entsprechenden Listen dar und gibt diese auf Nachfrage an stationäre oder mobile Agenten weiter. Der Naming Service von Voyager erhält so Kenntnis von den Namen und Adressen von Orten außerhalb der eigenen Region. Die Verantwortung für die Konsistenz dieser Listen liegt beim MASIF-Port und dem externen Naming Service. Als Aktualisierungsfrequenz für die Daten hat sich in der realisierten Lösung eine Frequenz von 60 Hz als geeignet erwiesen, um einerseits eine aktuelle Ortsliste zu garantieren und andererseits die Leistungsfähigkeit der eigenen Region aufrechtzuerhalten.

Durch die Einführung des MASIF-Ports in eine Voyager-Agentenwelt ergänzt ein permanenter Ort die Systemdienste. Dieser realisiert die Logik zur Anbindung von externen Systemen und reduziert damit den Implementierungsaufwand auf der Seite von mobilen Objekten, die Agenten einer fernen Plattform kontaktieren möchten. Konkrete Aufträge an Softwareagenten lassen sich nur mit Kenntnissen über die Systemstruktur, in der sich der Agent bewegen wird, formulieren. Eine vollständige Transparenz zwischen verschiedenen Architekturen ist jedoch derzeit nicht mit überschaubarem Aufwand erreichbar.

4.6 Zusammenfassung

In diesem Kapitel wurden die neu entwickelten Konzepte beschrieben, die mit Hilfe des CORBA-Standards möglich sind, um heterogenen Kommunikationssystemen, wie den existierenden Agentensystemen, die Zusammenarbeit zu erlauben und es wurde erläutert, in welchen Bereichen dies mit Vorteil eingesetzt werden kann. Weiterhin wurde eine MASIF-konforme Systemkommunikation zwischen Grasshopper- und Voyager-Agentensystemen realisiert und vorgestellt. Dazu wurde mit dem MASIF-Port ein spezieller Voyager-Ort entworfen, der MASIF-konform Informationen vom Grasshopper-System erhält. Es hat sich gezeigt, dass so eine einfache Zusammenarbeit zwischen verschiedenen Agentensystemen derart erreicht wurde, dass Dienste eines fremden Systems angekündigt und bekannt gemacht werden können. Der Aufwand für darüber hinausgehende Funktionen, wie das Erwerben von Detailkenntnissen über Dienste oder existierende Agenten, ist jedoch so erheblich, dass eine solche Lösung nicht als sinnvoll erachtet wird.

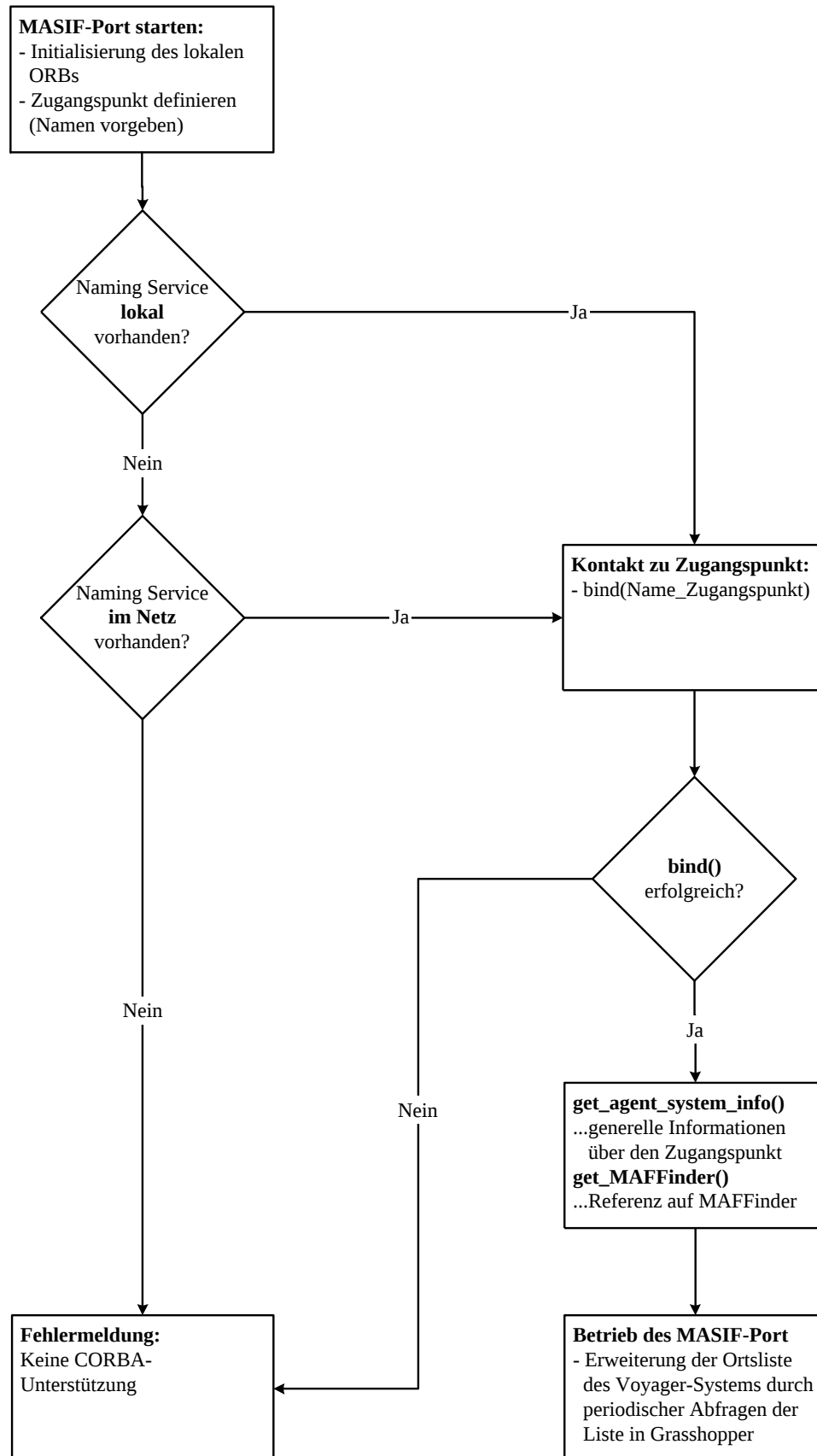


Abbildung 4.9: Startprozess des MASIF-Ports

Agentenbasiertes Kommunikationskonzept für M-Commerce-Dienste

In den vergangenen Jahren wurden mehrere agentenbasierte Kommunikationskonzepte für die breitbandige Internetwelt entworfen und implementiert [BHRS97, BGM⁺99, SM99]. In diesem Kapitel werden zunächst Anforderungen an eine intelligente, agentenbasierte Dienstleistung für Mobilfunknetze diskutiert, die bei der Konzeption und Implementierung der in dieser Arbeit entworfenen Kommunikationsplattform MCAP berücksichtigt worden sind. Anschließend wird die MCAP-Plattform vorgestellt und erläutert. Danach werden die Konzepte und die Implementierung zweier agentenbasierter Dienste, welche auf dieser Kommunikationsplattform basieren, vorgestellt. Beide untersuchten Dienste gehören zu dem vielversprechenden Bereich des sogenannten *M-Commerce*, dem mobilen elektronischen Handel (*mobile electronic commerce*), welchem Marktanalysten ein großes Potenzial vorhersagen [MV00]. In dieser Arbeit gilt die folgende Definition für M-Commerce:

M-Commerce (Mobile E-Commerce) ist eine Untermenge des E-Commerces und umfasst dabei alle elektronischen Geschäftsvorgänge zwischen Unternehmen und Endkunden (B2C¹) sowie zwischen verschiedenen Unternehmen (B2B²), bei denen sich die involvierten Parteien bei der Nutzung physikalisch bewegen können.

5.1 Anforderungen an eine intelligente Agentenmigration

Mobile Agenten (MAs) erlauben neue Möglichkeiten zur Strukturierung und Implementierung von offenen Verteilten Systemen (VS). Mit ihrer Fähigkeit, autonom und asynchron zu agieren, können sie die Nachteile von mobilen Kommunikationsnetzen, wie z.B. eingeschränkte Verfügbarkeit und geringe Datenraten, verringern. Die Standardisierungsgremien der OMG und FIPA haben sich in der Vergangenheit zwar mit dem Management von MAs beschäftigt, es mangelt jedoch an detaillierten Vorgaben für die Implementierung. Weiterhin fehlen Lösungen für den Einsatz von MAs in Kommunikationsnetzen mit sogenannten *nomadischen Benutzern* (nomadic users) mit mobilen Endgeräten [HK00].

Nomadische Benutzer sind Kommunikationsteilnehmer, die sich an wechselnden Orten aufhalten und häufig *offline* sind, d.h. zwischen ihren Verbindungen mit einem Kommunikationsnetz gibt es längere zeitliche Unterbrechungen.

Agentensysteme können über verschieden zuverlässige Netze miteinander verbunden sein. In diesem Abschnitt wird insbesondere die uneingeschränkte Kommunikation bzw. Agentenmigration über Zugangsnetze mit geringer Verfügbarkeit und hoher Wahrscheinlichkeit des Verbindungsabbruches untersucht. Es wird davon ausgegangen, dass die eingeschränkte Agentenmigration, d.h., dass nur bestimmte Orte von den Agenten aufgesucht werden dürfen, einen Spezialfall der uneingeschränkten Migration darstellt.

¹Business-to-Consumer

²Business-to-Business

5.1.1 Zugangsnetz

Zur Durchführung der Agentenmigration benötigt ein Agentensystem die Möglichkeit, eine Verbindung zu einem Kommunikationsnetz, dem Zugangsnetz (*access network*), aufzubauen. Bei der Betrachtung des Zugangsnetzes wird in dieser Arbeit zwischen Festnetz- und Mobilfunkzugängen unterschieden, da die Gegenwart von Mobilfunknetzen die Fehlereigenschaften und die Kommunikationszeit des Gesamtsystems signifikant verändern können. Neben der Art des Zugangsnetzes ist die Verfügbarkeit eines Kommunikationsnetzes von wichtiger Bedeutung. Kommerzielle Netzbetreiber bemühen sich naturgemäß um möglichst geringe Systemausfallzeiten und ausreichende Netzkapazitäten, während bei Privatnetzen der Netzzugang aus Wartungs- und Pflegegründen häufig eingeschränkt ist. Es ist daher sinnvoll, zwischen zwei verschiedenen Agentensystemtypen mit jeweils zwei Dienstgütern zu unterscheiden [PH00], siehe Abb. 5.1:

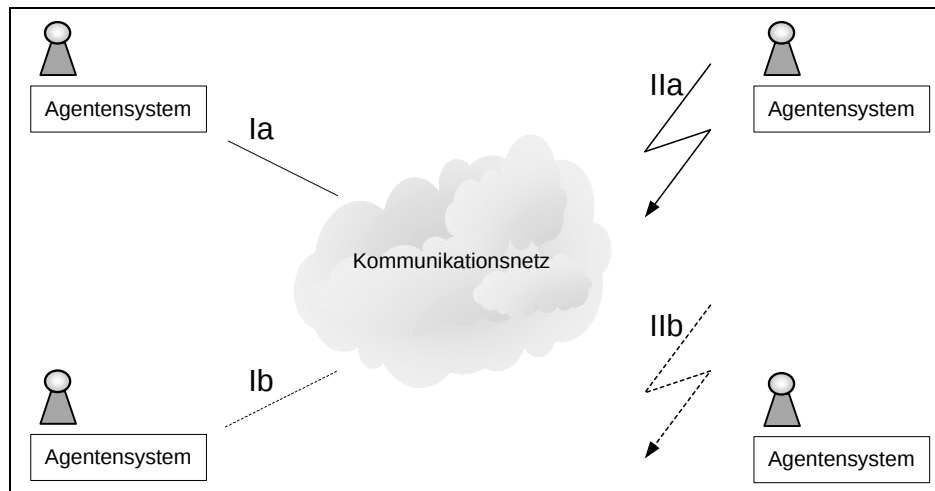


Abbildung 5.1: Agentensystemtypen

- | | |
|---------|---|
| Typ Ia | Agentensysteme, die an ein Telekommunikationsfestnetz mit hoher Verfügbarkeit, d.h. mit Ausfallzeiten, die geringer als 10 Sekunden im Jahr sind, angeschlossen sind (z.B. öffentliches Telefonnetz). Diese Systeme werden meist durch kommerzielle Dienstanbieter betrieben. |
| Typ Ib | Agentensysteme, die ebenfalls an ein Festnetz angeschlossen sind, allerdings im Hinblick auf die Verfügbarkeit nicht die Dienstgüte von Agentensystemen des Typs Ia erfüllen (z.B. Corporate Network). |
| Typ IIa | Agentensysteme, die an ein Funknetz angeschlossen sind, das Ausfallzeiten von weniger als 10 Sekunden im Jahr hat (z.B. öffentliches GSM-Netz). |
| Typ IIb | Agentensysteme, die ebenfalls an ein Funknetz angeschlossen sind, aber nicht die Dienstgüte von Typ IIa-Agentensystemen anbieten können (z.B. privates WLAN-Netz). |

In dieser Arbeit wird eine Lösung für die Dienstangebotung in Mobilfunknetzen nach dem GSM-Standards entworfen und untersucht. Daher werden nachfolgend hauptsächlich Typ

IIa-Agentensysteme betrachtet. Für diese Systeme ist die **Benutzermobilität** (*personal mobility*) und die **Endgerätemobilität** (*terminal mobility*) von besonderer Relevanz. Abb. 5.2 zeigt die Komponenten der totalen Mobilität.

Totale Mobilität oder **ideale Benutzermobilität** liegt vor, wenn ein Kommunikationsnetz sowohl Benutzermobilität als auch Endgerätemobilität unterstützt.

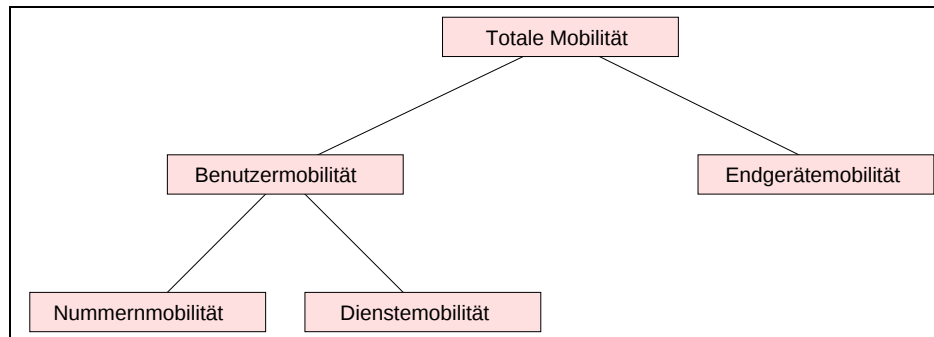


Abbildung 5.2: Stufen der Mobilität

5.1.2 Benutzermobilität

Funktionen zur Unterstützung von Benutzermobilität erlauben einem Benutzer auf seine Teledienste von verschiedenen Endgeräten (z.B. PCs, Laptops, Mobilfunkgeräten und Handcomputer) von unterschiedlichen Orten aus zuzugreifen. Das bedeutet, der Benutzer besitzt mehrere Dienstzugangspunkte (*service access points*) [Wal97] und ist netz- und endgeräteunabhängig. Anrufe und Nachrichten werden immer zu dem Endgerät weitergeleitet, welches er gerade benutzt [HK00]. Wie Abb. 5.2 zeigt, sind die Nummern- und Dienstmobilität zwei wichtige Eigenschaften der Benutzermobilität.

Nummernmobilität: Nummernmobilität (*number mobility*) erlaubt den Teilnehmern, Anrufe und Nachrichten unabhängig vom Aufenthaltsort mit derselben Telefonnummer bzw. Kommunikationsadresse zu senden und zu empfangen. Der Erfolg des Mobilfunkstandards GSM (Kap. 2.2.2) ist zweifelsohne eng verknüpft mit der Tatsache, dass das GSM-System Nummernmobilität unterstützt, d.h. ein GSM-Teilnehmer kann unter seiner persönlichen Rufnummer auch in fremden Kommunikationsnetzen Anrufe und Nachrichten empfangen. Die Abrechnung über die genutzten Dienste erfolgt über den Dienstanbieter.

Dienstmobilität: Dienstmobilität³ (*service mobility*) erlaubt dem Teilnehmer den uneingeschränkten Zugriff auf seine Teledienste, unabhängig vom Endgerät und Kommunikationsnetz, auf das er gerade zugreift. Das VHE-Konzept (Kap. 2.2.4) soll ermöglichen, dass Teilnehmern in zukünftigen Kommunikationsnetzen Dienstmobilität zur Verfügung gestellt wird.

Die Benutzermobilität hat einen wesentlichen Einfluss auf die Konzeption von Agentensystemen, da der Teilnehmer davon ausgeht, dass ein MA, den er gestartet hat, nach getätigter Arbeit zu dem Ort, an dem sich der Teilnehmer dann gerade befindet, und auf

³wird auch Dienstportabilität genannt

das Endgerät, welches der Teilnehmer gerade benutzt, zurückkehren kann. Drei wichtige Anforderungen leiten sich daher für ein Agentensystem aus der Benutzermobilität ab [HP00]:

1. Das dienst anbietende Agentensystem benötigt aktuelle Ortsinformationen über das vom Teilnehmer benutzte Agentensystem, so dass der MA sein Migrationsziel und die entsprechende Migrationsroute ausreichend gut kennt.
2. Im Vorfeld der Agentenmigration werden Angaben über die verfügbaren Dienste benötigt, da die Möglichkeit besteht, dass das gerade benutzte Agentensystem einen speziellen Agententyp (z.B. aus Lizenzgründen) nicht unterstützt.
3. Vor der Migration sollte auch abgeklärt werden, welche Systemressourcen der Agent benötigt, um das Auftreten von Fehlern oder gar den Zusammenbruch des benutzten Agentensystems zu vermeiden.

5.1.3 Endgerätemobilität

Endgerätemobilität erlaubt einem Benutzer, dass er sich während der Nutzung von Kommunikationsdiensten physikalisch bewegen kann. Dies erfordert in der Regel, dass der Teilnehmer ein mobiles Endgerät innerhalb des Funkversorgungsgebietes mit sich führt [Gun96]. Agentensysteme mit Funkschnittstelle, d.h. Systeme vom Typ II, stellen die Entwickler vor neue Herausforderungen. Aus der Endgerätemobilität ergeben sich die folgenden Anforderungen an diese Agentensysteme [PH00]:

1. Sollte der Transport eines MAs zu einem entfernten Ort scheitern, so sollte der Grund für dieses Scheitern, z.B. Endgerät ist abgeschaltet, heruntergefahren, existiert nicht oder das Netz ist nicht verfügbar, bekannt sein, damit das Agentensystem entsprechend darauf reagieren kann. Grundsätzlich gibt es dabei zwei Reaktionsmöglichkeiten: der MA entscheidet, nicht zu migrieren oder er wartet, bis das Migrationsziel wieder verfügbar ist.
2. Für MAs, deren Migration fehlgeschlagen ist, sollten die Agentensysteme einen Dienst anbieten, der den MAs mitteilt, wann das System wieder verfügbar und außerdem in der Lage ist, MAs für eine bestimmte Zeit zwischenspeichern.
3. TypII-Agentensysteme sollten selbständig die Verbindung abbauen können, wenn diese nicht mehr benötigt wird.
4. Die Agentensysteme benötigen einen Mechanismus, der eine Kopie eines verlorengegangenen Agenten erstellen kann. Dabei muss sicher gestellt werden, dass der verlorene MA zerstört wird.
5. Bereits vor der Agentenmigration sollte die ungefähre Migrationsdauer bekannt sein, um eine netzzustandsabhängige Migration zu erlauben. Die Notwendigkeit für diese Anforderung wird im folgenden Abschnitt motiviert und erläutert.

5.1.4 Netzzustandsabhängige Agentenmigration

MAs können den Nachrichtenaustausch über die Luftschnittstelle verringern. Anstatt einen neuen, komplexen Dienst durch eine Reihe von Nachrichten anzufordern und zu übertragen,

muss lediglich ein Dienstagent (DA) zum Endgerät transportiert werden. Dies ermöglicht den universellen Gebrauch von Diensten, unabhängig vom Netzzugang und vom benutzten Betriebssystem. Eine verkehrsabhängige Agentenmigration erscheint insbesondere dann geeignet zu sein, wenn der MA die Wahl zwischen mehreren Zielsystemen hat, die durch unterschiedliche Kommunikationsnetze miteinander verbunden sind. Da bestimmte Teile eines Kommunikationsnetzes überlastet sein können, hängt die Dienstgüte nicht nur von der Leistungsfähigkeit des Dienstes selber, sondern auch von den Zugangszeiten zum Dienst ab.

Es erscheint daher sinnvoll, dass der MA die zu erwartenden Transferzeiten im Vorfeld der Migration abfragt. Anhand dieser Information kann der MA bzw. das sendende Agentensystem beurteilen, ob eine Agentenmigration zweckmäßig ist und welche Migrationsstrategie angewendet werden soll [BER01]. Weiterhin wird festgestellt, ob das aufzusuchende Agentensystem überhaupt verfügbar ist. Mit Hilfe dieser Information kann der MA bzw. das System ebenfalls auf wechselnde Rahmenbedingungen reagieren. Betrachtet man einen Agenten, der mehrere Stellen (z.B. Händler bei einem E-Commerce-Dienst) aufsuchen möchte, so wird ein proaktiver Agent (Kap. 3.2.2) zunächst die Agentensysteme aufsuchen, die mit geringen Transferzeiten erreichbar sind, um damit den Transfer zu langsamen Systemen ganz zu vermeiden oder zumindest mit der Möglichkeit, dass sich die Zugangsbedingungen zu den anderen Systemen in der Zwischenzeit verbessern. Andere Agenten werden in der Lage sein, die Datenmenge (Selektion) oder das Datenformat (Konvertierung), welches sie mit sich tragen, an die verfügbare Übertragungskapazität anzupassen. Kann ein MA die Übertragungszeiten bereits vor dem Transfer abschätzen, wird er seine Aufgaben u.U. so organisieren, dass sie besonders schnell ausgeführt werden können. Dadurch kann auch eine Netzüberlastung vermieden werden, d.h. eine netzzustandsabhängige Agentenmigration verbessert die Dienstgüte und sorgt für eine bessere Verteilung der Netzlast zwischen den bei der Kommunikation beteiligten Netzknoten.

Die Vorteile der netzzustandsabhängigen Agentenmigration hängen allerdings stark von der Güte der Verkehrsvorhersage ab. Da die Vorhersage zusätzlichen Signalisierungsaufwand bedeutet, kann eine schlechte Schätzung sogar die Leistungsfähigkeit von Agenten, Diensten und Kommunikationsnetzen vermindern. Es bieten sich mehrere Mechanismen für die netzzustandsabhängige Agentenmigration an:

1. **Stichprobe:** Mit Hilfe einer Testnachricht wie z.B. einem Standard-Ping kann festgestellt werden, ob ein Netzknoten im verteilten System physikalisch angebunden ist und welche Umlaufzeit (*round-trip time*) für ein Datenpaket benötigt wird. Ein solcher Mechanismus ist in [GKN⁺96] für das Agentensystem *Agent TCL* beschrieben worden. Der Nachteil dieser Lösung ist, dass die Stichproben regelmäßig durchgeführt werden müssen und damit eine nicht unerhebliche Netzlast darstellen. Ferner erhöhen die Stichproben die Kommunikationskosten.
2. **Vorhersage:** Dieser Mechanismus beruht auf der Annahme, dass die angeschlossenen Agentensysteme regelmäßig Informationen über ihre Auslastung und den Zustand ihrer Kommunikationskanäle einer zentralen Instanz, dem Netzzustandsinformationsdienst (NZID), mitteilen. Diese Lösung ähnelt dem SPAND-Modell [SSK97], in dem die in einem Kommunikationsnetz angeschlossenen Einheiten Leistungsberichte (*performance reports*) zu Leistungsknoten (*performance servers*) senden. Um für einen MA eine geeignete Vorhersage tätigen zu können, benötigt das sendende Agentensystem die folgenden Angaben:
 - Netzleistungsdaten, d.h. Durchsatz, Umlaufzeit, Paketverlustrate etc.

- die Größe des zu übertragenden mobilen Agenten

Während das Agentensystem die Größe des MA selbst berechnen muss, kann es die restlichen Informationen vom NZID erhalten. Die Kommunikationskosten werden durch den zusätzlichen Informationsaustausch ebenfalls erhöht.

5.1.5 Paketorientierte Trägerdienste

Die Dienstangebot nach dem Agentenparadigma erlaubt eine flexible und kontextabhängige Verarbeitung von digitalen Daten. Doch insbesondere bei der Übertragung von Daten gibt es noch einen relativ großen Spielraum für Optimierungen. Besonders vorteilhaft für agentenbasierte Dienste wird sich die Einführung paketorientierter Mobilfunksysteme wie GPRS (Kap. 2.3) und UMTS auswirken, denn bei Anbietung von agentenbasierten Diensten in kanalorientierten GSM-Mobilfunknetzen (Kap. 2.2.2) wird eine erhebliche Zeit für den Aufbau eines Kommunikationskanals benötigt, wodurch sich die Dauer für die Agentenmigration wesentlich verlängert.

Für den Erfolg von zukünftigen mobilen Datenanwendungen, wie z.B. M-Commerce-Diensten, erscheint es notwendig, dass der Teilnehmer die Dienste unabhängig von seinem momentanen Aufenthaltsort und weiteren speziellen Anforderungen ohne erhebliche Verzögerungen nutzen kann. Paketorientierte Funkdatennetze, die dies ermöglichen, wurden bereits seit Ende der 80er Jahre installiert. Doch Netze wie MOBITECH [Wal00] und TETRA⁴ [Sch00] konnten hierfür nicht genutzt werden, da diese Netze lediglich eine sehr eingeschränkte Funkabdeckung anbieten. Die öffentlichen Mobilfunknetze GPRS und UMTS hingegen werden die Einführung von neuen Anwendungen mit globaler Funkabdeckung für geringe Gebühren erlauben. Der Einsatz von MAs erscheint in paketorientierten Netzen wie GPRS generell vorteilhaft, da im Vergleich zu RPC-Mechanismen die Laufzeitverzögerungen des Systems aufgrund der verhältnismässig großen Datenpakete nicht besonders ins Gewicht fallen.

Allerdings ist die Leistungsfähigkeit von GPRS nur schwer voraussagbar. Generell sind zwar hohe Datenraten erzielbar, wenn diese jedoch von mehreren Teilnehmern gleichzeitig genutzt werden, so verbleiben für jeden Teilnehmer nur Datenraten von weniger als 10 kbit/s. Verglichen mit einer LAN-Internetverbindung muss bei der GPRS-Kommunikation berücksichtigt werden, dass diese

- hohe und variierende Laufzeitverzögerungen aufweist (*latency*),
- lediglich eine geringe und variierende Bandbreite anbietet (*bandwidth*),
- und dass Verbindungsunterbrechungen auftreten können (*interruptions*).

Beim Entwurf von M-Commerce-Diensten sollten diese Rahmenbedingungen ausreichend berücksichtigt werden. Aufgrund der Tatsache, dass auch in zukünftigen Mobilfunksystemen die Datenkapazität eine begrenzte Ressource darstellt, wurde bei der nachfolgenden Konzeption der Teledienste auf redundante Kommunikation verzichtet. Ferner werden - auch aus Gründen der Kostentransparenz - die Dienste in der Regel vom Teilnehmer selbst initiiert.

⁴Das Bündelfunksystem TETRA (Terrestrial Trunked Radio, früher Trans European Trunked Radio) wurde 1991 von ETSI entwickelt, um nationale Bündelfunknetze wie MOBITECH zu ersetzen.

5.2 Agentenplattform für M-Commerce-Dienste

Ausgehend von den im Kap. 5.1 beschriebenen Anforderungen an eine flexible agentenbasierte Dienstleistung entstand die nachfolgende Konzeption und Implementierung einer Kommunikationsplattform, welche insbesondere für die Ausführung von M-Commerce-Diensten geeignet ist.

5.2.1 Konzept

Mittels der M-Commerce Agentenplattform (MCAP) sollen flexibel neue Dienste im Bereich des M-Commerces für Benutzer von zellularen Mobilfunknetzen ermöglicht werden. Die entworfenen Dienste sollen dabei dem Agentenparadigma entsprechen, eine Personalisierung erlauben und für nomadische Benutzer geeignet sein.

Der **MCAP** ist ein Protokollumsetzer (*gateway*), der sowohl die für die Übertragung und Bearbeitung von MAs verwendeten Protokolle, als auch andere gängige Kommunikationsprotokolle unterstützt.

Den generellen Entwurf dieses Gateways zeigt Abb. 5.3. Der MCAP ist für die Personalisierung und endgerätespezifische Anpassung von Dienstinhalten verantwortlich. Dafür vermittelt der MCAP Dienstanfragen zwischen agentenbasierten Netzknoten (z.B. mobiles Endgerät, Händlerrechner) und Dienstknoten (z.B. Bankserver oder WAP-Gateway). Dabei ist der MCAP derart konzipiert, dass er unabhängig von bestimmten Trägerdiensten agieren kann. Für die Kommunikation mit mobilen Endgeräten nach dem GSM-Standard unterstützt der MCAP leitungsorientierte (CSD) und paketorientierte (GPRS) Trägerdienste. Der Einsatz von MAs kann sowohl über die GSM-Luftschnittstelle, als auch über das Internet erfolgen. Dadurch werden die Einsatz- und Bewertungsmöglichkeiten der agentenbasierten Dienstleistung verbessert.

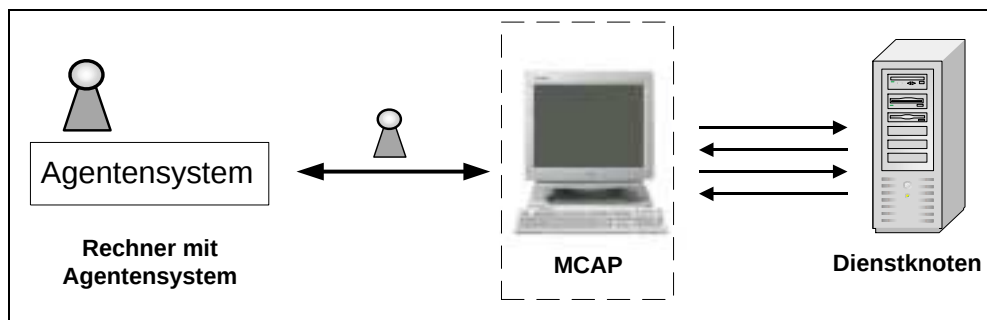


Abbildung 5.3: Funktion des MCAPs

5.2.2 Sicherheit

Für den MCAP wurde eine Implementierung der Version 3.0 von SSL der Technischen Universität Graz eingesetzt [Gra01]. SSL bietet sichere Verbindungen zwischen zwei Kommunikationspartnern und baut auf ein beliebiges Transportprotokoll auf. Gleichzeitig agiert SSL transparent für die Anwendungsschicht, so lässt sich SSL beispielsweise zwischen den HTTP- und dem TCP-Protokoll verwenden, ohne dass eines der beiden Protokolle geändert werden muss. Trotzdem sind die durch HTTP übertragenen Daten bis zur Transportschicht (ISO/OSI-Schicht 4) gesichert.

Die folgenden Eigenschaften werden von SSL unterstützt:

- **Authentifizierung:** SSL unterstützt den sicheren Austausch von digitalen Zertifikaten zur Authentifizierung. Tauschen beide Kommunikationspartner Zertifikate aus, so können sie sich wechselseitig authentifizieren.
- **Verschlüsselung:** Über SSL verschickte Daten werden mit einem MAC-Code versehen, womit ihre Authentizität sichergestellt wird, und dann verschlüsselt übertragen. Mit Hilfe der Verschlüsselung wird auch die Datenintegrität und die Vertraulichkeit (Kap. 3.2.6) erreicht.

Die konkreten Parameter einer SSL-Verbindung, wie Schlüssellängen und die zu verwendenden Verschlüsselungsalgorithmen, können individuell zwischen den beiden Kommunikationspartnern vereinbart werden. Zusammenfassend kann man sagen, dass SSL auf einfachem Weg die Möglichkeit bietet, bestehende TCP/IP-basierende Protokolle zu schützen. Somit lassen sich auch die im MCAP verwendeten Kommunikationskanäle mit SSL gegen Angriffe von außen zusätzlich sichern. Ferner werden Kommunikationspartner und ausgetauschte Daten authentifiziert.

5.2.3 Architektur

Der MCAP soll in der Lage sein, MAs zu starten, zu empfangen und zu verarbeiten. Der Zugriff auf die mittels MAs realisierten Dienste soll von verschiedenen Mobilfunknetze aus möglich sein. Der realisierte Prototyp beherbergt hierfür ein herkömmliches Agentensystem. Dies führt dazu, dass der MCAP mit Netzknoten, die ebenfalls dieses Agentensystem installiert haben, Agenten austauschen kann. Agentensysteme können einen oder mehrere Orte der gleichen Agentensystemtechnologie verwalten. Um aber auch die verfügbaren Dienste/Agenten von Orten anderer Agentensysteme nutzen zu können, sollte der MCAP auch auf einem Ort des Agentensystems einen MASIF-Port (Kap. 4.5) betreiben. Generell unterstützt der MCAP neben der Agentensystem-spezifischen Anwendungsschnittstelle (Agenten API) verschiedene Anwendungsprotokolle, d.h. er besitzt entsprechende Software-Schnittstellen (z.B. XML, SSL, MASIF). Außerdem beherbergt der MCAP unterschiedliche Serverkomponenten (z.B. WAP-Gateway), siehe Abb. 5.4.

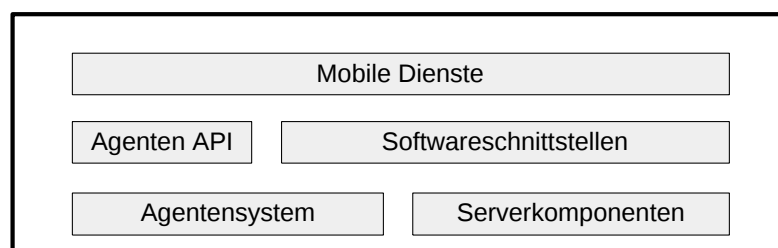


Abbildung 5.4: Architektur des MCAPs

Für M-Commerce-Dienste ist oftmals eine Kommunikation mit externen SQL-Datenbanken (Kap. 3.3.5) notwendig. Diese Datenbanken haben häufig einen breitbandigen Zugang. Die Interaktion mit den Datenbanken sollte daher mit MAs realisiert sein, die dann lediglich die gefilterten Ergebnisse weiterleiten. Für diese Weiterleitung bietet sich u.a. der WAP-Standard (Kap. 2.2.5) an, da WAP speziell für die Anforderungen mobiler Kommunikationsnetze der zweiten und dritten Generation konzipiert worden ist. Für die Kommunikation mit SQL-Datenbanken und WAP-Servern ist die Bereitstellung entsprechender

Schnittstellen ausreichend. Um erhebliche Verzögerungen, bedingt durch die Kommunikation mit gegebenenfalls leistungsschwachen, externen Komponenten zu vermindern, werden jedoch auf der Kommunikationsplattform MCAP nicht nur die Schnittstellen, sondern auch die entsprechenden Funktionen durch eine SQL-Datenbank (mSQL) und einen Web/WAP-Server (Apache) zur Verfügung gestellt. Darüberhinaus wurde auch ein WAP-Gateway (Kap. 2.2.5) im MCAP integriert, so dass sich der Teilnehmer im Falle einer WAP-Kommunikation mit seinem WAP-fähigen mobilen Endgerät direkt auf den MCAP via GSM oder GPRS einwählen kann.

Abb. 5.5 zeigt den realisierten Aufbau des MCAPs sowie die unterstützten Kommunikationsprotokolle der ISO/OSI-Schicht 1-4. Es fällt auf, dass die gesamte Kommunikation mit externen Komponenten auf dem IP-Protokoll beruht. Als Agentensystem wird Voyager in der Version 3.2 eingesetzt (Kap. 3.4.1). Neben den Serverkomponenten SQL-Datenbank, Web/WAP-Server und WAP-Gateway wird von der agentenbasierten Kommunikationsplattform MCAP das HBCI-Protokoll (Kap. 5.3), das SSL-Protokoll (Kap. 5.2.2), das XML-Syntax (Kap. 3.3.5) und der MASIF-Port unterstützt. Mit Hilfe von XML können MAs derart entworfen werden, dass die Anfragen der MAs bei den besuchten Agentensystemen leicht in andere Protokolle transformiert werden können. Der MASIF-Port des MCAPs erlaubt eine MASIF-konforme Kommunikation über einen CORBA ORB mit dem Agentensystem Grasshopper (Kap. 3.4.2), um dort vorhandene Dienste dem Teilnehmer bekanntzumachen und vom MCAP aus starten zu können.

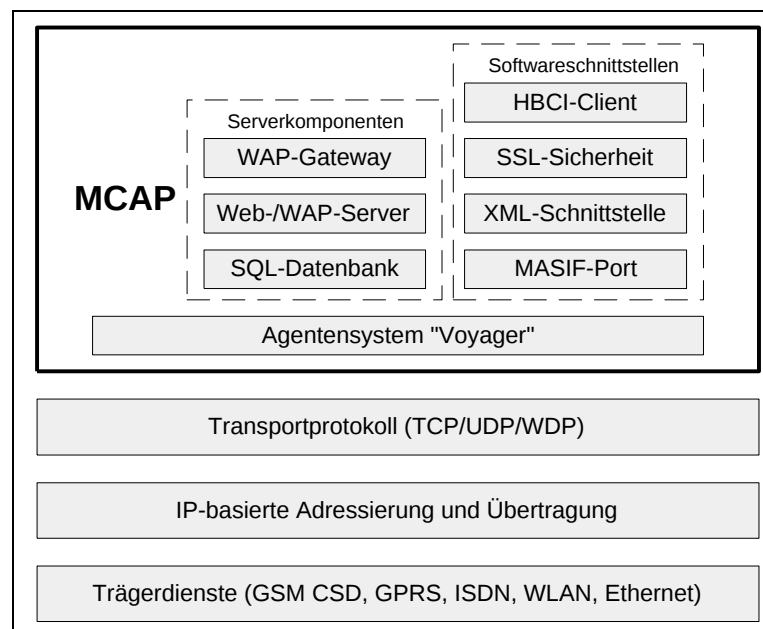


Abbildung 5.5: Aufbau des MCAPs

Der MCAP kann entweder professionell von einem Netzbetreiber bzw. von einem Drittanbieter (*third party provider*) betrieben oder als persönlicher Kommunikationsmanager in den Räumen des Teilnehmers installiert werden. Im letzten Fall ist es jedoch aufgrund der zahlreichen Kommunikationsverbindungen zu externen Agentensystemen und Dienstknoten aus Kostengründen wünschenswert, dass eine breitbandige und kostengünstige Kom-

munikationsverbindung zum Internet, z.B. durch eine sogenannte *flat rate*⁵, existiert.

Bei dem entwickelten MCAP besteht die generelle Verfahrensweise darin, dass ein Mobilfunkteilnehmer dem MCAP seine persönlichen Daten, z.B. Name, Anschrift, Präferenzen, Passwörter etc. übermittelt. Sollte der MCAP professionell betrieben werden, so können eine Reihe der personenbezogenen Daten mit Hilfe der Rufnummer (CLIP⁶) aus dem Mobilfunkvertrag des Teilnehmers mit dem Netzbetreiber entnommen werden, d.h. ein Datenaustausch zwischen HLR und MCAP wäre sinnvoll. Da ein GSM-Mobilfunknetz als ein geschlossenes Kommunikationsnetz betrachtet wird, dient die Rufnummer des Teilnehmers bei der Kommunikation mit dem MCAP stets auch zum Nachweis von dessen Authentizität.

5.2.4 Kommunikationsfluss

In der Regel wird der Mobilfunkteilnehmer die Kommunikation mit dem MCAP selbst initiieren (*mobile originated*), d.h. der Teilnehmer sendet mobile Agenten und WAP-Nachrichten zum MCAP. Danach besteht für den Teilnehmer die Möglichkeit, dass die Kommunikationsverbindung aufrecht erhalten wird, bis der MCAP eine Antwort schickt oder die Verbindung abubrechen. Aus Kostengründen erscheint das Aufrechterhalten der Verbindung lediglich dann sinnvoll, wenn mit einer zügigen Antwort gerechnet werden kann oder wenn paketerorientierte Trägerdienste eingesetzt werden, deren Abrechnung (hauptsächlich) auf der Datenmenge und nicht auf der Kommunikationszeit basieren. Im Falle des sofortigen Verbindungsabbruch nach der Übermittlung der Anfrage kann der Teilnehmer (abhängig von seinem Benutzerprofil und den Anwendungsparametern) die Antwort beim nächsten von ihm initiierten Verbindungsaufbau abholen oder die Antwort wird vom MCAP selbständig zugestellt (*mobile terminated*). Für diese Variante bieten sich neben dem Aufbau einer vom MCAP initiierten CSD-Verbindung vor allem die GSM-Dienste SMS und WAP-Push [For00a], sowie der Aufbau einer GPRS-Verbindung an.

Im Hinblick auf existierende M-Commerce-Dienste bietet der MCAP als Gateway-Lösung die folgenden Vorteile:

- Der Datenfluss zwischen dem mobilen Endgerät und dem MCAP kann reduziert werden. Um möglichst wenig Information auszutauschen, werden möglichst viele Benutzerdaten beim MCAP gespeichert. Die Hauptlast der Kommunikation wird auf den breitbandigen Internetzugang des MCAPs verlagert.
- Der Teilnehmer erhält durch die verschiedenen unterstützten Kommunikationsprotokolle einen effizienten, mobilen Zugang zu unterschiedlichsten Telediensten.
- Der Teilnehmer kann die Dienstauführung nach der Übermittlung seiner Transaktionen (Bankaufträge, Such- und Bestellvorgänge, etc.) unterbrechen und zu einem späteren Zeitpunkt sich die Resultate ansehen.

Bezogen auf die Sicherheit gilt, dass sich der Teilnehmer auf den MCAP verlassen muss, da z.B. sämtliche Transaktionen mit finanziellen Auswirkungen vom Teilnehmer zum MCAP übertragen werden. Um die Agentenmigration zusätzlich zu sichern, unterstützt der MCAP das SSL-Protokoll. Weiterhin wird davon ausgegangen, dass es sich bei Mobilfunknetzen nach dem GSM-Standard um geschlossene Systeme handelt, d.h. dass kein unbefugter

⁵Flat rate bedeutet, dass der Kunde einen unlimitierten Zugang zum Internet für einen monatlichen Pauschalpreis erhält.

⁶Calling Line Identification Presentation, GSM-Dienstmerkmal, das die Übertragung der Rufnummer des A-Teilnehmers, d.h. des Anrufers erlaubt.

Zugang möglich ist. Ferner erlaubt das WTLS-Protokoll des WAP-Standards eine gesicherte Verbindung zum WAP-Gateway. Damit der WAP-Gateway selbst nicht zum Sicherheitsrisiko wird, ist es vorteilhaft, wenn der WAP-Gateway und der MCAP auf dem gleichen Netzknoten installiert sind. Die Vereinigung von WAP-Gateway und MCAP auf einem physikalischen Rechner ist allerdings für das Gesamtkonzept der agentenbasierten Dienstleistung nicht zwingend notwendig und wird zusätzlich dadurch erschwert, dass GSM-Netzbetreiber vermehrt die Nutzung ihres eigenen WAP-Gateways vorschreiben.

5.3 Bankgeschäfte mittels mobiler Endgeräte

Elektronische Bankgeschäfte mittels eines privaten Rechners sind in Deutschland via *T-Online* bzw. dessen Vorgänger *Bildschirmtext* (BTX) bereits seit November 1980 möglich. Zu dieser Zeit machten die Banken gerade ihre ersten Versuche mit Geldautomaten. Nach dem bundesweiten Start von BTX im Sommer 1983 mit rund 10.000 sogenannter *Telekonten* stieg die Anzahl der elektronisch geführten Konten stetig weiter an. In einer ersten Prognose ging man sogar davon aus, dass man bereits 1986 eine Million Teilnehmer erreichen würde. Diese Zahl wurde aber erst rund 10 Jahre später, im Januar 1996, erreicht [KZHW98]. Die BTX-Nachrichten wurden als Texte oder Grafiken im öffentlichen Kommunikationsnetz übermittelt und auf dem Bildschirm eines Fernsehempfängers, PCs oder speziellen BTX-Terminals wiedergegeben. Später wurde unter der Bezeichnung *Datex-J* ein neues Konzept eingeführt. Nun konnten auch Endgeräte angeschlossen werden, die im ASCII-Zeichenmodus arbeiten. 1995 wurde der Dienst mit einem Zugang zum Internet versehen und der Produktname *T-Online* eingeführt. Unter diesem Namen firmiert heute die Aktiengesellschaft mit Sitz in Darmstadt, die mit rund 9 Millionen Kunden Europas größter Internetdienstanbieter ist [T-O01].

HBCI

Von Anfang an dienten beim BTX-Homebankingdienst die sogenannten PINs⁷ und TANs⁸ als Sicherheitsinstrumente. Um gerade diese Sicherheitsmechanismen zu substituieren und dem höheren Sicherheitsbedarf, bedingt durch die Nutzung des offenen Internets anstelle des geschlossenen BTX/Datex-J/T-Online-Systems, Rechnung zu tragen, wurde 1996 das *Home-Banking Computer Interface* (HBCI) vom Bundesverband deutscher Banken (BdB) entworfen und spezifiziert. Zum Leidwesen der Mobilfunkbenutzer wurden allerdings die schmalbandigen, mobilen Zugangsnetze nicht in die Konzeption des Protokolls miteinbezogen. Die aktuelle HBCI-Version 2.2 wurde im Mai 2000 veröffentlicht und unterstützt die folgenden Geschäftsfälle:

- Konto- und Umsatzinformationen
- Saldenabfrage
- Überweisung (Einzel- und Sammelüberweisung)
- Lastschrift
- Daueraufträge
- Termineinlagen
- Wertpapiere
- Zahlungsverkehr Ausland
- Bestellung von Karten, Schecks, Sorten und Devisen
- Kartensperre

⁷Persönliche Identifikationsnummer

⁸Transaktionsnummer

Der HBCI-Standard wird inzwischen von praktisch allen deutschen Banken und Sparkassen unterstützt. Weiterhin ist seit Anfang 1999 durch eine Arbeitsgruppe des ECBS⁹ die Ausweitung von HBCI auf europäischer Ebene geplant. Abb. 5.6 zeigt die im HBCI-Standard festgelegten plattform- und endgeräteunabhängigen Schnittstellen [BdB00]:

1. Schnittstelle zwischen dem Kundenprodukt und dem Kreditinstitutssystem
2. Schnittstelle zwischen dem Kundenprodukt und den Sicherheitsmedien (Chipkarte, Disketten, Festplatte etc.)

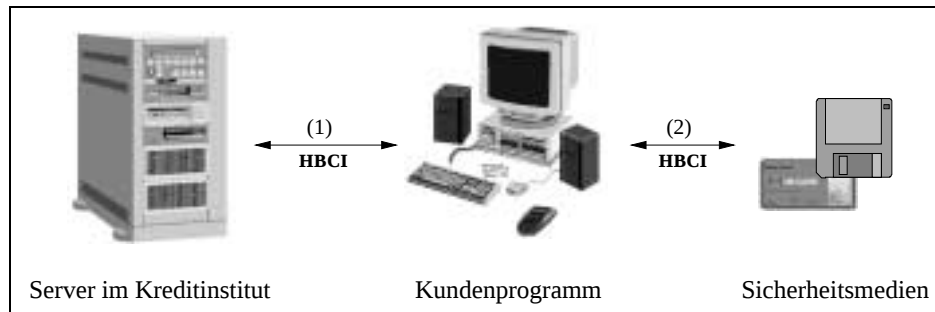


Abbildung 5.6: HBCI-System

Das HBCI-System ist als klassisches Client/Server-System konzipiert. Dabei ist der HBCI-Server bei einem Kreditinstitut angesiedelt. Das Kundenprogramm kann als ausführbares Programm und als Java-Applet, das von einem Webserver geladen wird, konzipiert sein. Beide Lösungen laufen lokal auf dem Rechner des Kunden. Die Teilnehmerdaten sind auf einer Chipkarte oder einer Diskette gespeichert, und das Speichermedium ist durch das Kennwort des Teilnehmers geschützt. Details über den Ablauf eines HBCI-Dialogs und die verwendeten Sicherheitsmechanismen befinden sich im Anhang dieser Arbeit.

5.3.1 Anforderungen an eine mobile HBCI-Lösung

Der Einsatz von HBCI in Mobilfunkgeräten nach dem GSM-Standard ist nicht sinnvoll, da die Übermittlung der umfangreichen, synchronen HBCI-Nachrichten über die schmalbandigen Mobilfunkkanäle erhebliche Verzögerungen verursachen und die daraus resultierende Wartezeit für den Benutzer nicht akzeptabel und teuer ist. In dieser Arbeit wird daher das Ziel verfolgt, eine Anwendung zu entwickeln, die einen agentenbasierten Homebankingdienst (AHB) für das zellulare GSM-Mobilfunknetz (Kap. 2.2.2) ermöglicht [HGF98]. Die Anforderungen an diesen Dienst, die als Richtlinien für die Konzeption dienen, können wie folgt beschrieben werden:

- maximale Reduktion der Datenübertragung,
- maximale Reduktion der Online-Zeit eines Benutzers,
- Skalierbarkeit der Anwendung für hohe Benutzerzahlen,
- komfortable und einheitliche Benutzeroberfläche (speziell für mobile Endgeräte) und
- ausreichende Sicherheit für einen Bankdienst.

GSM-Datenverbindungen erlauben lediglich geringe Übertragungsraten. Daher ist die Reduzierung der zu übertragenden Datenmenge ein vorrangiges Ziel. Bei einer traditionellen Homebanking-Anwendung werden die kompletten HBCI-Nachrichten durch das Kommunikationsnetz übertragen. Mit Hilfe eines MAS ist nur noch die Übertragung des Agenten

⁹European Committee for Banking Standards

durch das Netz notwendig und der MA arbeitet fortan als ein Vertreter des Benutzers. Die gesamte Kommunikation zwischen dem MA und dem Homebanking-Dienstanbieter findet dann lokal auf der Serverseite statt. Hauptsächlich wird eine Reduktion der Online-Zeit dadurch erzielt, dass der MA nur zweimal (einmal hin und einmal zurück) durch das Netz übertragen werden muss und keine weiteren Nachrichten für die verschiedenen Banktransaktionen ausgetauscht werden müssen. Erfahrungen aus der Praxis lehren, dass auch die Bearbeitungszeit des Transaktionsservers der Bank nicht zu vernachlässigen ist. Da in einer agentenbasierten Lösung eine Online-Verbindung nur zum Übertragen eines MAs benötigt wird, muss ein Benutzer nicht auf die Ergebnisse des Servers warten. Die Online-Zeit wird dadurch weiter verkürzt.

Weiterhin soll der AHB-Dienst in der Lage sein, gleichzeitig mehrere Aufträge zu bearbeiten. Darüberhinaus ist eine komfortable Benutzeroberfläche (*graphical user interface*) für den Teilnehmer von großer Bedeutung. Beim mobilen AHB-Dienst wird daher die Anwendung als ein allgemeiner Dienstagent (Kap. 3.2.4) konzipiert, der den AHB-Dienst zu verschiedenen Endgeräten transportieren kann, so dass der Teilnehmer dort jeweils die ihm bekannte, personalisierte Menüführung vorfindet [HB99].

5.3.2 Mobiles, agentenbasiertes HBCI-Szenario

In diesem Abschnitt wird der neu konzipierte AHB-Dienst vorgestellt. Abb. 5.7 zeigt eine mobile Bankanwendung auf Basis der klassischen HBCI-Lösung. Beim Bankinstitut läuft ein Transaktionsserver (Bankserver), der Aufträge entgegennimmt und bearbeitet. Auf dem mobilen Endgerät (Client) läuft die HBCI-Software, die es erlaubt, Aufträge an die Bank zu definieren und abzugeben. In diesem Szenario besteht zwischen diesem Client und dem Bankserver eine GSM-Datenverbindung, die für die Kommunikation über das HBCI-Protokoll genutzt wird.

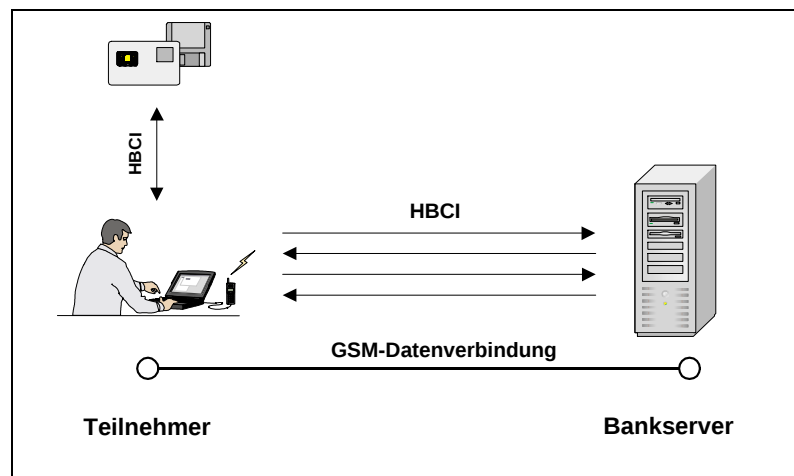


Abbildung 5.7: Klassische HBCI-Anwendung

Ein Vorteil dieses Szenarios ist es, dass die HBCI-Nachrichten direkt beim Teilnehmer generiert werden. Zwar handelt sich hierbei um den einfachsten Weg, eine mobile Homebanking-Anwendung zu realisieren, aber aus Sicht des Teilnehmers ist dieser Dienst nicht effektiv genug, da das Anwendungsprogramm auf dem Client recht umfangreich ist und alle HBCI-Nachrichten vom mobilen Endgerät generiert und bearbeitet werden müssen. Aus diesen

Gründen eignet sich die klassische HBCI-Anwendung nicht für kleine, prozessorschwache Endgeräte wie z.B. Mobiltelefone. Weiterhin muss beachtet werden, dass sämtliche Nachrichten über eine GSM-Verbindung mit geringer Übertragungskapazität übertragen werden müssen. Während der Server dann die Nachrichten bearbeitet, verbleibt der Client in Warteposition, da es sich um eine synchrone Kommunikation handelt. Möchte der Teilnehmer den Dienst von verschiedenen Rechnern aus nutzen, dann muss auf jedem Rechner entweder ein Chipkartenleser installiert oder die Benutzerdaten müssen gespeichert sein. Dies verursacht eine Hardware-Redundanz. Durch die verschiedenen Speicherstellen der privaten Schlüssel wird ferner die Sicherheit des Systems verringert.

Abb. 5.8 zeigt das entwickelte Konzept des AHB-Dienstes [HS99], bei dem die Nachteile der Ende-zu-Ende HBCI-Lösung beseitigt wurden. Dafür wird zwischen dem Teilnehmer und der Bank die Kommunikationsplattform MCAP (Kap. 5.2) eingesetzt. Der MCAP ist dabei aus Sicht der Bank ein gewöhnlicher HBCI-Client. Aufgrund der Datenmenge sollte die Datenübertragung zwischen dem Bankserver und dem MCAP durch eine breitbandige Verbindung realisiert werden. Aus Benutzersicht arbeitet der MCAP wie ein Homebanking-Dienstanbieter für mobile Teilnehmer. Die Teilnehmerdaten werden zentral im MCAP gespeichert. Der Teilnehmer nutzt einen allgemeinen Dienstagenten, den Homebanking Terminal Agent (HTA), um MAs über eine GSM-Verbindung zum MCAP zu senden.

Ein **Homebanking Terminal Agent** ist der Programmcode für die Benutzeroberfläche und Logik des AHB-Dienstes, der das Erstellen und Verwalten von MAs für HBCI-basierte Banktransaktionen ermöglicht.

Bei den für die Kommunikation mit dem MCAP eingesetzten MAs handelt es sich um spezielle Dienstagenten (Kap. 3.2.4).

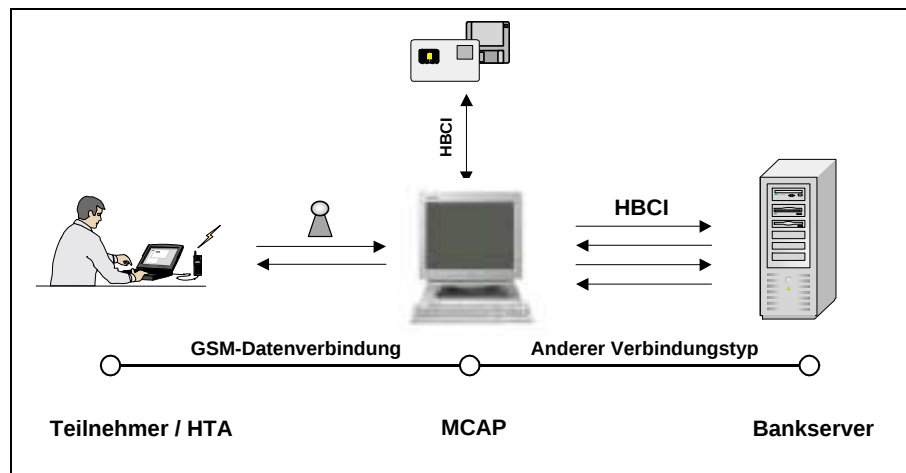


Abbildung 5.8: Konzeption des AHB-Dienstes

Da die HBCI-Nachrichten nicht direkt vom HTA generiert werden, kann der HTA einfach und klein sein. Für die Verbindung zwischen dem HTA und dem MCAP wurde die asynchrone Kommunikation gewählt, was den Vorteil hat, dass Verbindungen nur für die Übertragung der Dienstagenten (DAs) aufgebaut werden müssen.

Dienstagenten werden beim AHB-Dienst dafür eingesetzt, um Bankaufträge und Ergebnisse zwischen dem mobilen Endgerät und dem MCAP zu übertragen.

Durch die Einführung des MCAPs und die DAs wird zwar insgesamt eine größere Datenmenge über die involvierten Kommunikationsnetze übertragen, aber die über die Mobilfunkstrecke zu übertragene Datenmenge wird reduziert. Ein weiterer Vorteil ist, dass der Teilnehmer seine GSM-Verbindung schließen kann und nicht mehr auf die Bearbeitung seiner Aufträge warten muss. Der intensive Nachrichtenaustausch findet nun zwischen dem Bankserver und dem MCAP statt, zwischen denen eine breitbandige Verbindung besteht. Dadurch, dass alle Daten des Teilnehmers zentral gespeichert werden, wird weiterhin erreicht, dass der Teilnehmer keinen Chipkartenleser am Endgerät benötigt und dass es nur eine einzige Speicherstelle der privaten Schlüssel gibt. Ein Nachteil dieser Lösung ist, dass der MCAP ein eigenes Sicherheitskonzept benötigt.

5.3.3 Konzeption und Ablauf des AHB-Dienstes

Dieser Abschnitt beschreibt detailliert das Konzept und den Ablauf des AHB-Dienstes. Wie in Abb. 5.8 zu sehen ist, besteht der AHB-Dienst aus zwei Client/Server-Szenarien. Einerseits kommuniziert der MCAP mit einem Bankserver über das HBCI-Protokoll. Der MCAP ist dabei aus der Sicht der Bank ein gewöhnlicher HBCI-Client, der HBCI-Nachrichten verschicken und bearbeiten kann. Alle HBCI-relevanten Details sind ausschließlich im MCAP implementiert. Andererseits arbeitet der MCAP aus der Sicht des HTAs als ein Server, der dem Teilnehmer Bankdienste anbietet. Ein Teilnehmer benutzt den HTA, um Aufträge abzugeben und Ergebnisse zu erhalten. Die Funktionen des AHB-Dienstes werden durch den MCAP bereitgestellt. Zusätzlich müssen die Teilnehmerdaten eines Benutzers entsprechend konfiguriert werden. Es können mehrere Versionen des HTAs für unterschiedliche Hardware-, Kommunikations- bzw. Dienstqualitäten existieren und je nach Bedarf wird die entsprechende Version zum Endgerät übertragen. Damit wird die Forderung nach einer personalisierten Benutzerschnittstelle erfüllt.

Die Kommunikation zwischen dem MCAP und einem HTA erfolgt über DAs, die Aufträge und Ergebnisse zwischen den beiden übertragen. In dieser Hinsicht kann man die Aktivität eines DAs als eine Art von Kommunikationsprotokoll bezeichnen. Ein DA kann gleichzeitig mehrere Aufträge übertragen und bearbeiten. Der Dienst sieht vor, dass eine Kommunikationsverbindung *nur* zur Übertragung von Programmcode oder DAs aufgebaut wird und dass diese nach der Übertragung direkt wieder abgebaut wird.

Da die Auftragsbearbeitung beim Bankserver eine relativ lange Zeit in Anspruch nimmt, wurde die Kommunikationsplattform MCAP derart konzipiert, dass für den AHB-Dienst zwei Prozesstypen, Dienstanbieter- (DAP) und Bearbeitungsprozesse (BP), unterstützt werden. Der erste nimmt die DAs entgegen und erzeugt für jeden DA einen BP. Der BP schickt die Aufträge des DAs dann zur Bank und teilt dem DA später die Antworten mit. Abb. 5.9 zeigt schematisch, wie der AHB-Dienst durch MCAP, BPs, DAs und TAs realisiert wird. Dabei hilft der Terminal Agent (TA) bei der Laden des HTAs, falls der AHB-Dienst auf dem Endgerät zum ersten Mal genutzt wird.

Während in Abb. 5.10 die Benutzeroberfläche des TAs dargestellt, bei dem der HTA noch nicht auf dem Endgerät verfügbar ist, zeigt Abb. 5.11 die Realisierung der graphischen Oberfläche des HTAs zur clientseitigen Verwaltung der DAs des AHB-Dienstes.

Abb. 5.12 zeigt den vollständigen Ablauf des AHB-Dienstes in sieben Schritten. Dabei wird angenommen, dass das Agentensystem aus vier Orten (A bis D) besteht, wobei B die Heimat ist. Der Teilnehmer befindet sich am Ort A, der MCAP steht in C, und der Regi-

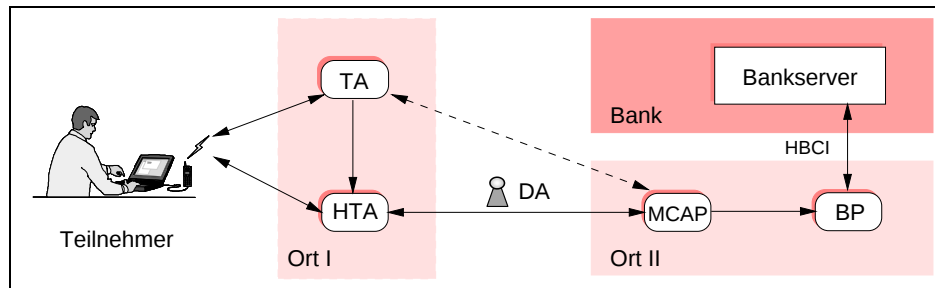


Abbildung 5.9: Realisierung des AHB-Dienstes

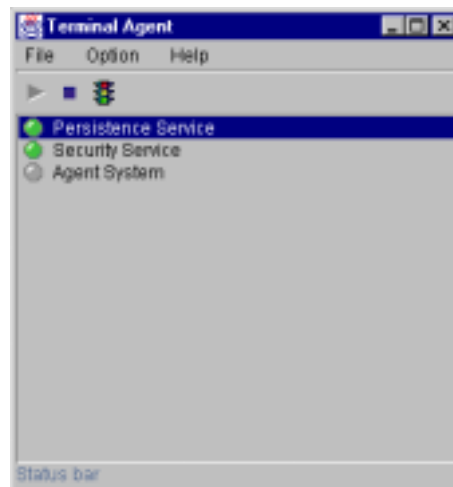


Abbildung 5.10: Graphische Oberfläche des TAs

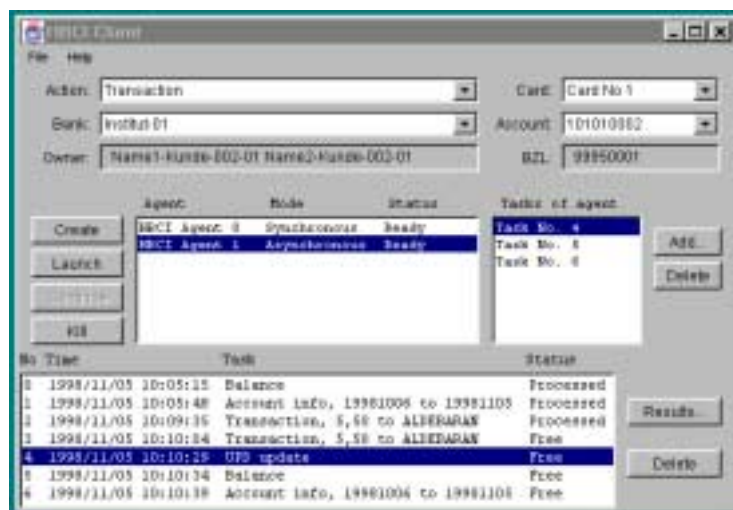


Abbildung 5.11: Graphische Oberfläche des HTAs

strierungsort heißt D. E repräsentiert die Domäne einer Bank, in der sich der Bankserver befindet.

1. Starten des MCAPs

Zunächst muss sich der MCAP beim Registrierungsdienst anmelden, damit TAs den Dienst jederzeit lokalisieren können.

2. Authentifizierung eines Benutzers

Beim Teilnehmer (Ort A) ist das lokale Agentensystem bereits gestartet und der TA agiert als Benutzerschnittstelle zwischen dem Teilnehmer und dem System. Der Teilnehmer muss sich mit seinem Benutzernamen und einem Kennwort beim TA anmelden. Für diesen Vorgang gibt es zwei mögliche Wege: Entweder die Authentifizierung über einen TA (Abb. 5.12(a)) bei einer zentralen Datenbank (2a bis 2d) oder die lokale Authentifizierung über eine Chipkarte bzw. Diskette. Bei diesem Vorgang werden keine Daten über das Kommunikationsnetz transportiert. Nach erfolgreicher Anmeldung zeigt der TA eine Liste der für den Teilnehmer verfügbaren Dienste an (2e).

3. Lokalisierung des MCAPs (optional)

Wenn der Teilnehmer den AHB-Dienst von dem gewählten Endgerät aus zum ersten Mal (3a) nutzt, dann überprüft der TA zunächst, ob der benötigte Programmcode bereits lokal verfügbar ist. Ist dies nicht der Fall, versucht er mit Hilfe des Registrierungsdienstes zunächst den MCAP zu lokalisieren (3b und 3c). Gelingt dies nicht, so erhält der Teilnehmer eine Fehlermeldung und der ganze Vorgang wird abgebrochen. Ist der Programmcode bereits lokal verfügbar, sind die Schritte 3b bis 4b nicht notwendig.

4. Übertragung des Programmcodes (optional) und Starten des HTAs

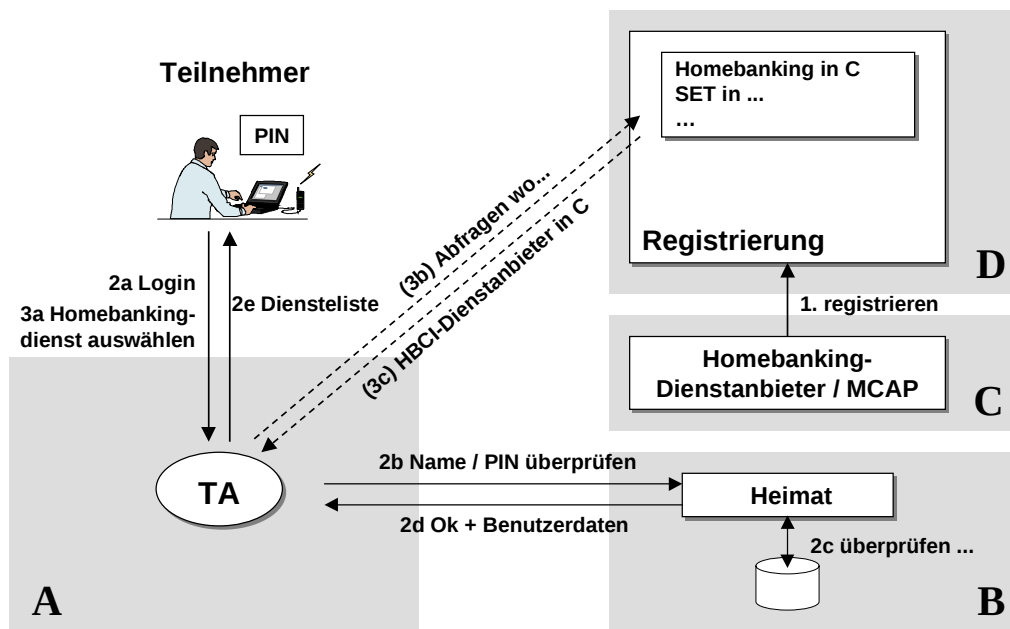
Ist der Code lokal noch nicht vorhanden, dann lädt der TA den HTA vom Agentenserver MCAP. Dabei überträgt der TA die Anforderungen des Teilnehmers und Details über das verwendete Endgeräte als Parameter zum MCAP, damit dieser einen entsprechenden (auf den Teilnehmer zugeschnittenen) Programmcode (HTA) auswählen kann. Nach der Übertragung startet der TA den HTA als einen getrennten Prozess, der ebenfalls über eine Benutzerschnittstelle verfügt (4c). Der herkömmliche TA wird für den weiteren Ablauf des AHB-Dienstes nicht mehr benötigt.

5. - 6. Bearbeitung der Aufträge

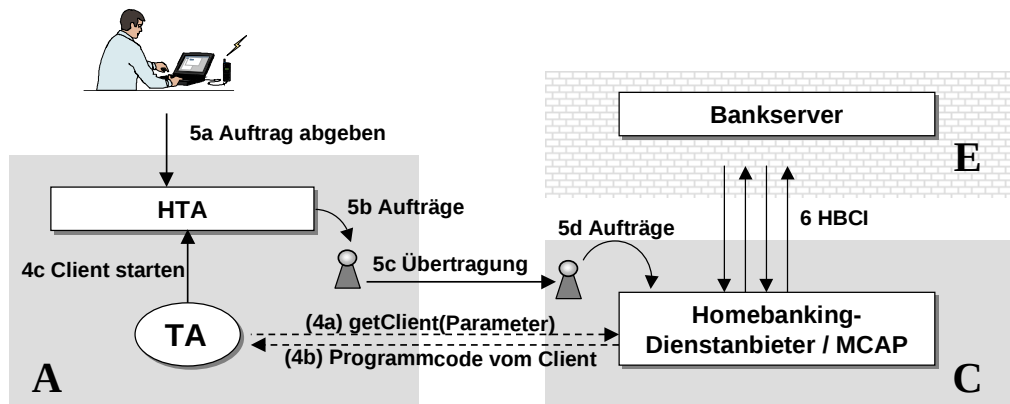
Der Teilnehmer kann mit Hilfe des HTAs Agenten erzeugen, Aufträge definieren und DAs abschicken (5a und 5b), die dann zum MCAP übertragen werden (5c). Die Verbindung wird nur für diesen Zweck aufgebaut und kann direkt nach dem Transport des DA wieder abgebaut werden. Nach der Ankunft des Agenten in C, meldet sich dieser beim MCAP. Dies geschieht durch eine lokale Authentifizierung. Der MCAP erzeugt nach erfolgreicher Authentifizierung einen BP für den DA und kann danach weitere DAs bedienen. Der DA übergibt dem BP seine Aufträge (5d). Der BP generiert nun aus diesen Aufträgen HBCI-Nachrichten, sendet diese über das klassische HBCI-Protokoll (6) zum Transaktionsserver in der Bank, wo die weitere Bearbeitung der Aufträge erfolgt und wartet auf die Resultate.

7. Rückkehr des DAs

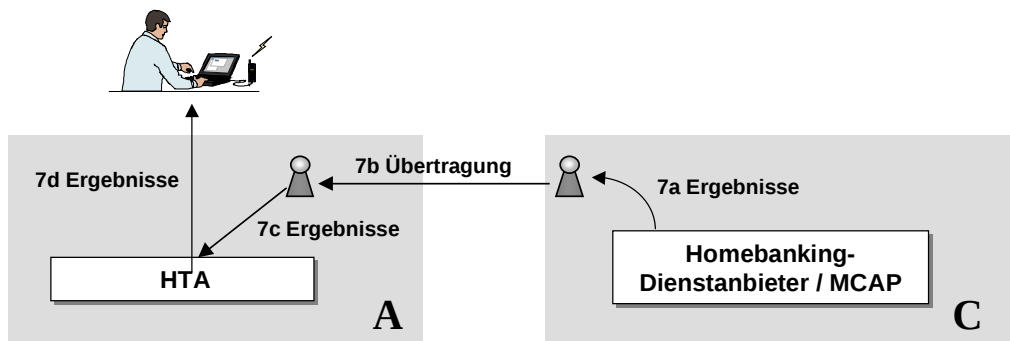
Nach der Auftragsbearbeitung erhält der DA die Ergebnisse vom BP zurück (7a) und bereitet seine Rückkehr zum HTA dadurch vor, dass er unnötige Datenfelder löscht (Selektion). Dies reduziert die Größe des DAs und verkürzt die Übertragungszeit. Für die Rückkehr des DAs zum Teilnehmer gibt es mehrere Möglichkeiten. Ist der Benutzer zu diesem Zeitpunkt



(a) Authentifizierung eines Teilnehmers und Suche nach dem AHB-Dienst



(b) Start des HTAs und Bearbeitung der Aufträge



(c) Rückkehr des DA

Abbildung 5.12: Ablauf des AHB-Dienstes

noch online, kann der DA direkt zurück migrieren. Wenn die Verbindung allerdings nicht mehr besteht, dann wechselt der DA zunächst in den Zustand *Warten* und versucht in regelmäßigen Zeitabständen wieder zum Teilnehmer zurückzukehren. Der Teilnehmer selbst kann den DA auch jederzeit mit Hilfe des Agentenmanagers zurückzuholen. Nachdem der DA zurückkehrt ist, meldet er sich beim HTA durch den in Schritt 5 beschriebenen Authentifizierungsvorgang. Bei Erfolg erhält der HTA dann die Bearbeitungsergebnisse, die er dann an den Teilnehmer weiterreicht.

Nachrichtensequenzen

Nachfolgend werden die wichtigsten Nachrichtensequenzen vom AHB vorgestellt. Abb. 5.13 zeigt die Nachrichtensequenzen beim erstmaligen Starten des Dienstes, d.h. wenn der HTA des AHB-Dienstes noch nicht lokal verfügbar sind. Der TA versucht zunächst, beim Registrierungsdienst den MCAP zu lokalisieren, dann veranlasst er die Übertragung des Programmcodes. Beim nochmaligen Starten des AHB ist der Programmcode bereits lokal vorhanden und der HTA kann direkt gestartet werden.

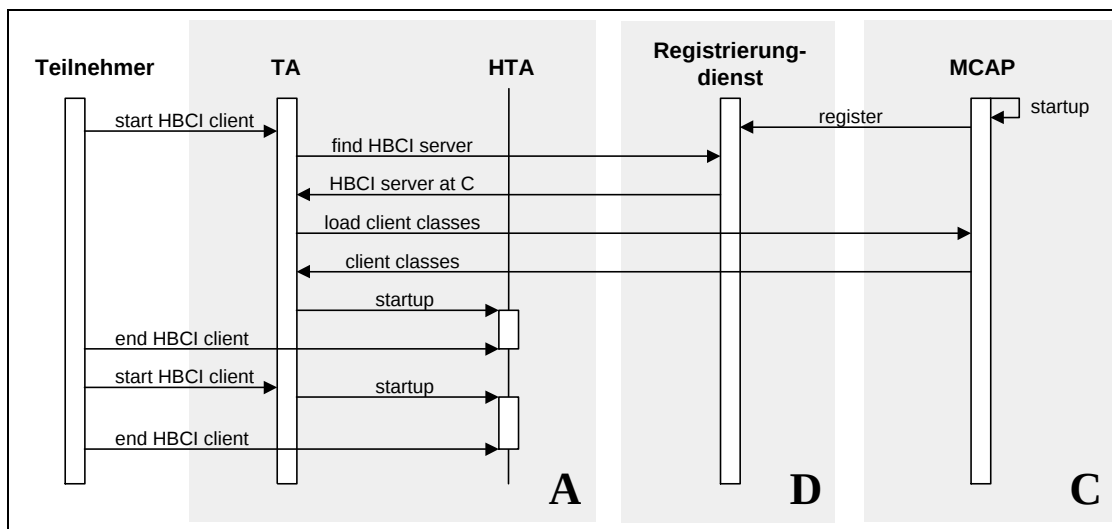


Abbildung 5.13: Nachrichtenfluss beim Start des Dienstes

In Abb. 5.14 ist die Nachrichtensequenz für die Auftragsbearbeitung dargestellt. Man erkennt deutlich, dass im Gegensatz zum klassischen HBCI bei der neu konzipierten Lösung lediglich zwei Verbindungen zur Übertragung des DAs (hin und zurück) über die Luftschnittstelle benötigt werden.

5.4 Such- und Bestelldienste

Ein zweiter M-Commerce-Dienst, der mittels mobiler Agenten konzipiert und realisiert wurde, ist ein Such- und Bestelldienst. Heutige E-Commerce-Dienste erfordern in der Regel, dass der Benutzer einen PC mit Internetzugang nutzt und über eine Festnetztelefonleitung mit Hilfe eines Internetbrowsers verschiedene Händler aufsucht, um deren Sortiment zu begutachten. Dabei werden Produktqualität, Preis und die Vertrauenswürdigkeit des Händlers in der Regel kritisch vom Kunden untersucht, und im Falle eines Kaufwunsches wird das Produkt häufig auch direkt online erworben. Der gesamte Einkaufsvorgang dauert

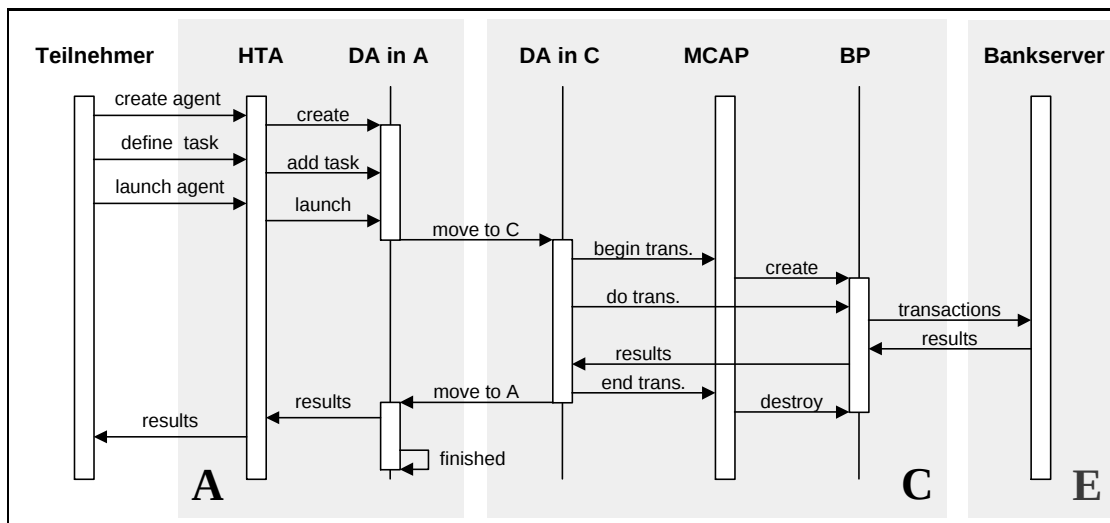


Abbildung 5.14: Nachrichtenfluss bei der Auftragsbearbeitung

zwischen 5 und 15 Minuten, wobei zwischen 150 und 450 *kbyte* Daten ausgetauscht werden [Hüc00].

Soll der gleiche Vorgang, wie in Abb. 5.15 dargestellt ist, über eine GSM-Datenverbindung abwickeln werden, ergeben sich einige schwerwiegende Nachteile. Weder die Wartezeiten beim Aufruf von Webseiten, noch die Kosten (um den Faktor 10 bis 15 höher) sind akzeptabel. Auch WAP (Kap. 2.2.5) und GPRS (Kap. 2.3) werden Einkaufsszenarien im Internet vom mobilen Endgeräten aus alleine nicht entscheidend verbessern können. Vielmehr muss der Dienst derart verändert werden, dass sowohl die Wünsche des Kunden in einer dem Endgerät angepassten Art und Weise entgegengenommen werden, als auch entsprechende Ergebnisse in einem angemessenen Format zurückgeliefert werden. Ein solcher im Rahmen der Arbeit entstandener agentenbasierter Such- und Bestelldienst (ASB), der speziell für nomadische Benutzer konzipiert wurde, wird in den nächsten Abschnitten vorgestellt.

5.4.1 Agentenbasierte Suche

Bisher wurden bereits eine Reihe von E-Commerce-Diensten entwickelt, doch der Gebrauch dieser Dienste ausgehend von herkömmlichen mobilen Endgeräten mit begrenzten Darstellungsmöglichkeiten, wird häufig vernachlässigt. Durch den Einsatz des MCAPs (Kap. 5.2) wird beim ASB-Dienst ein komfortabler Zugang zu E-Commerce-Diensten mittels MAS ermöglicht, siehe Abb. 5.16. Der MCAP nutzt für die agentenbasierte Suche neben dem Agentensystem Voyager auch eine SQL-Datenbank und einen Web/WAP-Server. Voyager ist dabei für die Kommunikation zwischen den verschiedenen Management- und Dienstagenten (vgl. Kap. 3.2.4), sowie für die Agentenmigration mittels des VRMP-Protokolls (Kap. 3.4.1) verantwortlich. Die Datenbank speichert die diversen Anfragen und Benutzer- bzw. Händlerdaten, und der Web/WAP-Server unterstützt die WAP-Kommunikation mit dem Endgerät des Benutzers.

Der MCAP besitzt für jeden registrierten Benutzer und Händler ein eigenes Profil, in dem u.a. Adress- und Bezahlungsinformationen abgelegt sind. Während die Benutzer sich über WAP in den MCAP einwählen, sind die Händler über das Internet mit dem MCAP

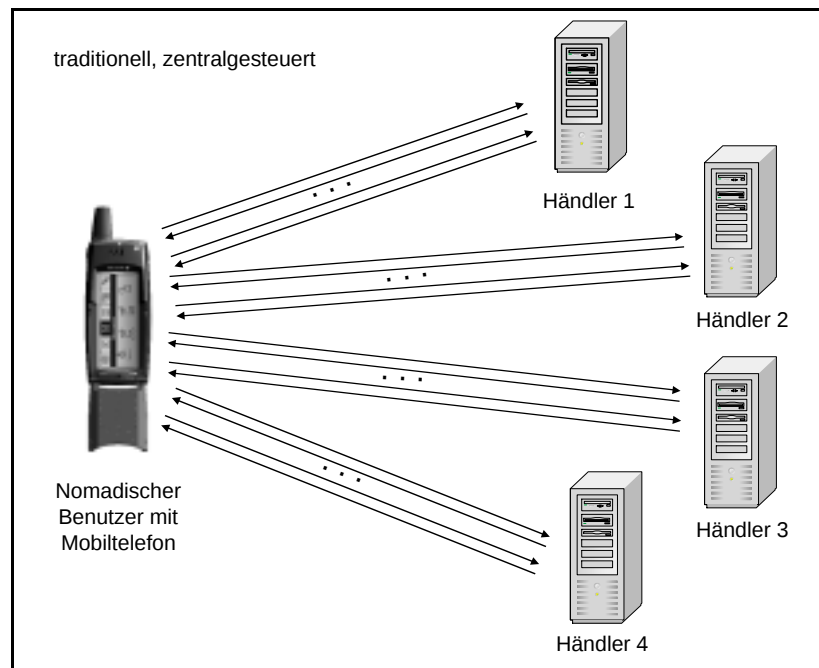


Abbildung 5.15: Heutiger M-Commerce-Dienst

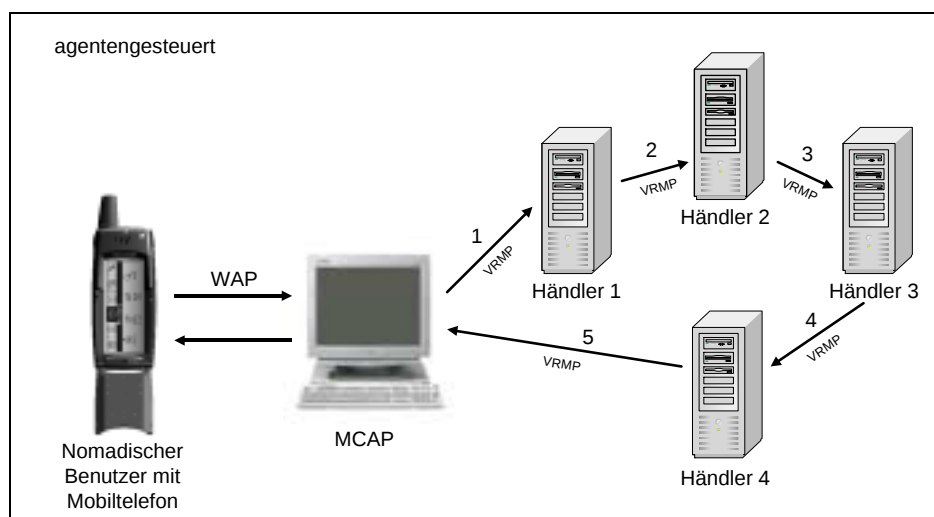


Abbildung 5.16: ASB-Dienst mittels MCAP

verbunden. Der Ablauf des ASB-Dienstes sieht aus Benutzersicht wie folgt aus:

1. Der Benutzer wählt sich in das System ein und authentifiziert sich mit seinem Benutzernamen und seinem Passwort, oder alternativ mittels des CLIP-Dienstmerkmals und seiner MSISDN¹⁰.
2. Nun wählt der Benutzer mittels WAP aus der Liste der verfügbaren Produktkategorien das von ihm gesuchte Produkt aus, spezifiziert seine Suche, und schickt eine entsprechende Anfrage an den MCAP.
3. Für die Dauer der Suche kann der Benutzer den Dienst und die WAP-Verbindung beenden oder die Verbindung halten bis das Resultat gesendet wird.
4. Der Benutzer erhält das Ergebnis und hat danach die Möglichkeit, eine Bestellung und die zugehörige Bezahlung vorzunehmen.
5. Bis zum Erhalt einer Bestätigung der Bestellung und ggf. der Bezahlung kann der Benutzer online bleiben oder den Dienst zwischenzeitlich abbrechen.

5.4.2 Suchmethoden

Abhängig von den Ansprüchen und Anforderungen des Benutzers kann die Suche nach einem bestimmten Produkt beliebig komplex sein. Um unterschiedliche Anforderungen von Benutzern zu unterstützen, erlaubt der neu entwickelte und implementierte ASB-Dienst die Wahl zwischen vier verschiedenen Suchmechanismen:

- | | |
|----------------|--|
| ERSTES: | Der Suchagent bricht die Suche ab, sobald er ein Produkt gefunden hat, das alle verlangten Eigenschaften erfüllt. Bei einfachen Suchen ist die Abbruchbedingung erfüllt, wenn das Produkt zum gewünschten Preis verfügbar ist. Der Kaufvorschlag repräsentiert nicht notwendigerweise das beste Angebot. |
| DREI: | Bei dieser Suchmethode fragt der Agent so lange bei den verfügbaren Händlern nach, bis er drei Händler gefunden hat, die das Produkt entsprechend der vorgegebenen Sucheigenschaften liefern können. Die Kaufvorschläge repräsentieren nicht notwendigerweise die besten drei Angebote. Die Menge der bei diesem Suchmechanismus aufgesuchten Händler hängt stark vom gesuchten Produkt ab. |
| BESTES: | Es werden alle Händler aufgesucht, die das gesuchte Produkt anbieten. Im Falle der agentenbasierten Suche bringt der Agent lediglich die besten drei Ergebnisse zurück zum Benutzer. |
| SMART: | SMART ist eine komplexe Suchmethode, die mehrere Parameter bei der Wahl geeigneter Produkte auswertet, um die Ergebnisse einstufen zu können. Hierbei können den einzelnen Parametern verschiedene Prioritäten zugeordnet werden. Ein geeignetes Suchobjekt für diesen Mechanismus ist z.B. ein Gebrauchtwagen, bei dem neben dem Preis auch der Kilometerstand, das Baujahr, die Farbe, die Ausstattung, etc. eine nicht unerhebliche Rolle bei der Kaufentscheidung spielen. |

¹⁰Mobile Subscriber ISDN-Nummer, die Rufnummer des Mobilfunkteilnehmers

Die bei den vier vorgestellten Suchmethoden übermittelten Datenvolumina und benötigten Suchzeiten werden in Kap. 6.6 analysiert und bewertet.

5.4.3 Konzeption mobiler, agentenbasierter Such- und Bestellszenarien

Zur Leistungsbewertung wurden drei verschiedene Suchszenarien des ASB-Dienstes konzipiert und implementiert. Während die ersten beiden Konzepte einen mobilen Agenten einsetzen und damit eine *ringförmige* Kommunikation unterstützen, das dritte Konzept basiert auf einem stationären Agenten, der mit Hilfe von SQL-Datenbankabfragen die Suchergebnisse zusammenträgt und dabei *sternförmig* kommuniziert. Abb. 5.17 zeigt die unterschiedlichen Informationsflüsse bei der Verwendung von mobilen und stationären Agenten im Detail.

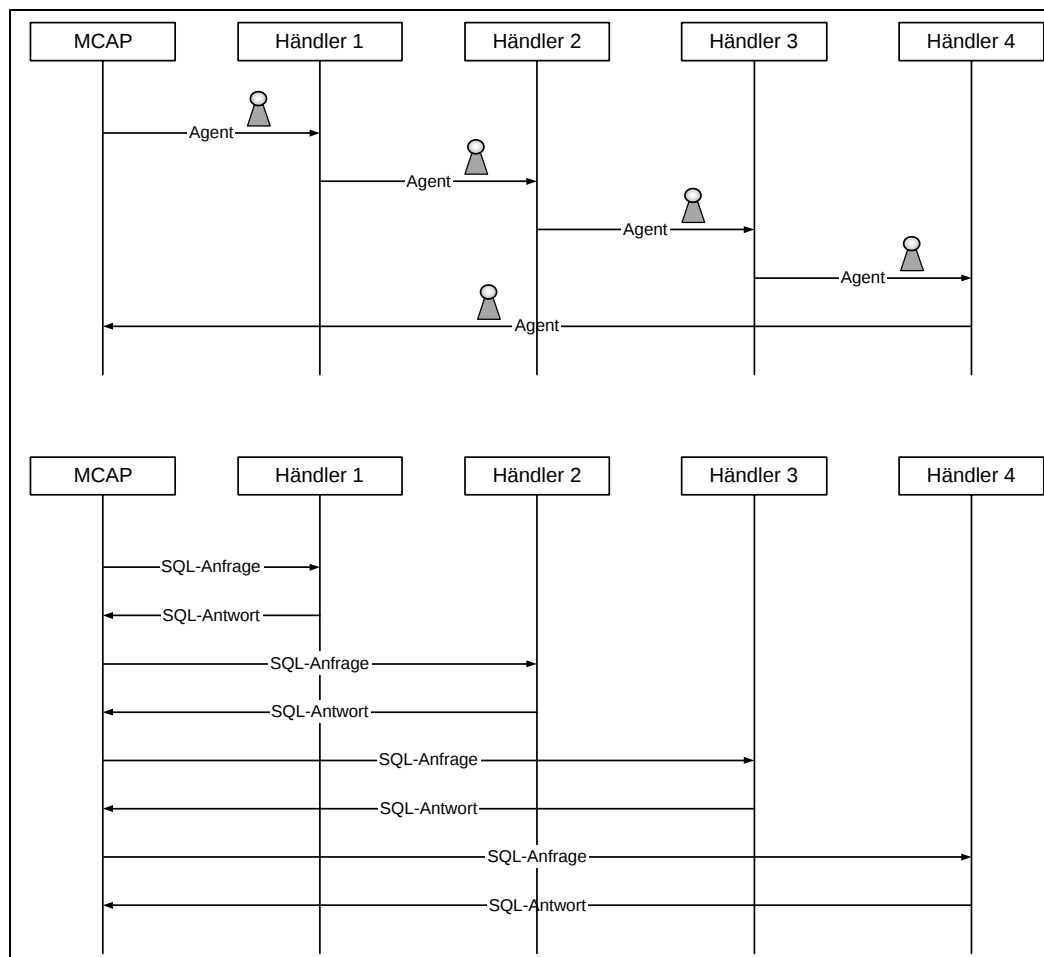


Abbildung 5.17: Informationsfluss beim Einsatz von MAs (oben) und SQL (unten)

(1) Der mobile Suchagent

Diese Lösung sieht auf den Rechnern der Händler die Installation der Agentenplattform *Voyager* (Kap. 3.4.1) und einer SQL-Datenbank¹¹ (Kap. 3.3.5) vor [HH01]. Wenn der Benutzer seine Suche über WAP spezifiziert hat, werden die Suchparameter von einem

¹¹bei der Implementierung wurde die Datenbank mini SQL 2.0 von Hughes verwendet

PHP-Skript aus der vom WAP-Gateway gesendeten HTTP-PDE ausgelesen und an einen mobilen Suchagenten (MSA) der Kategorie allgemeiner Dienstagent (Kap. 3.2.4) übergeben. Der MSA erstellt aus den Datenbankeinträgen des MCAPs in Abhängigkeit vom gesuchten Produkt und ggf. auch vom momentanen geographischen Standort des Benutzers eine Händlerliste und migriert danach zum ersten Händler dieser Liste, siehe Abb. 5.17(oben) und Abb. 5.18. Nach der Ankunft beim Händler durchsucht der MSA dort lokal dessen Datenbank nach dem gewünschten Produkt. Wenn der MSA ein Produkt findet, welches den Suchkriterien entspricht, legt er die entsprechende Information in seinem internen, temporären Speicher (*temporary memory* [CAM99]) ab. In Abhängigkeit von der Suchmethode (Kap. 5.4.2) und der Qualität des gefundenen Produktes wird der MSA seinen temporären Speicher bereinigen bzw. überschreiben. Danach verlässt der MSA die Händlerplattform und entscheidet, ob er zum nächstem Händler auf seiner Liste migrieren soll, oder ob er zum MCAP zurückkehrt. Nachdem der Agent entweder seine Suche erfolgreich beendet hat oder kein Händler mehr besucht werden kann, kehrt der MSA zum MCAP zurück. Dort überträgt der mobile Agent seinen temporären Speicher an die Datenbank des MCAPs.

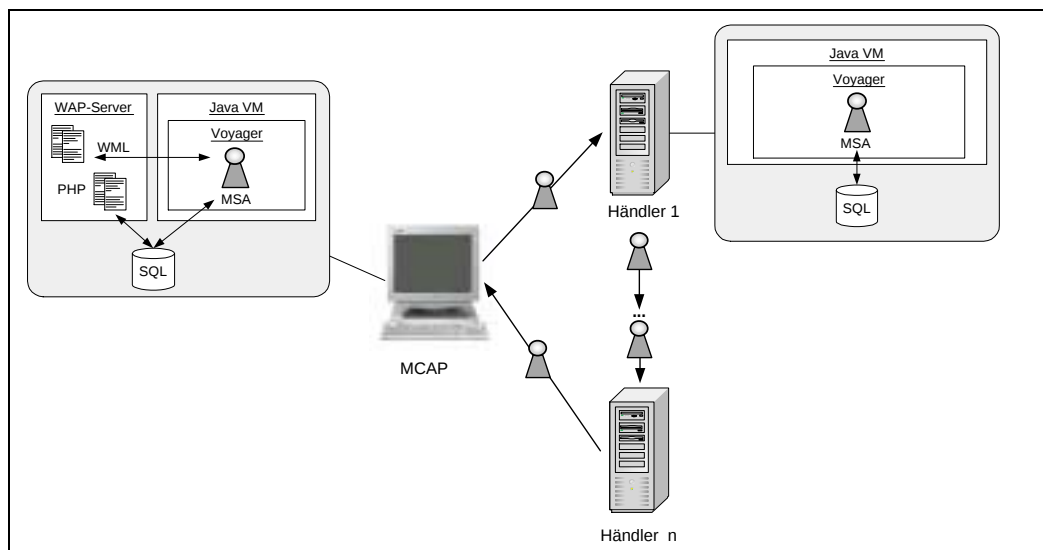


Abbildung 5.18: Mobiler Suchagent für den ASB-Dienst

Zusammengefasst kann man festhalten, dass die agentenbasierte Suche im Vergleich sowohl zu herkömmlichen als auch zu neueren, intelligenten datenbankbasierten Suchanfragen eine Reihe von Vorteilen bietet:

- Agenten können die Suchergebnisse direkt auf der Händlerseite filtern, d.h. die Information herausfinden, die den Benutzer wirklich interessieren.
- Es ist möglich, die Suchagenten mit einer Verhandlungsstrategie für eventuelle Preisverhandlungen auszustatten.
- Agenten können kryptographisch einfacher geschützt werden als Datenbankabfragen.
- Die agentenbasierte Suche kann auch über heterogene Netze und Firewalls hinweg durchgeführt werden (siehe auch Kap. 4). Dies ist für gewöhnliche Datenbankabfragen, wie z.B. mittels SQL, problematisch, da diese auf bestimmte Transport- und Netzwerkprotokolle angewiesen sind.

(2) Der mobile Suchagent mit XML-Schnittstelle

Diese Realisierung des ASB-Dienstes ist der vorher beschriebenen ähnlich. Auch für den Einsatz eines mobilen Suchagenten mit XML-Schnittstelle (MSA+) ist auf allen Händlerrechnern die Installation einer Agentenplattform erforderlich. Der entscheidende Unterschied besteht darin, dass der MSA diesmal mit dem MCAP und den Händlern über XML (Kap. 3.3.5) kommuniziert, siehe Abb. 5.19, und dabei keine direkte Kommunikation mit der SQL-Datenbank stattfindet. Der MSA benötigt daher auch keine SQL-Fähigkeiten, sondern kommuniziert lediglich mit anderen Agenten.

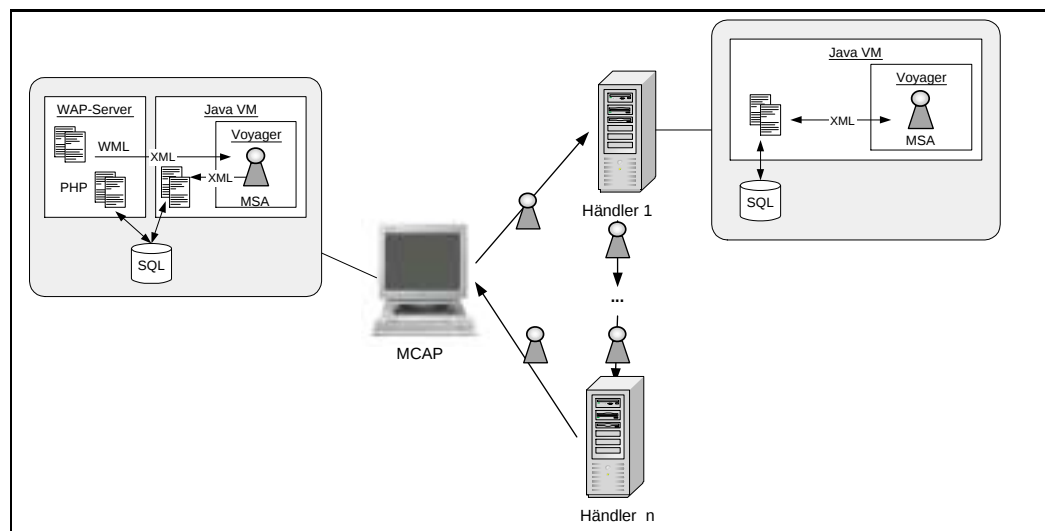


Abbildung 5.19: Mobiler Suchagent mit XML-Schnittstelle

Wenn der Benutzer seine Suche über WAP spezifiziert hat, werden die Suchparameter von einem PHP-Skript aus der vom WAP-Gateway gesendeten HTTP-PDE ausgelesen und der MCAP erstellt eine entsprechende Händlerliste im XML-Datenformat (Kap. 3.3.5). Anschließend wird ein MSA+ gestartet, dem eine Händlerliste übergeben wird, so dass der MSA+ direkt in die Lage versetzt wird, zum ersten Eintrag auf der Liste zu migrieren. Der MSA+ arbeitet die Händlerliste sequentiell ab und bricht seine Reise erst ab, wenn die Suche erfolgreich war (abhängig von der gewählten Suchmethode (Kap. 5.4.2)) oder die Liste komplett abgearbeitet wurde. Auf dem Händlerrechner kommuniziert der MSA+ in diesem Suchszenario nicht direkt mit der Datenbank des Händlers, sondern durch den Austausch von XML-Dateien mit einem lokalen Agenten des Händlers. Die Fähigkeiten des lokalen Agenten liegen im direkten Einflussbereich des Händlers. Daher ergeben sich die folgenden qualitativen Vorteile für mobile Agenten mit XML-Schnittstelle:

- Der XML-basierte MA ist flexibel einsetzbar, da dem Händler eine generelle Schnittstelle angeboten wird. Existierende Produktdatenbanken können besser eingebunden werden. Die Notwendigkeit, eine SQL-Datenbank zu führen, existiert für den Händler nicht.
- Die zusätzliche Kommunikationsschicht zwischen MSA+ und dem Händlerkommunikationssystem erlaubt den Austausch von weiteren Informationen zwischen Agent und Händler.
- Die XML-Schnittstelle vereinfacht Veränderungen am Kommunikationsprotokoll, an den Agenten oder am Händlersystem.

- Durch den Austausch von Informationen über XML werden die Möglichkeiten der Interoperabilität von MAs verschiedener Agentensysteme verbessert.

(3) Der stationäre Suchagent mit SQL-Schnittstelle

Bei dieser Suchanwendung kommt ein stationärer Suchagent (SSA) zum Einsatz, der vom MCAP aus nacheinander die SQL-Datenbanken der Händler abfragt, siehe Abb. 5.17 (unten). Der SSA ist in PHP programmiert und benötigt kein Agentensystem, da bei dieser Suche keine Agentenmigration und -kommunikation eingesetzt werden, siehe Abb. 5.20. Der SSA entspricht daher dem RPC-Mechanismus (Kap. 3.7).

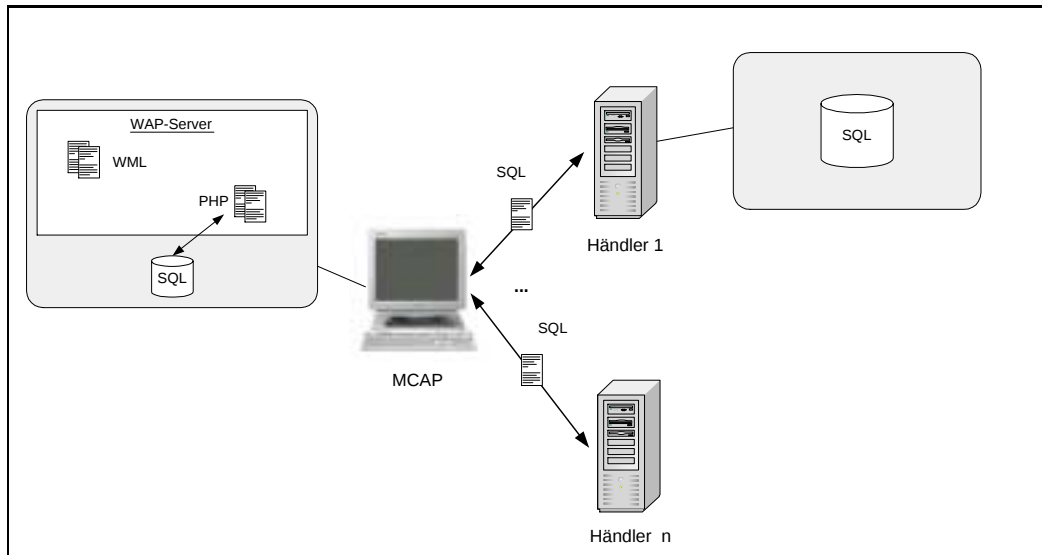


Abbildung 5.20: Stationärer Suchagent

Wenn der Benutzer seine Suche über WAP spezifiziert hat, werden die Suchparameter von einem PHP-Skript aus der vom WAP-Gateway gesendeten HTTP-PDE ausgelesen und dem SSA zusammen mit der entsprechenden Händlerliste übergeben. Der SSA baut nun sukzessive eine Verbindung zu den Händlern auf und untersucht die SQL-Datenbank der Händler nach dem entsprechenden Produkt. Dabei arbeitet der SSA die Händlerliste solange ab, bis die Suchkriterien erfüllt sind bzw. bis der letzte Händler auf der Liste erreicht ist. Alle Resultate werden direkt zum MCAP übertragen und dort ausgewertet. Die gesamte Verarbeitung erfolgt beim MCAP. Die Rechenleistung der Händlerinfrastruktur kann für diesen Zweck nicht genutzt werden. Die Handelsplattformen müssen lediglich SQL-Anfragen unterstützen, d.h. die Installation eines Agentensystems ist in diesem Suchszenario nicht erforderlich. Da die SQL-Anfragen direkt von PHP aus durchgeführt werden können, ist diese Suchanwendung im Vergleich zu den vorher vorgestellten agentenbasierten Lösungen erheblich schneller. Dennoch lassen sich eine Reihe qualitativer Nachteile feststellen:

- Mit mobilen Agenten kann eine Kommunikation über die Grenzen von Firewalls und homogenen Kommunikationnetzen hinaus einfacher ermöglicht werden als mit SQL-Anfragen, da man weitgehend unabhängig von den unterliegenden Protokollschichten ist.
- Es sind umfangreiche Bemühungen notwendig, um SQL-Anfragen zu kodieren und zu signieren.
- Der Händler ist an den Einsatz einer SQL-Datenbank gebunden.

- Es wird dem Händler keine Möglichkeit gegeben, individuell auf die Anfrage zu reagieren. Der Händler kann den Zugriff auf seine Daten nur ein- oder ausschalten.
- Die komplette Rechenleistung des Suchszenarios muss vom MCAP erbracht werden.
- Eine intelligente Suchmethode wie SMART (Kap. 5.4.2) kann nicht eingesetzt werden. Die Suchergebnisse können nicht vor Ort gefiltert werden, sondern müssen immer komplett zum MCAP übertragen werden.

5.4.4 Bestellung und Bezahlung

Im Anschluss an die Suche wird in allen drei Szenarien ein Benutzeragent (BA) ausgesendet, um den Teilnehmer von den Suchergebnissen in Kenntnis zu setzen. Je nach Konfiguration und abhängig vom benutzten mobilen Endgerät kann der BA für die Ergebnisübersendung ein WML-Dokument, den SMS-Kurznachrichtendienst oder WAP-Push¹² [For99] einsetzen.

Möchte der Benutzer die Ware aufgrund des Suchergebnisses direkt bestellen, so kann die Bezahlung vollständig oder teilweise online oder offline geschehen. Mögliche, klassische Offline-Bezahlungsmechanismen sind z.B.:

- Lieferung auf Rechnung und Zahlung per Überweisung
- Zahlung per Nachnahme

Hybride Lösungen sind Verfahren, bei denen die Übermittlung zahlungsrelevanter Daten zum Händler elektronisch geschieht, z.B.:

- Bankeinzug
- Bezahlung über die Telefonrechnung

Für die Online-Bezahlung wurden die folgenden zwei Verfahren untersucht:

- SET (Secure Electronic Transaction)
- HBCI (Home-Banking Computer Interface)

Beide Verfahren lassen sich in Mobilfunknetzen lediglich mit der Hilfe eines Protokollumsetzers effizient betreiben. Eine für mobile Benutzer optimierte Lösung für HBCI wurde in Kapitel 5.3 mit dem AHB-Dienstes entworfen. Für SET stellt der sogenannte SET-Wallet Server [Lim98] eine interessante und leistungsstarke Lösung dar. Im Rahmen dieser Arbeit wird diese Konzeption aber nicht detaillierter untersucht.

5.5 Zusammenfassung

In diesem Kapitel wurden zunächst die Anforderungen an eine intelligente und effiziente Dienstangebotung mittels mobiler Agenten beschrieben. Dabei wurde herausgestellt, dass geeignete Agentensysteme insbesondere Anforderungen unterstützen sollten, die sich aus den Eigenschaften Benutzer- und Endgerätemobilität ergeben. Danach wurden die bei der Realisierung eingesetzten Softwareschnittstellen, Sicherheitsüberlegungen und Serverkomponenten der entworfenen agentenbasierten Kommunikationsplattform MCAP vorgestellt. Anschließend wurden zwei mit Hilfe des MCAPs und mobiler Agenten realisierte M-Commerce-Dienste ausführlich erläutert und deren Vor- und Nachteile diskutiert. Es wurde gezeigt, warum die im Rahmen der Arbeit untersuchten und umgesetzten Verbesserungen speziell für mobile (nomadische) Telekommunikationsteilnehmer notwendig und interessant sind.

¹²WAP-Push wurde im WAP-Standard 1.2 spezifiziert. Es erlaubt die Übertragung eines WML-Dokuments auch dann, wenn im Moment keine WAP-Verbindung aktiv ist.

Modellierung und Analyse des agentenbasierten Kommunikationskonzeptes

Die vorausgegangenen Kapitel haben qualitativ gezeigt, dass mobile Agenten (MAs) eine intelligente und flexible Dienstleistung ermöglichen, die herkömmliche Dienstleistungsmodelle ergänzt und verbessert. MAs helfen dabei, die Schwachstellen in Kommunikationsnetzen mit Funkteilstrecken zu verringern. Dennoch wird es Situationen und Dienste geben, für die andere Verfahren, wie z.B. RPC (Kap. 3.3) besser geeignet sind. Es wird häufig argumentiert, dass die wahre Stärke von MAs darin liegt, dass eine Reduktion der zu übermittelnden Kommunikationsdaten und damit eine Verringerung der anfallenden Kosten und der zur Durchführung erforderlichen Zeit erreicht werden kann. Dies wird in diesem Kapitel anhand der Ergebnisse der Leistungsbewertung untersucht. Mit Hilfe des dafür entwickelten analytischen Modells für die Agentenmigration werden die Kommunikationsplattform MCAP und die beiden agentenbasierten Dienste (AHB und ASB) im Hinblick auf das Datenvolumen, die Kommunikationszeit und die Kommunikationskosten bewertet. Obwohl sich viele Wissenschaftler in den vergangenen Jahren mit der Entwicklung von Agentensystemen und deren Diensten beschäftigt haben, existieren nur wenige Leistungsvergleiche. Diese Leistungsvergleiche [IH99, BER01] berücksichtigen jedoch nicht die einschränkenden Rahmenbedingungen von Mobilfunksystemen. Das Ziel dieser Leistungsbewertung ist es daher vor allem, durch einen Leistungsvergleich zwischen der traditionellen und agentenbasierten Dienstleistung in Mobilfunknetzen den möglichen Gewinn durch MAs zu ermitteln, und existierende Schwachpunkte und Erweiterungsmöglichkeiten für die Zukunft aufzuzeigen.

6.1 Ziele der Analyse

Mathematische Modelle sind sinnvoll, um Kommunikationsabläufe, z.B. hinsichtlich ihrer Auswirkungen auf die Netzlast, zu untersuchen. Softwareagenten besitzen verschiedene Fähigkeiten (Migration, Interaktion und Selektion), die es mathematisch zu erfassen gilt. In dieser Arbeit werden MAs hauptsächlich bezüglich ihrer Auswirkungen auf die Datenmenge B , die Zeit t , die zur Ausführung eines Kommunikationsdienstes benötigt wird, und die Kosten C , die beim Teilnehmer für die Nutzung des Dienstes anfallen, untersucht und bewertet. Eine Übersichtstabelle aller verwendeten Formelzeichen befindet sich im Anhang dieser Arbeit.

6.1.1 Datenvolumen

Betreiber von Kommunikationsnetzen müssen aus Gründen der Wirtschaftlichkeit ihre technischen Ressourcen, d.h. vor allem ihre Netzkapazität, so effizient wie möglich einsetzen. Dies gilt insbesondere für die Anbieter von öffentlichen Mobilfunknetzen und die von ihnen gegen eine nicht unerhebliche Lizenzgebühr erworbenen, lediglich begrenzt verfügbaren Funkkapazitäten. Aber auch für den Endkonsumenten wird in Zukunft die übertragene Datenmenge von immer wichtigerer Bedeutung sein, denn während sich die Kosten für Telekommunikationsteilnehmer bisher fast ausschließlich aus der Kommunikationszeit berechnet haben, wird bei paketorientierten Mobilfunksystemen wie GPRS (Kap. 2.3)

und UMTS auch die Datenmenge B zur Kostenberechnung herangezogen. Mit dem Formelbuchstaben B wird in dieser Arbeit die Menge der digitalen Daten bezeichnet, während die erreichten Daten- bzw. Übertragungsraten mit b benannt werden.

6.1.2 Kommunikationszeit

In den vergangenen Jahren hat die Nachfrage nach Mobilfunkdiensten stetig zugenommen. Dabei stieg nicht nur die Anzahl der Teilnehmer und der Kommunikationsverbindungen pro Teilnehmer kontinuierlich an, sondern auch die Kommunikationsdauer pro Verbindung hat sich, auch aufgrund der immer populärer werdenden Datenverbindungen, weiter erhöht. Dies ist hauptsächlich in der vermehrten Nutzung des Internets begründet. Insbesondere mobile Teilnehmer wünschen sich jedoch, dass z.B. die Ergebnisse einer Suchanfrage so schnell wie möglich zurück an die Mobilstation übermittelt werden, um die notwendige Kommunikationszeit t und damit auch die Wartezeit gering zu halten. In der vorliegenden Arbeit wird hauptsächlich mit Mittelwerten gearbeitet, die aus Gründen der Überschaubarkeit durch große Formelbuchstaben, z.B. für die Zeit mit T , wiedergegeben werden. Prinzipiell werden in dieser Arbeit drei Kommunikationszeiten unterschieden, siehe Abb. 6.1:

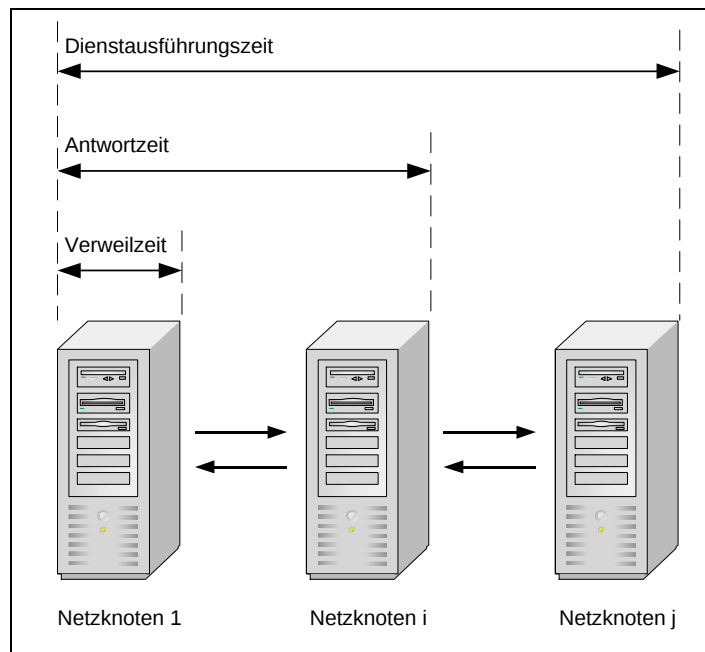


Abbildung 6.1: Kommunikationszeiten

Die **Verweilzeit** (*sojourn time*) beschreibt die Aufenthaltszeit für einen Auftrag in einem Netzknoten.

Die **Antwortzeit** (*response time*) ist die benötigte Zeit für eine bestimmte Dienstanfrage. Sie beinhaltet in der Regel mehrere Verweilzeiten und Übertragungsverzögerungen (*propagation delays*).

Die **Dienstaussführungszeit** (*service time*) ist die Zeit, die der Teilnehmer bei der Nutzung eines bestimmten Kommunikationsdienstes erfährt. Sie umfasst meist mehrere Antwortzeiten, sowie Zeiten für die Dateneingabe des Teilnehmers (*user interactions*).

6.1.3 Kommunikationskosten

Die Ziele eines Mobilfunkteilnehmers im Hinblick auf die Kommunikationskosten C sind eng verknüpft mit Überlegungen zur Dienstausführungszeit, da die Kosten bei den Tarifierungsmodellen der Telekommunikationsanbieter in der Regel proportional zur Onlinezeit sind. Aber auch für den Dienstanbieter spielen die Kommunikationskosten eine entscheidende Rolle, da bereits im Vorfeld einer Dienst Einführung darüber Klarheit herrschen sollte, zu welchen Rahmenbedingungen man neue Dienste anbieten kann. Speziell für Agentensysteme wurden im Rahmen dieser Arbeit solche Untersuchungen erstmalig vorgenommen und umfassende Ergebnisse gewonnen.

6.2 Prinzipien der verkehrstheoretischen Modellierung

Um Aussagen über die Kommunikationszeiten bei der vermehrten Nutzung von MAs für M-Commerce-Dienste vornehmen zu können, wird ein verkehrstheoretisches Modell entwickelt. Dabei kann allgemein davon ausgegangen werden, dass Prozesse in Kommunikationsnetzen mit Hilfe von zufallsabhängigen Aufträgen¹ der Teilnehmer an die zur Verfügung stehenden Betriebsmittel beschrieben werden können.

6.2.1 Warteschlangenmodelle

Warteschlangenmodelle sind ein geeignetes und beliebtes Instrument zur analytischen Bewertung von Kommunikationsnetzen [Gun96, Geu98]. Hierbei wird jeder Netzknoten, der bei der Erbringung eines Dienstes involviert ist, durch ein passendes, analytisches Warteschlangenmodell nachgebildet. Da generell jederzeit ein neuer Auftrag (z.B. eine Dienst anforderung) in einem Netzknoten eingehen kann, werden die Aufträge bei jedem Bearbeiter (Prozessor) in Warteschlangen abgelegt, siehe Abb. 6.2. Mit Hilfe der Gleichungen und Modelle der Verkehrstheorie (oft auch als Warteschlangentheorie bezeichnet) kann nun unter gewissen Annahmen über Verteilungen und Abfertigungsstrategien errechnet werden, welche zeitliche Verteilung für jeden Auftrag zu erwarten ist [Bol89, Wal93b]. Die einzelnen Zeiten und Verteilungen können zu einer Gesamtzeit aufaddiert werden, die eine Vorhersage über die auftretende Gesamtverzögerung darstellt [Jai91]. Für Systeme im stabilen Zustand, d.h. nachdem der Einschwingvorgang beendet ist, können mit Hilfe eines Warteschlangenmodells Ergebnisse berechnet werden.

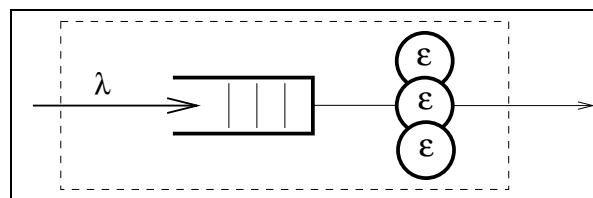


Abbildung 6.2: Warteschlange

Dabei werden die Warteschlangenmodelle durch die sogenannte *Kendall'sche Notation* beschrieben [Wal93b]:

$$a/b/c - d/e/f$$

¹Die Übermittlung/Übernahme einer Anweisung heißt Auftrag (*task*). Ein Auftrag beginnt mit der Übermittlung/Übernahme und endet mit der Fertigstellung der durch die Anweisung festgelegten Handlung [Wal93a].

Die ersten beiden Parameter (a und b) der Kendall'schen Notation beschreiben dabei den Ankunfts- und den Bedienprozess des Modells. Mögliche Verteilungsfunktionen für kontinuierliche Zufallsvariablen sind in Tab. 6.1 aufgelistet.

Symbol	Beschreibung
M	Markoff'sche (negativ-exponentielle) Verteilungsfunktion (Poissonprozess)
E_k	Erlang-k Verteilungsfunktion
H_k	hyperexponentielle Verteilungsfunktion
D	konstante (deterministische) Verteilungsfunktion
G	allgemeine (generelle) Verteilungsfunktion (keine Annahmen)

Tabelle 6.1: Verteilungsfunktionen der Ereignisabstände

Mit dem Parameter c der Kendall'schen Notation wird die Anzahl der Bediener angegeben, während d die Anzahl der Wartplätze in der Warteschlange nennt. Für Warteschlangen stehen verschiedene Abfertigungsstrategien e zur Verfügung, die in der Tab. 6.2 beschrieben sind. Mit Hilfe des Parameters f der Kendall'schen Notation können unterschiedliche Prioritätsverfahren angegeben werden. Häufig werden jedoch nur Ankunfts- und Bedienprozess, sowie die Anzahl der Bediener angegeben. Markoff'sche Ankunfts- und Bedienprozesse sind besonders wichtig und mathematisch vergleichsweise einfach zu behandeln. Viele Ankunfts- und Bedienprozesse lassen sich damit exakt oder näherungsweise beschreiben [Bol89].

Abkürzung	Bedeutung	Beschreibung
FCFS	First Come First Serve	Die Aufträge werden in der Reihenfolge bearbeitet, in der sie im System abgelegt wurden.
LCFS	Last Come First Serve	Der letzte Auftrag wird als erster bearbeitet.
SIRO	Service In Random Order	Die Aufträge werden in zufälliger Reihenfolge bearbeitet.
RR	Round Robin	Jeder Auftrag wird für eine bestimmte Zeit bearbeitet. Ist er danach noch nicht abgeschlossen, wird er erneut in der Warteschlange abgelegt.

Tabelle 6.2: Abfertigungsstrategien von Warteschlangen

Nachfolgend werden die wichtigsten Kenngrößen bei einer Leistungsbewertung mit Hilfe der Verkehrstheorie vorgestellt [Bol89, Kin90, Wal93b]. Die in der Analyse und Bewertung verwendeten Formelzeichen können in ihrer Gesamtheit auch dem Verzeichnis der Formelzeichen im Anhang dieser Arbeit entnommen werden.

- **Mittlere Ankunftsrate** λ (*mean traffic arrival rate*)
Anzahl der im Mittel pro Zeiteinheit ankommenden Aufträge, $[\lambda] = 1/s$
- **Mittlere Bedienrate einer Station** μ (*mean processing rate*)
Die Rate mit der Aufträge in einer Station, die einen oder mehrere Prozesse enthält,

bearbeitet werden $[\mu] = 1/s$

Die mittlere Bedienrate kann aus der mittleren Dienstrate ϵ (*service rate*) errechnet werden, wobei ϵ die Bedienrate der als gleichartig angenommenen Prozessoren einer Station ist:

$$\mu = n * \epsilon \quad (6.1)$$

Dabei gibt n die Anzahl der Prozessoren im System an.

- **Mittlere Bediendauer β** (*mean processing time*)

Die Bediendauer hängt vom Auftrag und vom Rechnertyp ab. Die mittlere Bediendauer ist der reziproke Wert der Bedienrate μ , $[\beta] = s$

$$\beta = 1/\mu \quad (6.2)$$

- **Auslastung ρ** (*traffic load*)

Die Auslastung beschreibt den Prozentsatz der Zeit während dem eine Station belegt ist und kann wie folgt berechnet werden:

$$\rho = \frac{\lambda}{n * \epsilon} = \frac{\lambda}{\mu} \quad (6.3)$$

In einem eingeschwungenen System gilt (Gleichgewichtsbedingung):

$$\rho < 1 \quad (6.4)$$

- **Mittlere Wartezeit W** (*mean queuing time*)

Die durchschnittliche Wartezeit eines Auftrages in der Warteschlange, $[W] = s$

- **Mittlere Verweilzeit V** (*mean sojourn time*)

Als mittlere Verweilzeit bezeichnet man den Erwartungswert der Zeit, den ein Auftrag im System verweilt:

$$V = W + \beta \quad (6.5)$$

- **Mittlere Anzahl von Aufträgen im System L** (*mean number of jobs*)

Die mittlere Anzahl von Aufträgen kann mit Hilfe von *Little's Theorem*, das für beliebige G/G/n-Systeme und alle Abfertigungsstrategien gilt [Bol89], berechnet werden:

$$L = \lambda * V \quad (6.6)$$

Eine zentrale Gleichung für Warteschlangen mit Markoff'schem Ankunftsprozess ist die *Pollaczek-Khinchine-Mittelwertformel*, mit der die mittlere Anzahl von Aufträgen L im System berechnet werden kann:

$$L = \rho + \rho^2 \frac{1 + c_b^2}{2(1 - \rho)} \quad (6.7)$$

Dabei ist c_b der Variationskoeffizient der Bediendauer. Der Variationskoeffizient kann aus dem zweiten zentralen Moment, der sogenannten Varianz, gewonnen werden. Das n -te zentrale Moment einer kontinuierlichen Zufallsvariable X lässt sich allgemein berechnen, wobei $\bar{X}$ das erste Moment (Erwartungswert) ist:

$$\overline{(X - \bar{X})^n} = E[(X - E[X])^n] = \int_{-\infty}^{\infty} (x - \bar{X})^n f_X(x) dx \quad (6.8)$$

Für das zweite zentrale Moment, die Varianz $var(X)$, gilt daher:

$$var(X) = \sigma_X^2 = \overline{(X - \bar{X})^2} = \bar{X^2} - \bar{X}^2 \quad (6.9)$$

Aus der Standardabweichung σ_X kann nun die normierte Standardabweichung, der Variationskoeffizient c_X , errechnet werden:

$$c_X = \frac{\sigma_X}{\bar{X}} \quad (6.10)$$

Für den Variationskoeffizienten c_b der Bediendauer ergibt sich daher:

$$c_b = \frac{\sigma_b}{\beta} \quad (6.11)$$

Während sich für M/D/1 Warteschlangen ein Variationskoeffizient $c_b = 0$ ergibt, da die Standardabweichung $\sigma_b = 0$ ist, gilt für M/M/1 Warteschlangen mit negativ-exponentiell verteilter Bediendauer $c_b = 1$.

Die Pollaczek-Khinchine-Mittelwertformel, Gleichung 6.7, kann mit Hilfe von *Little's Theorem*, Gleichung 6.6, umgestellt werden zu:

$$V = \frac{1}{\mu} \left\{ 1 + \rho \frac{1 + c_b^2}{2(1 - \rho)} \right\} \quad (6.12)$$

6.2.2 Dekompositionsverfahren

Mit Hilfe von Modellen kann die Realität abstrahiert werden. Diese Modelle sind dann eine mathematische Präsentation der Realität. Werden verschiedene Warteschlangenmodelle (Teilsysteme) miteinander gekoppelt und können dabei auch Aufträge von außen in das Gesamtsystem gelangen, dann spricht man von einem offenen Wartenetz. Die Vielzahl der unabhängigen Nachrichtenflüsse in einem Telekommunikationsnetz wird überlagert und führt, den einzelnen Knoten betrachtend, zu einem die Markoffbedingung erfüllenden Ankunftsprozess. Möchte man jedoch ein vollständiges Modell für die komplette Dienstleistung in einem Telekommunikationsnetz betrachten, das alle bei der Dienstleistung involvierten Netzknoten und Protokollschichten widerspiegelt, so stellt man fest, dass das entstehende offene Wartenetz für eine genaue mathematische Analyse aufgrund

seiner Komplexität ungeeignet ist [BK93]. In dieser Arbeit wird daher die Technik des Dekompositionsverfahrens verwendet [Bol89]. Beim Dekompositionsverfahren wird generell das Gesamtsystem (offenes Wartenetz) in mehrere Teilsysteme zerlegt, um die Komplexität zu verringern. Wenn diese Teilsysteme hauptsächlich durch lokale Einflüsse beeinträchtigt werden, so kann man die mit Hilfe des Dekompositionsverfahrens gewonnenen Einzelergebnisse zu einer Lösung für das Gesamtsystem zusammenfassen und erhält für Produktformnetze² exakte und für Nichtproduktformnetze approximative Ergebnisse. So lässt sich die mittlere Dienstaussführungszeit S gemäß Gleichung 6.13 als Summe der mittleren Verweilzeiten V bzw. der mittleren Antwortzeiten T eines Dienstes x über alle (n bzw. m) bei der Diensterbringung beteiligten Knoten (Teilsysteme) ermitteln:

$$S_x = \sum_{i=0}^n V_{x,i} = \sum_{j=0}^m T_{x,j} \quad i, j, n, m \in \mathbb{N} \quad (6.13)$$

6.2.3 Werkzeuge für die Analyse

Die mathematische Analyse der beiden agentenbasierten Dienste wurde mit Hilfe der Werkzeuge MATLAB, GATE und ETHEREAL durchgeführt. Während MATLAB Berechnungen komplexer Matrizen und deren Visualisierung ermöglicht, kann mit GATE eine GPRS-Kommunikationsverbindung simuliert und mit Hilfe von ETHEREAL die Paketgrößen der unterschiedlichen Protokolle ermittelt werden.

MATLAB

Die Software *MATLAB*³ des Unternehmens MathWorks Inc. [Mat01] erlaubt umfangreiche numerische Berechnungen basierend auf der Matrizenrechnung, deren Ergebnisse mit Hilfe eines speziellen Visualisierungswerkzeuges ansprechend dargestellt werden können. Hierzu stellt MATLAB eine entsprechende einfache Programmiersprache zur Verfügung, die für den Einsatz in technischen Projekten entwickelt wurde und für Ingenieure und Wissenschaftler relativ leicht zu erlernen ist. Alle Graphen in dieser Arbeit wurden mit Hilfe von MATLAB erstellt.

GATE

GPRS Application Test Environment (GATE) ist ein Werkzeug zum Testen von GPRS-Anwendungen, welches vom Unternehmen Ericsson entwickelt wurde [Eri00]. Eine Kommunikation über GPRS unterscheidet sich erheblich von herkömmlichen LAN-Verbindungen. Zum einen handelt es sich bei GPRS um ein Mobilfunksystem, d.h. mit Interferenzen und damit verbundenen Datenfehlern und Datenverlusten muss gerechnet werden. Ferner ist GPRS ein paketorientiertes System, in dem sich die Teilnehmer die zur Verfügung stehenden Betriebsmittel teilen. GATE simuliert eine GPRS-Umgebung, bei der sich die folgenden Parameter einstellen lassen:

- Die Anzahl der zur Verfügung stehenden Zeitschlitze während einer Verbindung,
- der C/I-Wert⁴, mit dem der Signalstörabstand beschrieben wird,

²Warteschlangennetze, für die eine Lösung der lokalen Gleichgewichtsbedingung (*local balance*) existiert, nennt man Produktformnetze (oder BCMP-Netze).

³Abkürzung für Matrix Laboratory

⁴Carrier-to-Interference

- die Anzahl der von der Mobilstation genutzten Zeitschlitzte,
- und die Anzahl der gleichzeitigen Benutzer.

GATE ermöglicht die Simulation von GPRS-Teilnehmern, die einen direkten Zugang zu einem Internet Service Provider (ISP) haben. Die GATE-Software muss dafür auf einem PC installiert sein, der mit zwei Netzwerkkarten arbeitet. Bei den durchgeführten Messungen war eine Karte mit dem simulierten Mobiltelefon, die zweite mit der Kommunikationsplattform MCAP, verbunden. Abb. 6.3 zeigt den entsprechenden Messaufbau.

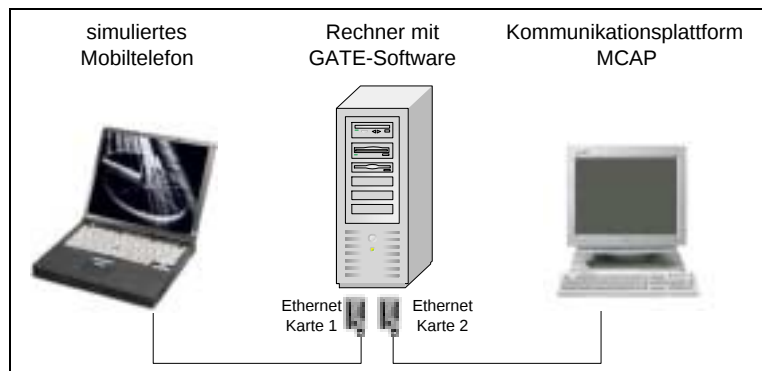


Abbildung 6.3: Messungen mit GATE

ETHERREAL

Weiterhin wurden im Rahmen dieser Arbeit Datenmessungen bei den agentenbasierten AHB- und ASB-Diensten mit Hilfe der Software ETHEREAL in der Version 0.18.8 durchgeführt. ETHEREAL ist ein freies Protokollanalyseprogramm⁵ für UNIX und Windows, welches die Untersuchung und Überwachung von HTTP, WSP, WTP, TCP, UDP, IP und Ethernet-Paketen erlaubt. ETHEREAL wurde auch eingesetzt, um die Verzögerungen durch das GPRS-Mobilfunknetz, welche durch GATE simuliert wurden, zu messen.

6.3 Verkehrsmodelle der Netzknoten

Ein reales Telekommunikationsnetz beinhaltet eine Vielzahl von Vermittlungsstellen, Signalisierungsknoten, verschiedenste Übertragungskanäle und unterschiedlichste Mechanismen zur Übermittlung, Segmentierung, und Wiederausammensetzung von Nachrichten. Die exakte Analyse eines derart großen und komplexen Systems kann gut durch das Dekompositionsverfahren angenähert werden, indem man das System in mehrere Teilsysteme zerlegt und die Ergebnisse aus den Teilsystemen nachher wieder zusammensetzt. Zur Untersuchung der beiden realisierten agentenbasierten M-Commerce-Dienste (AHB und ASB) werden die folgenden Teilsysteme betrachtet:

1. das GSM-Mobilfunknetz,
2. der WAP-Gateway,
3. die Kommunikationsplattform MCAP,
4. das Internet

⁵ Ethereal ist ein Freeware-Produkt, welches unter <http://www.ethereal.org/> verfügbar ist

6.3.1 Mobilfunknetz

Für die Leistungsbewertung von MAs in mobilen Kommunikationsnetzen wurden zwei unterschiedliche Übertragungsmechanismen des GSM-Standards, GSM-Datenrufe und GPRS, untersucht und modelliert.

GSM-Datendienst

Beide agentenbasierten M-Commerce-Dienste können über herkömmliche GSM-Datenverbindungen durchgeführt werden. Die für die Netzsicht angebotene Übertragungsrate des GSM-Datendienstes b_{gsm} von 9.600 bit/s ist zwar verglichen mit heutigen Festnetzübertragungsraten (z.B. V.90 Modem, ISDN) relativ gering, doch für die beiden auf zellulare Mobilfunksysteme zugeschnittenen M-Commerce-Dienste durchaus ausreichend. Die leitungsvermittelte Datenübertragung über GSM (CSD⁶) unterscheidet sich vom gewöhnlichen GSM-Telefoniedienst hauptsächlich in der Kanalkodierung. Der ausgewählte Trägerdienst (Kap. 2.1) nutzt den nicht-transparenten Modus⁷ (*non-transparent mode*), d.h. dass neben der sogenannten Vorwärtskorrektur durch das FEC⁸-Verfahren RLP (*radio link protocol*) zur Fehlerfeststellung und -korrektur eingesetzt wird. Eine mit RLP hergestellte Verbindung ist effizient und sicher, da im Fehlerfall nur das kleine, fehlerbehaftete Datenpaket zurückgesendet wird. RLP gewährleistet eine gewisse Zuverlässigkeit der empfangenen Daten, verhindert jedoch eine konstante Übertragungsrate und Verzögerung. Vielmehr hängen die Übertragungsrate und die Verzögerung in diesem Modus direkt von der Qualität der Funkverbindung ab. Abb. 6.4 zeigt, wie die Datenrate zustande kommt, die CSD dem Verbindungsprotokoll PPP⁹ (*point-to-point-protocol*) anbietet.

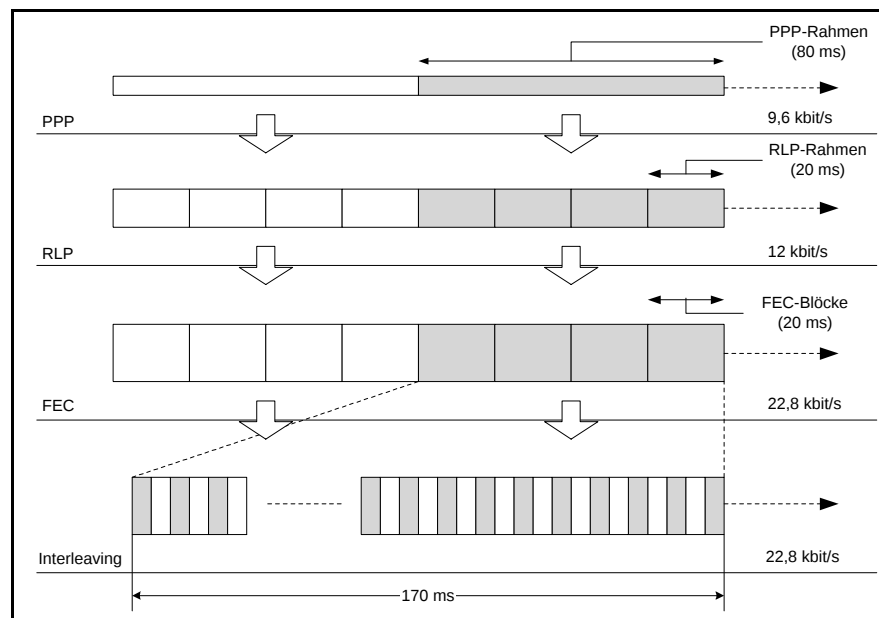


Abbildung 6.4: CSD-Datenraten

⁶circuit-switched data

⁷neben dem nicht-transparenten Modus bietet GSM auch transparente Datendienste, die lediglich ISO/OSI-Schicht 1 Funktionen anbieten

⁸forward error correction

⁹PPP ist ein optimiertes Einwahlprotokoll für die IP-Datenübertragung mittels Modem und Telefonleitung

Sowohl der AHB-Dienst als auch der ASB-Dienst werden im Falle einer CSD-Datenverbindung über PPP abgewickelt. Abb. 6.5 zeigt den prinzipiellen Ablauf einer TCP/IP- (AHB-Dienst, links) und einer WAP-Verbindung (ASB-Dienst, rechts) über CSD. Zunächst wird ein Datenkanal etabliert, danach wird eine PPP-Verbindung aufgebaut. Im Falle des AHB-Dienstes wird daraufhin eine TCP-Verbindung etabliert und optional eine SSL-Sitzung eingerichtet. Bei der verbindungsorientierten WAP-Kommunikation (Kap. 2.2.5) des ASB-Dienstes wird nach dem Aufbau der PPP-Verbindung eine WSP-Verbindung etabliert. Im verbindungslosen Fall werden direkt UDP-Pakete versendet.

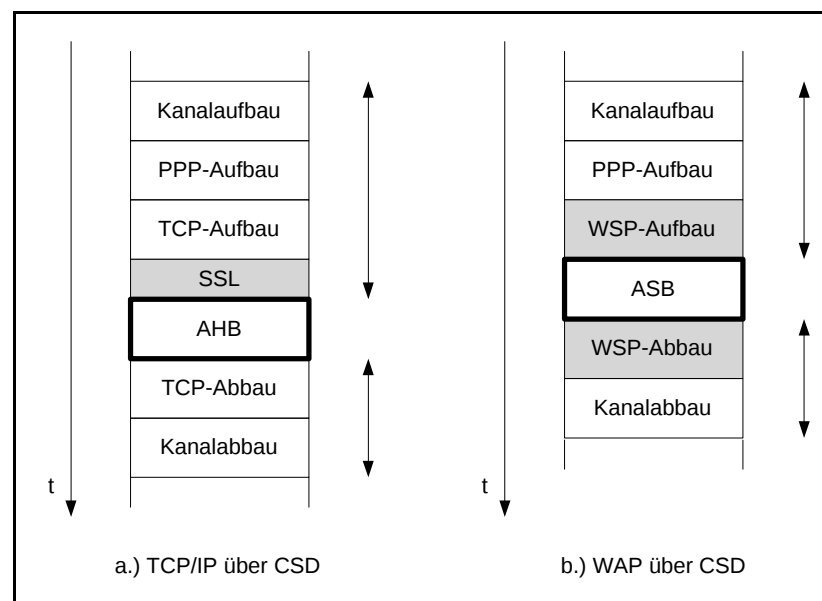


Abbildung 6.5: Verbindungsauf- und abbauverzögerungen bei CSD

Die **Transaktionsleistung** b_{csd} beschreibt den mittleren Durchsatz über CSD bei der Blockdatenübertragung für Nutzdaten der Transportschicht bestimmter Transaktionen unter normalen Alltagsbedingungen.

Die Transaktionsleistung der CSD-Verbindung berücksichtigt neben den Kopfinformationen (*header information*) der unterliegenden Protokollschichten auch Laufzeitverzögerungen einzelner Nachrichten aufgrund deren langen Umlaufzeiten. Die Transaktionsleistung liegt daher deutlich unter der Übertragungsrate der GSM-Datenrufe b_{gsm} von 9,6 kbit/s (1.200 byte/s). Dabei schneidet die Transaktionsleistung über TCP/IP im Gegensatz zu UDP/IP besser ab, da PPP lediglich für TCP/IP-Kopfinformation einen Kompressionsmechanismus zur Verfügung stellt [Jac90], wobei die TCP-Kopfinformation durch den Van-Jacobsen-Algorithmus von 40 byte auf 3 bis 6 byte pro TCP/IP-Paket komprimiert wird.

Die gemessenen Transaktionsleistungen b_{csd} für die Agentenmigration (4 bis 9 kbyte Nutzdaten) und für WAP-Nachrichten (50 bis 1.500 byte Nutzdaten) können Tab. 6.3 entnommen werden. Diese Werte wurden durch vielfältige Messungen gewonnen und sind in verschiedenen Arbeiten verifiziert [Bau97, Son98, Eve99, Hüc00, Elm01] worden. Die Transaktionsleistung war bei Messungen in den unterschiedlichsten geographischen Lagen¹⁰

¹⁰In den Forschungslaboren des Unternehmens Ericsson wurden in verschiedenen Ländern Europas und Asiens, sowohl in städtischen als auch in ländlichen Gebieten, Messungen durchgeführt.

relativ konstant, was zeigt, dass die Mechanismen zur Fehlerkorrektur des GSM-Systems sehr stabil sind und daher Störungen auf der Transportschicht kaum aufgetreten.

Transaktion	Transaktionsleistung b_{csd}
Agentenmigration (TCP)	882 <i>byte/s</i>
WAP-Nachrichten (UDP)	279 <i>byte/s</i>

Tabelle 6.3: Transaktionsleistung bei CSD

Die CSD-Antwortzeiten für den Auf- und Abbau der TCP/IP- und der WAP-Datenverbindung wurden ebenfalls mittels Messungen untersucht. Die mittleren Antwortzeiten $T_{cs, tcp}$ und $T_{cs, wap}$ sind in Tab. 6.4 gemeinsam mit den entsprechenden Varianzen und Standardabweichungen für eine Einwahl über ein analoges Modem angegeben. Wird für die Einwahl eine digitale ISDN-Verbindung verwendet, so können die Verbindungsauf- und abbauverzögerungen aufgrund des schnelleren Kanalaufbaus um bis zu 10 Sekunden verringert werden. Der Aufbau einer SSL-Verbindung hingegen verlängert die Antwortzeit T_{cs} um rund 9 Sekunden. Für die WAP-Kommunikation wird nicht zwischen verbindungsloser und verbindungsorientierter WSP-Kommunikation unterschieden, da die zeitlichen Differenzen für die versendeten WML-Dokumente nicht signifikant sind und herkömmliche WAP-Mobiltelefone meist nur die verbindungsorientierte Kommunikation über CSD erlauben.

Verbindungstyp	T_{cs}	$var(t_{cs})$	$\sigma(t_{cs})$
TCP/IP über CSD	26,01 <i>s</i>	8,83 <i>s</i> ²	2,97 <i>s</i>
WAP über CSD	17,61 <i>s</i>	19,05 <i>s</i> ²	4,36 <i>s</i>

Tabelle 6.4: Antwortzeit T_{cs} bei CSD

GPRS

Es wird oft behauptet, dass agentenbasierte Kommunikationskonzepte ideal für paketorientierte Netze geeignet sind. Daher wurden in dieser Arbeit der AHB- und der ASB-Dienst auch für das paketorientierte Kommunikationsnetz GPRS untersucht. Dabei musste ein geeignetes mathematisches Modell mit einer Reihe von Annahmen entworfen werden, da die verfügbaren Funkkapazitäten und damit auch die erreichbare Datenrate von GPRS von einer Vielzahl von Faktoren abhängen und daher nur schwer voraussagbar sind. Den getroffenen Annahmen für das Verkehrsmodell einer GPRS-Mobilfunkstrecke liegen jedoch Erkenntnisse über die reale Implementierung von GPRS zu Grunde [Eri01]. Bezüglich der Dienstgüteparameter von GPRS (Kap. 2.3.5) wurden daher die folgenden Annahmen getroffen:

- (1) **Priorität** Da unklar ist, bei welcher Verkehrslast das Mobilfunknetz anfängt Pakete zu löschen und nach welcher Methode die zu löschenden Pakete ausgewählt werden, werden keine Prioritäten berücksichtigt. Die ermittelten Werte würden sich ohnehin nur geringfügig unterscheiden.

- (2) **Verzögerung** Auch die vier verschiedenen Verzögerungsklassen des GPRS-Mobilfunkstandards werden vernachlässigt, da ihre Verwendung bei den untersuchten Diensten nicht sinnvoll erscheint.
- (3) **Zuverlässigkeit** Der Einfluss der Zuverlässigkeitsparameter wird ebenfalls nicht berücksichtigt. Datenpakete mit einer hohen Zuverlässigkeit gehen nur selten verloren bzw. sind nur selten inhaltlich verändert. Gleichzeitig wird die Datengröße aufgrund der zusätzlichen Fehlerkorrekturmethode nicht unwesentlich ansteigen. Es besteht daher der Grund zur Annahme, dass sich die beiden Einflüsse gegenseitig kompensieren.

Die weiteren drei Dienstgüteparameter eines GPRS-Netzes (Kap. 2.3), d.h. (4) Maximaler Durchsatz, (5) Durchschnittlicher Durchsatz und (6) Funkpriorität, sind Bestandteil des verwendeten GPRS-Verkehrsmodells. Dabei wurden vier verschiedenen Übertragungsqualitäten unterschiedliche GPRS-Parameter zugeordnet:

- Best Case** Diese Qualität beschreibt den Fall, in dem der Benutzer alle acht zur Verfügung stehenden Zeitschlitz zur Übertragung nutzen kann und sonstige Störeinflüsse eher vernachlässigbar sind.
- Normal Case** Hier bemühen sich fünf Benutzer gleichzeitig um vier Zeitschlitz. Dieser Fall soll das durchschnittliche Systemverhalten bei einer GPRS-Übertragung beschreiben.
- Limited Case** Dieser Fall unterscheidet sich vom *Normal Case* insbesondere dadurch, dass zur Übertragung lediglich ein Zeitschlitz genutzt wird.
- Worst Case** Mit Hilfe der eingestellten Parameter soll betrachtet werden, wie sich das System bei temporärer Überlast verhält.

Die für die Simulation verwendeten Werte der Parameter können Tab. 6.5 ebenso entnommen werden, wie die entsprechenden Einstellungen der GATE-Software (Kap. 6.2.3). Die erreichte GPRS-Datenrate wird nachfolgend mit b_{gprs} bezeichnet.

GPRS-Verkehrsmodell	Best Case	Normal Case	Limited Case	Worst Case
(4) Max. Durchsatz-Klasse	7	5	3	1
(5) Durchschn. Durchsatz	10	7	5	2
(6) Funkpriorität-Klasse	1	2	3	4
GPRS-Zeitschlitz	8	4	2	2
C/I [dB]	18	15	15	13
Gleichzeitige Benutzer	0	5	3	10
Zeitschlitz Uplink	8	1	1	1
Zeitschlitz Downlink	8	4	1	1

Tabelle 6.5: GPRS-Verkehrsmodelle

Für die Bewertung des Nutzen von GPRS für den AHB-Dienst wird davon ausgegangen, dass der Transport von GPRS-Nachrichten im Durchschnitt über drei Netzknoten

erfolgt, die alle die gleiche Auslastung erfahren. Im mathematischen Verkehrsmodell wird jeder SGSN mit einem M/M/1-Warteschlangenmodell betrachtet, da über diesen Netzknoten sämtliche Nachrichten zum Internet für unterschiedlichste Dienste abgewickelt werden und daher die Nachrichtenlängen und auch die Bediendauern als negativ-exponentiell verteilt angenommen werden können [HEG⁺99]. Ein Markoff'scher Ankunftsprozess kann angenommen werden, da von vielen überlagerten, unabhängig ankommenden Aufträgen ausgegangen werden kann.

6.3.2 WAP-Gateway

Im Falle des ASB-Dienstes wird für die Kommunikation zwischen dem Agentenserver MCAP und dem mobilen Endgerät ein WAP-Gateway (Kap. 2.2.5) eingesetzt. Im Rahmen der Untersuchungen wurden verschiedene WAP-Gateways untersucht. Neben den kommerziellen WAP-Gateways der deutschen GSM-Netzbetreiber T-Mobil (IP-Adresse 193.254.160.002) und D2-Vodafone (010.001.032.010), wurde ein eigener WAP-Gateway mit der Kannel-Software¹¹ Version 0.10.2 im MCAP integriert. Der WAP-Gateway wird durch eine IP-Adresse identifiziert und setzt als Protokollumsetzer Nachrichten vom WAP-Protokollstapel auf den Internet-Protokollstapel um, siehe Abb. 6.6.

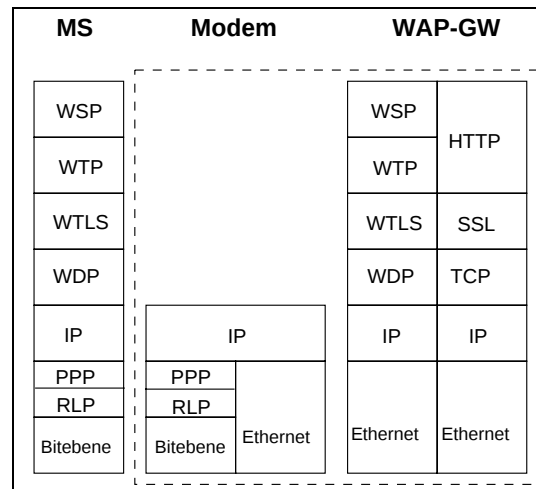


Abbildung 6.6: WAP-Kommunikation

Während WAP-Inhalte gewöhnlich in WML erstellt werden, können WAP-fähige Endgeräte lediglich die kodierte Variante von WML darstellen. Daher nimmt der WAP-Gateway neben der Protokollumsetzung auch die Transformation von WML nach *WAP Binary XML* (WBXML) und umgekehrt vor, siehe Abb. 6.7. Dabei wird jedes bekannte Tag¹² anhand einer standardisierten Kodierungstabelle zu einem ein byte langen Schlüssel reduziert. Die Kodierung tritt in beide Richtungen, d.h. vom mobilen Endgerät zum WAP-Gateway und umgekehrt, in Erscheinung. Die Effizienz dieser Kodierung hängt daher stark vom Verhältnis *Tags/Inhalt* (T/I) ab.

WML-Dateien mit umfangreichem Dokumentinhalt und wenigen Tags können mit der Standardkodierungstabelle nicht sehr effizient kodiert werden. Es besteht jedoch die Mög-

¹¹Kannel ist ein Freeware-Produkt, welches unter <http://www.kannel.3glab.org/> verfügbar ist.

¹²Tags sind Elemente, die eine eindeutige Anweisung beinhalten. In Aufzeichnungssprachen (*markup languages*) wie HTML, WML und XML werden Tags in der Syntax <Kennzeichen> verwendet.

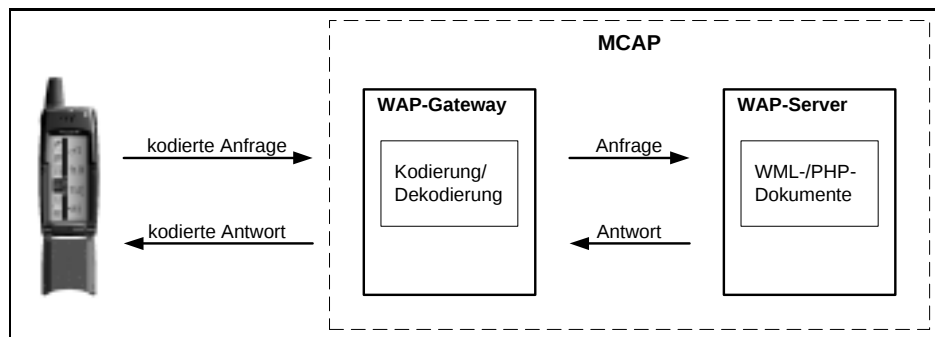


Abbildung 6.7: WAP-Gateway als Inhaltskodierer/-dekodierer

lichkeit, eine eigene Kodierungstabelle zu entwerfen. Da die Tabelle allerdings bei beiden Kommunikationspartnern, d.h. dem mobilen Endgerät und dem WAP-Gateway, bekannt sein muss, verringert die Verwendung von proprietären Kodierungstabellen die Flexibilität der WAP-Anwendung, da nur noch Endgeräte und WAP-Gateways benutzt werden können, die diese Tabelle kennen. Dies widerspricht der allgemeinen Zielsetzung des WAP-Standards nach einer weitgehend universellen Verwendung.

Die **Kodierungsrate** ξ ist ein Maß für die Reduktion des zu übertragenden Datenvolumens durch den WAP-Gateway.

In dieser Arbeit wird die Kodierungsrate ξ wie folgt definiert:

$$\xi = 1 - \frac{B_{wbxml}}{B_{wml}} \quad (6.14)$$

Dabei steht B_{wbxml} für die Nutzdaten (*payload*) des WSP-Protokolls. Diese Nutzdaten werden wie in Abb. 6.8 d) bzw. e) dargestellt für die verbindungslose bzw. verbindungsorientierte Kommunikation als IP-Pakete zwischen WAP-Gateway und mobilen Endgerät versendet. Beim Einsatz von WAP über CSD wird nachfolgend immer der verbindungsorientierte Dienst, d.h. WSP über WTP, eingesetzt, da die zeitlichen Differenzen für das Versenden von kleinen WML-Dokumenten nicht signifikant sind und herkömmliche WAP-Mobilfunktelefone meist nur die verbindungsorientierte Kommunikation über CSD erlauben. B_{wml} gibt die Nutzdaten des HTTP-Protokolls an, siehe Abb. 6.8 b), welches zwischen WAP-Gateway und WAP-Server eingesetzt wird. Es muss berücksichtigt werden, dass die Größe der Kopfinformationen bei HTTP sehr stark variiert und umfangreich ist. Zur Untersuchung der Effizienz der WBXML-Kodierung beim ASB-Dienst werden die Nutzdaten des WSP-Protokolls B_{WBXML} mit den Nutzdaten des HTTP-Protokolls B_{wml} in Kap. 6.6.2 verglichen.

Zunächst wurde die Leistungsfähigkeit des WAP-Gateways des Unternehmens Ericsson [Eri99a] mit Hilfe von Messungen überprüft. Dabei wurde festgestellt, dass die Bediendauer des WAP-Gateways β_{wapgw} in Abhängigkeit von der Größe der WBXML-Nachricht B_{wbxml} mit der linearen Gleichung 6.15 approximiert werden kann. Dabei wurde der Schnittpunkt mit der Ordinate θ zu 48 ms und die Steigung η mit $0,1064\text{ ms/byte}$ angenommen. Abb. 6.9 zeigt, dass mit Hilfe dieser Werte und der Gleichung 6.15 die Messungen gut abgebildet werden. Weiterhin muß berücksichtigt werden, dass Mobiltelefone aus Speicherplatzgründen meist nur WSP-Nachrichten bis 1.500 byte akzeptieren.

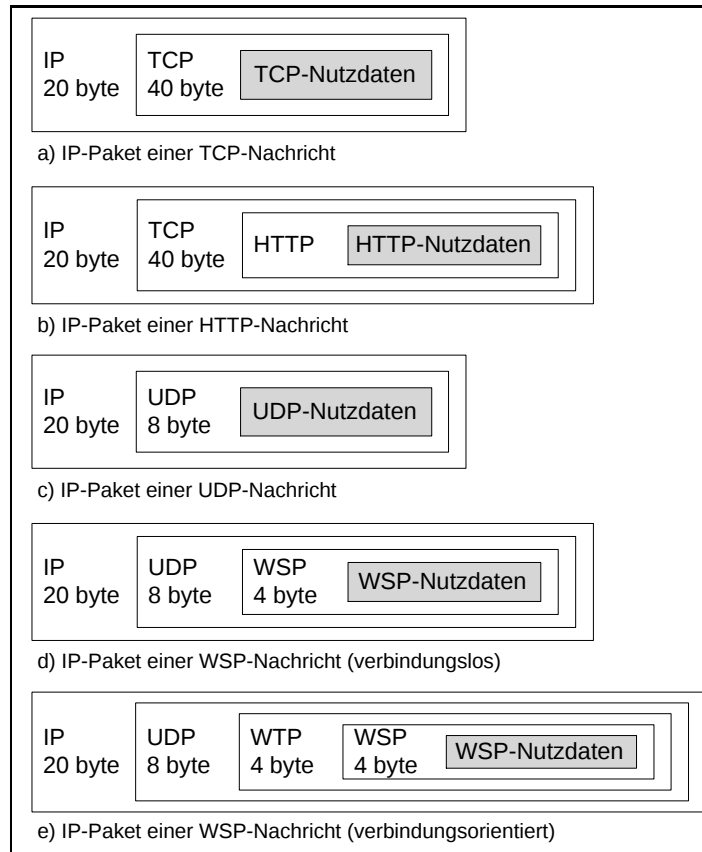


Abbildung 6.8: IP-Datenpakete

$$\beta_{wapgw} = \theta + \eta * B_{wbxml} \quad (6.15)$$

Zur Berechnung der Verweilzeit V_{wapgw} wird der WAP-Gateway durch ein M/M/1-Warteschlangenmodell abgebildet. Ein Markoff'scher Ankunftsprozess kann angenommen werden, da ein WAP-Gateway ein Internetknoten ist, bei dem unabhängige Aufträge von vielen Mobilfunkbenutzern unterschiedlicher Netzbetreiber eingehen. Ein Markoff'scher Bedienprozess liegt vor, da die Bediendauer durch die Größe der Nachrichten bestimmt wird, welche als negativ-exponentiell verteilt gemessen worden. Die mittlere Größe einer WBXML-Nachricht wurde mittels Messungen zu 362 *byte* ermittelt.

6.3.3 Agentenplattform für M-Commerce-Dienste

Die M-Commerce Agentenplattform (MCAP) unterstützt sowohl den agentenbasierten Bankdienst (AHB) als auch den Produktsuchdienst (ASB). Darüber hinaus sind noch eine Reihe von weiteren Diensten vorstellbar, die mit Hilfe des MCAPs angeboten werden können, wie z.B. Informationsdienste für Reisende, Ärzte, Handelsvertreter oder Logistikunternehmen. Der Betreiber einer derartigen Plattform sollte jedoch von Anfang an berücksichtigen, wieviele Benutzeranfragen bearbeitet werden können, um den MCAP den Bedürfnissen entsprechend korrekt zu dimensionieren, da zu hohe Verweilzeiten für die einzelnen Dienstanfragen zu inakzeptablen Dienstausführungszeiten führen können. Damit der Dienstanbieter seine Kunden behält und potentielle neue Kunden nicht verärgert, sollte er weiterhin dafür Sorge tragen, dass die angebotenen Dienste stabil und zuverlässig ausgeführt werden. Ferner möchte der Kunde die Kommunikationsverbindungen und

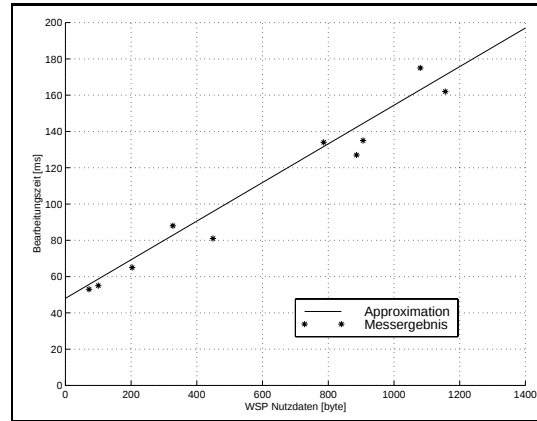


Abbildung 6.9: Antwortzeit des WAP-Gateways in Abhängigkeit der WSP-Nutzdaten

-dienste für einen angemessenen Preis erhalten. Nachfolgend wird daher der Einfluss der Dienstanfragen auf die durchschnittliche Verweilzeit untersucht.

Beim ASB-Dienst wird die Antwortzeit zwischen dem Eintreffen eines Auftrages beim MCAP und dem Zeitpunkt, an dem das Ergebnis den MCAP in Richtung WAP-Gateway verlässt als Produktsuchzeit t_{ps} (siehe auch Abb. 6.11) bezeichnet. Der MCAP, der im Falle des ASB-Dienstes seine Aufträge stets vom WAP-Gateway erhält, ist für die gesamte Koordination der Suchagenten verantwortlich. Zunächst wird mittels PHP und SQL eine Datenbankabfrage zu verfügbaren Händlern durchgeführt. Das entsprechende Ergebnis gibt der MCAP dann an einen mobilen (MSA) oder einen stationären (SSA) Suchagenten weiter.

In Tab. 6.6 werden die gemessenen Bediendauern für die drei verschiedenen Konzepte (Kap. 5.4.3) des ASB-Dienstes für die Kommunikationsplattform MCAP β_{mcap} und eine durchschnittliche Händlerplattform¹³ β_m angegeben.

	MSA	MSA+	SSA
β_{mcap}	298 ms	372 ms	24 ms
c_{mcap}	0,253	0,174	0,424
β_m	263 ms	288 ms	18 ms
c_m	0,242	0,139	0,501

Tabelle 6.6: Bediendauern und Variationskoeff. des MCAPs und der Händlerrechner

Tab. 6.6 ist zu entnehmen, dass die SQL-Anfragen (SSA), d.h. der RPC-Ansatz, deutlich schneller bearbeitet werden und daher auch eine geringere Rechenkapazität benötigen. Andererseits muss bedacht werden, dass aufgrund des RPC-basierten Nachrichtenaustausches der MCAP nach jeder Händleranfrage erneut aufgesucht wird, siehe Abb. 5.17. Daraus resultiert, dass der MCAP beim Einsatz von mobilen Suchagenten (MSA) lediglich zwei Aufträge (das Starten und Beenden des MSAs) bearbeiten muss, während beim RPC-Ansatz die Anzahl der Arbeitsprozesse proportional zur Anzahl der aufgesuchten Händler n_{vm} ist. D.h., dass im Fall MSA die Last des MCAPs unabhängig von der Händleranzahl

¹³die Werte basieren auf Messungen für einen LINUX-PC mit einem Intel Pentium 233 MHz Zentralprozessor und 128 MB Hauptspeicher

ist, da alle Aufträge und Filterprozesse lokal auf der Händlerplattform ausgeführt werden ohne dass eine Rücksprache mit dem MCAP notwendig wird. Für die MSAs mit XML-Schnittstelle (MSA+) wird ein erhöhter Rechenaufwand benötigt, da die XML-Dateien zunächst generiert und später übersetzt werden müssen.

Da der MCAP im Falle eines MSAs unabhängig von der Anzahl der aufgesuchten Handelsplattformen n_{vm} lediglich zwei Aufträge bearbeiten muss, ist die durch MSAs verursachte Prozessorlast annähernd konstant. Da der MCAP unabhängige Aufträge über verschiedene WAP-Gateways erhalten kann, wird ein Markoff'scher Ankunftsprozess angenommen. Weiterhin gibt es nur einen Bediener, so dass ein M/G/1 Warteschlangenmodell angewendet und die Gleichung 6.12 als Basis für die Berechnung der durchschnittlichen Verweilzeit V_{mcap} mit Hilfe der Formel 6.16 verwendet wird.

$$V_{mcap,msa} = 2 * \beta_{mcap,msa} \left\{ 1 + \rho \frac{1 + c_{mcap,msa}^2}{2(1 - \rho_{mcap})} \right\} \quad (6.16)$$

Beim SSA muss der MCAP sämtliche SQL-Anfragen bearbeiten. Dabei wird ebenfalls ein M/G/1 Warteschlangenmodell den Berechnungen zu Grunde gelegt. Die Verweilzeit der stationären Agenten kann daher durch Gleichung 6.17 beschrieben werden. Dabei gibt n_{vm} die Anzahl der aufgesuchten Händler an.

$$V_{mcap,ssa} = 2 * n_{vm} * \beta_{mcap,ssa} \left\{ 1 + \rho \frac{1 + c_{mcap,ssa}^2}{2(1 - \rho_{mcap})} \right\} \quad (6.17)$$

Beim AHB-Dienst ist die Verweilzeit der Aufträge beim MCAP von geringer Bedeutung, da die Kommunikationsverbindung zwischen dem Benutzer und dem MCAP direkt nach Erhalt des MAs wieder abgebaut wird.

6.3.4 Internet

Der Datenverkehr im Inter- und Intranet beruht auf dem IP-Protokoll. Bei Verbindungen, in denen das Internet und nicht die beteiligten Rechner den Flaschenhals darstellen, hängt die Nettodatenrate einer TCP/IP-Verbindung $b_{internet}$ in erster Linie von der Paketverlustrate P_{plr} (*packet loss rate*) und der Umlaufzeit T_{rtt} (*round-trip time*) ab. Die durchschnittliche Datenrate $b_{internet}$ lässt sich mit Gleichung 6.18 berechnen [MSMO97, PFTK98, Lud00]:

$$b_{internet} = \frac{B_{max}}{T_{rtt}} \frac{K_{tcp}}{\sqrt{P_{plr}}} \quad (6.18)$$

Dabei ist B_{max} die maximale Segmentgröße für ein TCP-Paket und K_{tcp} eine Konstante, die von der TCP-Implementierung abhängt.

Paketverluste treten hauptsächlich in den Internetroutern auf. Die Router können verschiedene Kommunikationsnetze - auch mit unterschiedlichen Datenraten - miteinander verbinden. Im Falle einer Überlast besitzen handelsübliche Router die Möglichkeit, einige Datenpakete in einem internen Speicher abzulegen. Ist jedoch der Speicher bereits vollständig belegt, muss der Router die Pakete verwerfen. Um eine solche Überlast frühzeitig zu erkennen, besitzt TCP den sogenannten *Congestion Avoidance* Algorithmus [APS99]. Dieser Algorithmus bestimmt das zustandsabhängige Verhalten von TCP durch variable

Fenstergrößen (*congestion window*).

Bei den untersuchten agentenbasierten Diensten bildet nicht das Internet, sondern die involvierten Rechner (MCAP, Bankserver und Händlerplattformen) bzw. die schmalbandigen, mobilen Zugangsnetze (CSD und GPRS) den Flaschenhals der Kommunikation. Außerdem werden die MAs innerhalb einer TCP-Verbindung übertragen, so dass nicht (wie beim Abruf einer gewöhnlichen Webseite) eine Vielzahl von TCP-Verbindungen benötigt wird. Es ist daher ausreichend, wenn die Verzögerungen durch Internetstrecken mit der Umlaufzeit T_{rtt} berücksichtigt werden. Typische Umlaufzeiten T_{rtt} und Paketverlustraten P_{plr} für Internetverbindungen können z.B. dem sogenannten Internet-Verkehrsbericht (Internet traffic report) [ITR01] entnommen werden, der alle 15 Minuten die wichtigsten Datenverbindungen weltweit überprüft. Die in der Analyse verwendete durchschnittliche Umlaufzeit beträgt 200 ms :

$$T_{rtt} = 200\text{ ms} \quad (6.19)$$

Dieser Wert wird im analytischen Gesamtmodell bei der Dienstausführungszeit und den Antwortzeiten des ASB-Dienstes (Kap. 6.6.5) als Verzögerung für das Internet mittels eines Bedienknotens berücksichtigt.

6.4 Mobile Agenten

In den beiden untersuchten M-Commerce-Diensten werden an verschiedenen Stellen mobile Agenten (MAs) zur Dienstangebotung und -ausführung eingesetzt. Während der MCAP beim ASB-Dienst ein WAP-Kommando von einem Mobilfunkteilnehmer erhält und die in diesem Kommando verpackten Aufgaben an MAs delegiert, werden die Agenten beim AHB-Dienst vom Mobilfunkteilnehmer selbst gestartet, und der MCAP setzt die Anforderungen der Agenten in HBCI-Nachrichten um. Unabhängig davon, dass die Dienste von ihrer Struktur her also grundverschieden sind, kann die für einen Agenten α bei der Migration von Ort ω_1 zum Ort ω_2 durchschnittlich zu übertragende Datenmenge generell durch Gleichung 6.20 beschrieben werden [SS97, Eve99, Hüc00]. Dabei wird mit α ein bestimmter MA benannt. Die weiteren Parameter bei der Agentenmigration sind in Tab. 6.7 erläutert.

$$B_{mig}(\alpha, \omega_1, \omega_2) = \begin{cases} 0 & \forall \omega_1 = \omega_2 \\ P_{code}(\alpha) * [B_{cr}(\alpha) + B_{code}(\alpha)] + B_{data}(\alpha) + B_{state}(\alpha) & \forall \omega_1 \neq \omega_2 \end{cases} \quad (6.20)$$

Die Größe des Agenten $B(\alpha)$ steht im direkten Zusammenhang mit der grundsätzlichen Leistungsfähigkeit des untersuchten Modells, denn das transferierte Datenvolumen hat unmittelbaren Einfluss auf die Kommunikationszeiten- und -kosten. Ferner muss beachtet werden, dass die Agentengröße keinesfalls immer konstant ist, sondern vom Auftrag und den Ergebnissen abhängt. D.h. die Größe eines MAs kann während der Migration größer oder kleiner werden. Er wird z.B. dann kleiner, wenn beim ASB-Dienst komplexere Produktangebote durch weniger komplexe, aber günstigere Angebote im Speicher des MAs substituiert werden können. Das Filtern und Austauschen von Ergebnissen wird *Selektivität* genannt und im Kap. 6.4.3 näher erläutert.

Symbol	Beschreibung
$B(\alpha)$	die Größe des Agenten in [byte]
$B_{code}(\alpha)$	Größe der Klassendatei in [byte]
$B_{cr}(\alpha)$	Größe der Anfrage nach der Klassendatei in [byte]
$B_{data}(\alpha)$	Größe des Datenspeichers des Agenten in [byte]
$B_{state}(\alpha)$	Größe der serialisierten Zustandsinformationen (Parameter, Laufzeitzustand, etc.) des Agenten in [byte]
$P_{code}(\alpha)$	Wahrscheinlichkeit, dass der Code (Klassendatei) des Agenten α nicht auf dem besuchten Rechner vorhanden ist, d.h. dass der Code übertragen werden muss

Tabelle 6.7: Parameter bei der Agentenmigration

6.4.1 Modell der Mobilität

Die **Mobilitätsfunktion eines Agenten** ist ein Vektor $m(\alpha)$, der aus allen Orten ω_i besteht, die der MA während der Migration besucht hat bzw. besuchen wird.

Wenn α ein mobiler Agent und ω_t ein Ort ist, bei dem der Agent zum Zeitpunkt t_i eintrifft, wobei i die Migrationsstationen chronologisch durchnummeriert, dann kann die Migrationsstrategie des Agenten α durch den Vektor $m(\alpha)$ wie folgt beschrieben werden:

$$m(\alpha) = \{\omega_{t_0}, \omega_{t_1}, \dots, \omega_{t_n}\} \quad (6.21)$$

Dabei ist ω_{t_0} der Ort, der den Agenten α zum Zeitpunkt t_0 startet, und ω_{t_n} der Ort, der den Agenten α zum Zeitpunkt t_n löscht. Folglich wird ω_{t_0} der *Startpunkt* und ω_{t_n} der *Zielort* des Agenten α genannt. Startpunkt und Zielort sind bei einer ringförmigen Migration, meist die Heimat (vgl. Kap. 3.2.3). Im entworfenen agentenbasierten Kommunikationskonzept ist der MCAP die Heimat der Such- und Bankagenten.

Mit $V_{place,i}$ wird die Verweilzeit beschrieben, die der MA an dem Ort i verbringt. Beim Startsystem verbleibt der Agent α demnach für die Dauer $V_{place,0}(\alpha)$. Allgemein lässt sich die Verweilzeit $V_{place,i}$ eines Agenten mit der Gleichung 6.22 berechnen, wobei durch $b_1, \dots, b_k$ unterschiedliche Teilstrecken bei der physikalischen Übertragung des MAs beschrieben werden können:

$$V_{place,n-1}(\alpha) = t_n - \sum_{i=1}^k \frac{B_{mig}}{b_i} - t_{n-1} \quad (6.22)$$

Mehrere Orte, die miteinander über Kommunikationsnetze verbunden sind, können eine Region (vgl. Kap. 3.2.3) bilden. Eine Region ist ein integraler Bestandteil der Agentenumgebung. Zur Betrachtung der Kommunikationszeit und -kosten ist es sinnvoll, zwischen lokalen und entfernten Orten, d.h. zwischen Orten, die auf dem gleichen bzw. einem entfernten Rechner installiert sind, zu unterscheiden.

Die **Agentenregion** $\Omega(t)$ setzt sich aus allen Orten $\omega_1(t), \dots, \omega_n(t)$ zusammen, die zum Zeitpunkt t über eine Kommunikationsinfrastruktur miteinander verbunden sind.

Gleichung 6.23 beschreibt diesen Sachverhalt:

$$\Omega(t) = \{\omega_1(t), \dots, \omega_n(t)\} \quad (6.23)$$

Hierbei muss berücksichtigt werden, dass Typ II-Agentensysteme (Kap. 5.1.1) und folglich auch deren Orte zwischenzeitlich von der Region abgetrennt sein können. Daraus folgt, dass die *Kernregion* Ω_{core} lediglich aus Orten von Typ I-Agentensystemen und $\Omega(t)$ eine temporäre Momentaufnahme der Region darstellt, wobei $\Omega(t) - \Omega_{core}$ die Teilmenge nur zwischenzeitlich mit der Region verbundener Orte beschreibt.

Für eine etwas weniger granulare Betrachtung ist es ausreichend, die Agentenmigration als einen Vektor $M(\alpha)$ der besuchten Regionen zu definieren:

$$M(\alpha) = \{\Omega_{t_0}, \Omega_{t_1}, \dots, \Omega_{t_n}\} \quad (6.24)$$

Gleichung 6.22 kann nun analog dafür eingesetzt werden, um die gesamte Verweilzeit $V_{region,i}$ des Agenten α in der Region Ω_{t_i} zu berechnen. Nachfolgend wird die Verweilzeit V_m der MAs auf den Händlerplattformen mit dem in Abb. 6.10 dargestellten Warteschlangenmodell abgebildet, wobei von einer Vielzahl von Agentensystemen und Orten ausgegangen wird und alle ähnlichen Orte, wie z.B. die Händler, durch dieselbe M/G/1 Warteschlange modelliert werden. Ein Markoff'scher Ankunftsprozess liegt vor, da von vielen überlagerten, ankommenden Aufträgen ausgegangen werden kann. Die Verteilung der Bediendauer β_m wurde in Kap. 6.3.3 diskutiert.

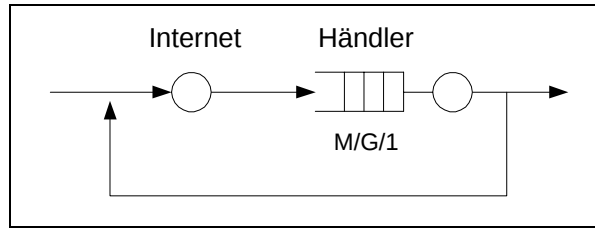


Abbildung 6.10: Verkehrsmodell für das Aufsuchen von Händlern

6.4.2 Modell der Interaktivität

Die **Ankunftsrate** λ_0 gibt die Anzahl der Benutzeranfragen pro Sekunde an, d.h. die Ankunftsrate ist ein Maß für die Dienstnutzung bzw. im Fall der AHB- und ASB-Dienste für die Interaktivität des Benutzers mit dem agentenbasierten Kommunikationskonzept.

Die Ankunftsrate hängt neben der Teilnehmerzahl n_{user} von der durchschnittlichen Anzahl von Anfragen pro Teilnehmer und Zeiteinheit r_{user} ab.

$$\lambda_0 = n_{user} * r_{user} \quad (6.25)$$

In der Hauptverkehrszeit, der sogenannten *busy hour*, gilt daher, wenn r_{bh} die korrespondierenden Rate für Aufträge pro Teilnehmer und Zeiteinheit ist, die folgende Relation:

$$\lambda_{bh} = n_{user} * r_{bh} \quad (6.26)$$

In Telekommunikationsnetzen wird häufig mit dem Verkehrswert in der Hauptverkehrsstunde A_{bh} gerechnet.

Der **Verkehrswert** A_{bh} ist ein anschauliches Mass für die Belastung eines Netzknotens in einem Kommunikationsnetz.

Für diesen Verkehrswert A_{bh} ist die Einheit *Busy Hour Call Attempts* (BHCA) gebräuchlich. Mit Gleichung 6.27 kann der Verkehrswert A_{bh} berechnet werden.

$$A_{bh} = \lambda_{bh} * 1 h \quad (6.27)$$

Kennt man die maximal verkraftbare Ankunftsrate λ_{max} eines Systems, so lässt sich mit Hilfe von Gleichung 6.26 die maximale Anzahl von Teilnehmern n_{max} wie folgt berechnen:

$$n_{max} = \frac{\lambda_{max}}{r_{bh}} \quad (6.28)$$

Die Dimensionierung der Kommunikationsplattform MCAP sollte in Abhängigkeit vom erwarteten Verkehrswert A_{bh} und den Bediendauern für die unterschiedlichen Suchmethoden und Banktransaktionen vorgenommen werden, so dass auch in der Hauptverkehrszeit alle Aufträge bearbeitet werden können und keine Überlastsituation, bei der die Verweildauer einzelner Aufträge erheblich steigt, eintritt.

Beim ASB-Dienst hängt die Anzahl der bei einer Produktsuche angefragten Händler n_{vm} , von der Gesamthändlerzahl n_m , von der Suchmethode (Kap. 5.4.2) und von der dedizierten Trefferwahrscheinlichkeit P_e ab.

Die **Trefferwahrscheinlichkeit** P_e gibt die Wahrscheinlichkeit an, mit der ein MA bei einem Händler das gesuchte Produkt unter den vom Benutzer spezifizierten Rahmenbedingungen findet.

Für die einzelnen Suchmethoden gilt daher:

- *Suchmethode ERSTES*: Bei dieser Suchmethode, bei der die Suche nach dem ersten gefundenen Treffer, der alle verlangten Eigenschaften erfüllt, direkt abgebrochen wird, werden die wenigsten Suchanfragen benötigt:

$$n_{vm} = \frac{1}{P_e} \quad (6.29)$$

- *Suchmethode DREI*: Aus Gleichung 6.29 lässt sich auch die durchschnittliche Anzahl von Händlern für diese Suchmethode ableiten, bei der insgesamt drei Treffer geliefert werden sollen:

$$n_{vm} = \frac{3}{P_e} \quad (6.30)$$

- *Suchmethoden BESTES und SMART*: Um das bestmögliche bzw. die geeignetsten Ergebnisse herauszufinden, muss der MA alle verfügbaren Händler aufsuchen.

$$n_{vm} = n_m \quad (6.31)$$

Beim ASB-Dienst finden drei Interaktionen mit dem Dienstinutzer (Login, Suchspezifikation und Bestellung) statt, siehe Abb. 6.11. Zur Berechnung der mittleren Dienstausführungszeit S_{asb} für die Produktsuche wurden für die Benutzerinteraktionen die folgenden mittleren Antwortzeiten angenommen:

$$T_{user,login} = 10 \text{ s} \quad (6.32)$$

$$T_{user,query} = 25 \text{ s} \quad (6.33)$$

Im Falle einer WAP-Verbindung über CSD werden dem Teilnehmer auch für die Antwortzeiten Gebühren in Rechnung gestellt. Die Verzögerungen durch Benutzereingaben beim AHB-Dienst hingegen werden nicht weiter betrachtet, da sie allesamt *offline*, d.h. ohne Netzverbindung, durchgeführt werden.

6.4.3 Modell der Selektivität

Die **Selektivität** γ eines Agenten ist seine Fähigkeit, die erhaltenen Informationen während der Migration zu verarbeiten und zu filtern.

Im Gegensatz zu RPCs haben MAs die Möglichkeit, redundante Informationen bereits bei der Informationsquelle, d.h. an dem Ort, wo die Information abgelegt ist bzw. abgefragt wird, zu löschen und lediglich die für den Teilnehmer interessanten Daten weiter zu transportieren. Gleichung 6.34 beschreibt die Selektivität als Verhältnis der gefilterten Daten B_{filter} zur angebotenen Datenmenge B_{sum} . Daraus folgt, dass die Selektivität γ ein dimensionsloser Faktor zwischen null und eins ist.

$$\gamma = \frac{B_{filter}}{B_{sum}} \quad 0 \leq B_{filter} \leq B_{sum} \quad (6.34)$$

Die Selektivität eines MAs ist einerseits abhängig von der Intelligenz des Agenten, andererseits hängt sie von seiner Autonomie ab, d.h. welche Rechte der Benutzer dem Agenten einräumt. Durch die Fähigkeit des MAs, Daten lokal zu filtern, kann die Größe des Datenspeichers B_{data} des Agenten vor der Migration teilweise nicht unerheblich reduziert werden. Gleichung 6.35 zeigt das mathematische Modell der Selektivität, wobei B_{sum} die Datenmenge kennzeichnet, die der MA zur Verarbeitung beim Ort ω_1 vor der Migration zum Ort ω_2 erhalten hat.

$$B_{data}(B_{sum}, \gamma, \omega_1, \omega_2) = \begin{cases} 0 & \forall \omega_1 = \omega_2 \\ (1 - \gamma)B_{sum} & \forall \omega_1 \neq \omega_2 \\ & \wedge 0 \leq \gamma \leq 1 \end{cases} \quad (6.35)$$

Gewinnt man die Datenmengen B_{data} und B_{sum} aus Messungen, so kann die Gleichung 6.35 nach der Selektivität aufgelöst werden und die Selektivität γ kann für die verschiedenen Suchszenarien errechnet werden.

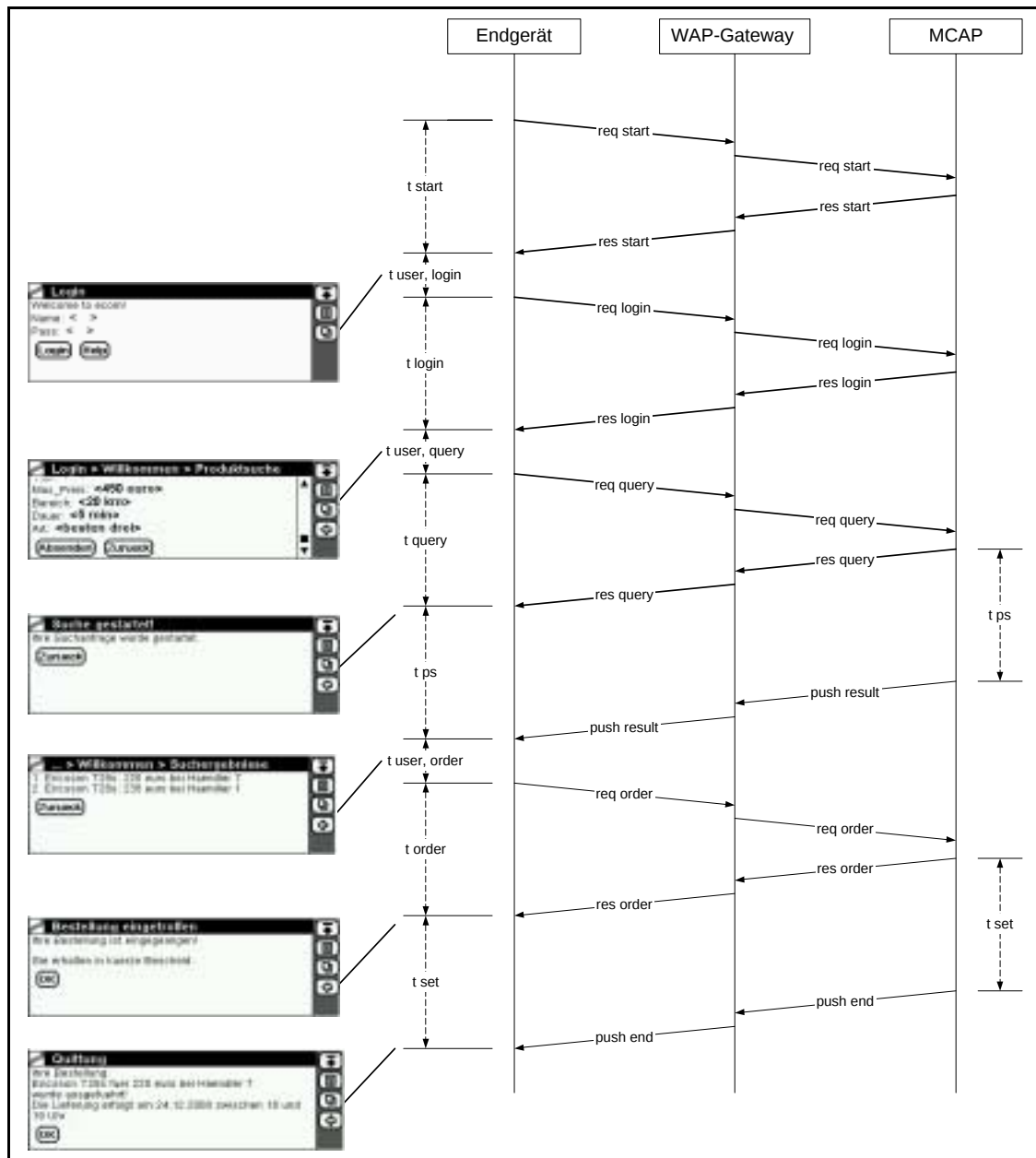


Abbildung 6.11: Antwortzeiten beim ASB-Dienst

$$\gamma = 1 - \frac{B_{data}}{B_{sum}} \quad \forall B_{sum} \geq B_{data} \quad (6.36)$$

6.5 Analyse des Homebankingdienstes

Die Analyse basiert auf dem im Rahmen dieser Arbeit entstandenen flexiblen AHB-Dienst, der in Kap. 5.3 ausführlich diskutiert worden ist. Für eine traditionelle Homebanking-Anwendung wird die Dienstausführungszeit durch die übertragene Datenmenge und die Bearbeitungszeit der Aufträge bestimmt. Beim AHB-Dienst spielt die Bearbeitungszeit der Bank keine Rolle mehr. Deshalb wurden zur Bewertung des Dienstes lediglich die Onlinezeit eines Benutzers und die übertragenen Datenvolumina durch Messungen und Verkehrsanalysen untersucht.

Abb. 6.12 zeigt den Ablauf einer herkömmlichen HBCI-Anwendung, die sich einer klassischen Ende-zu-Ende Client/Server-Architektur bedient. Der Bankserver, der die Kundenaufträge annimmt und bearbeitet, kann vom Teilnehmer (Kunden) via ISP und Internet mittels eines speziellen Anwendungsprogramms erreicht werden. Dabei befinden sich die Hauptakteure, d.h. das Endgerät des Teilnehmers und der Bankserver, am Netzrand und kommunizieren miteinander über das standardisierte HBCI-Protokoll (Kap. 5.3).

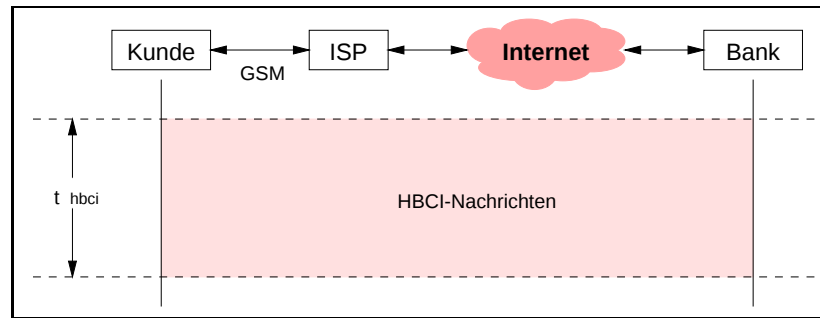


Abbildung 6.12: Klassische HBCI-Anwendung

Der Ablauf des AHB-Dienstes ist in Abb. 6.13 dargestellt. Aus der Sicht des Bankservers ist der MCAP ein gewöhnlicher HBCI-Client. Im untersuchten Fall wird zwischen dem MCAP und dem ISP eine Datenverbindung über ISDN aufgebaut. Für den kommerziellen Einsatz könnte der MCAP direkt bei einem ISP bzw. bei einem Netz- oder Diensteanbieter angesiedelt sein. Aus Sicherheitsgründen ist jedoch ein Standort in den Räumen des Benutzers zu bevorzugen. Möchte man den AHB-Dienst nutzen, so muss zunächst ein Allgemeiner Dienstagent (ADA), der *Homebanking Terminal Agent* (HTA), auf dem Endgerät des Benutzers gestartet werden. Im Transaktionsfall werden zwischen dem HTA und dem MCAP Spezielle Dienstagenten (SDA), die Bankagenten, übertragen.

Für den AHB-Dienst wurden drei Geschäftsfälle implementiert: die Saldenabfrage, die Umsatzabfrage und die Inlandsüberweisung. Die Daten für diese Geschäftsfälle wurden gemessen und verglichen. Die Reduzierung der Dienstausführungszeit der HBCI-basierten Bankanwendung war dabei die Anforderung, die den Entwurf und die Implementierung des AHB-Dienstes weitestgehend beeinflusst haben.

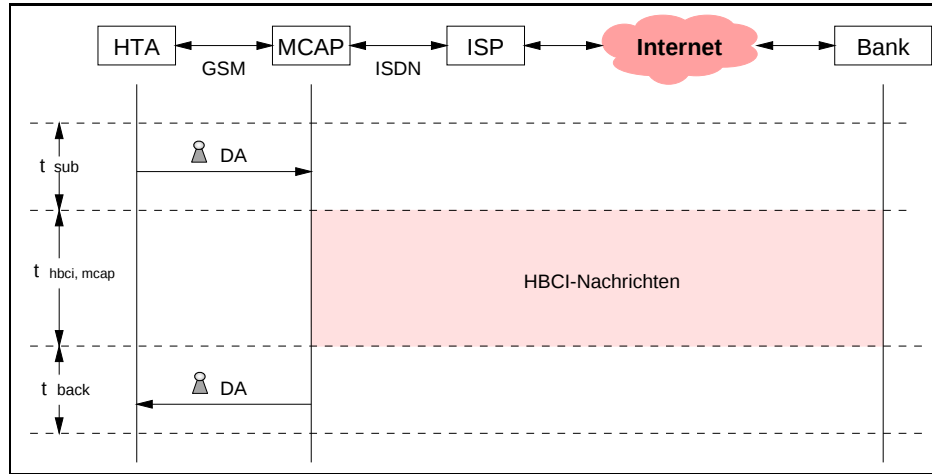


Abbildung 6.13: Definition der Antwortzeiten beim AHB-Dienst

6.5.1 Datenvolumen

Aus der Sicht der Anwendungsschicht haben die Nachrichtenanzahl und die zu übertragenden Nutzdaten der Anwendung den größten Einfluss auf die Dienstausführungszeit. Die Nutzdaten (*payload*) der Anwendung sind dabei die Daten, die direkt an die Transportschicht (TCP) gegeben bzw. von dieser Schicht nach oben gereicht werden, siehe auch Abb. 6.8 a). Das Datenvolumen der klassischen HBCI-Anwendung setzt sich daher aus der Menge der gesendeten ($B_{hbci, req}$) und empfangenen ($B_{hbci, res}$) HBCI-Nachrichten zusammen, siehe Gleichung 6.37.

$$B_{hbci} = B_{hbci, req} + B_{hbci, res} \quad (6.37)$$

Bei der klassischen HBCI-Anwendung werden für jeden Auftrag die gleichen Nachrichtentypen und dieselbe Anzahl von Nachrichten übertragen. Folglich besteht zwischen dem Datenvolumen und der Auftragsanzahl eine direkte Proportionalität.

Beim AHB-Dienst wird eine CSD-Mobilfunkverbindung lediglich für die Migration der Bankagenten aufgebaut. Für jeden dieser Dienstagenten (DAs) werden zwei Übertragungen (hin und zurück) benötigt. Das zu übertragende Datenvolumen für die Migration des DAs kann, wie in Gleichung 6.38, als Summe der Datenmengen aus beiden Übertragungen, d.h. vom Endgerät zum MCAP (B_{sub}) und zurück (B_{back}) beschrieben werden:

$$B_{mig} = B_{sub} + B_{back} = 2 * B_{ahb} \quad (6.38)$$

Da bei beiden Übertragungen annähernd das gleiche Datenvolumen übertragen wird, kann B_{mig} beim AHB-Dienst auch als doppeltes Datenvolumen einer durchschnittlichen Agentenmigration B_{ahb} berechnet werden. Nach der erstmaligen Einrichtung des AHB-Dienstes ist der Programmcode des Agenten $B_{code, ahh}$ auf dem mobilen Endgerät vorhanden, so dass bei folgenden Transaktionen das Datenvolumen B_{ahb} lediglich aus den Zustandsinformationen $B_{state, ahh}$ und dem Datenspeicher $B_{data, ahh}$ des Agenten besteht:

$$B_{ahb} = B_{state, ahh} + B_{data, ahh} \quad (6.39)$$

Die durchgeführten Messungen haben ergeben, dass für einen MA ohne Auftrag im Mittel 4.413 *byte* Zustandsinformationen $B_{state,ahb}$ übertragen werden. Dabei handelt es sich hauptsächlich um Daten für die Voyager-interne Kommunikation und Administration, die nicht verringert werden können, obwohl wesentlich mehr Informationen transportiert werden, als für den AHB-Dienst wirklich notwendig sind. Der Grund dafür liegt darin, dass sich mehrere kleine Softwareobjekte zu einem großen Overhead summieren. Eine Verbesserungsmöglichkeit wäre, alle anwendungsspezifischen Daten in Datenfeldern abzulegen. Dies verringert die Datenmenge, führt aber auch dazu, dass sich der Bearbeitungsaufwand und damit auch die Bediendauer im MCAP β_{mcap} erhöht. Dennoch ist diese Lösung für eine kommerzielle Anwendung durchaus denkbar.

In Abb. 6.14 sind nun die Datenvolumina der klassischen Bankanwendung B_{hbc} und des AHB-Dienstes B_{mig} gegeneinander für Überweisungsaufträge aufgetragen. Es wird deutlich, dass durch den Einsatz von MAs insbesondere dann eine nachhaltige Wirkung erzielt wird, wenn dem DA mehrere Aufträge übergeben werden. Dabei besteht der Datenspeicher des DAs $B_{data,ahb}$ lediglich aus der Auftragsinformation und erhöht sich pro Überweisungsauftrag durchschnittlich um 266 *byte*.

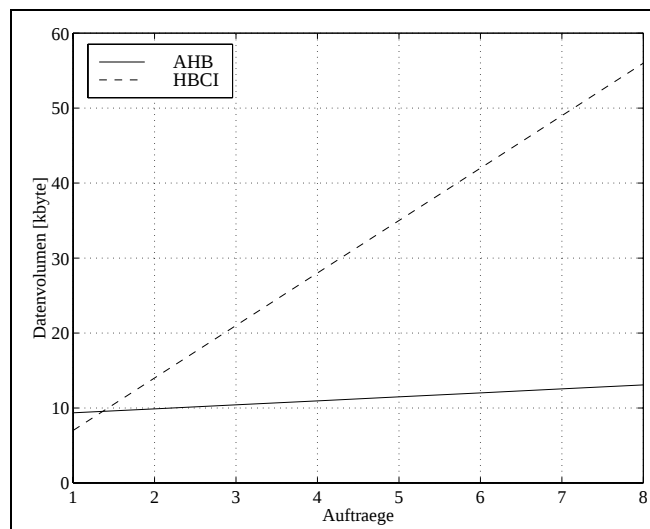


Abbildung 6.14: Vergleich der Datenvolumina B_{ahb} der Bankanwendungen

Für die Untersuchung des Einflusses von SSL wurden Messungen bei der Agentenmigration mit und ohne SSL durchgeführt. Die dabei ermittelten Ergebnisse zeigen, dass für den SSL-Sitzungsaufbau weitere 1,5 *kbyte* Daten übertragen werden. In Tab. 6.8 wird das Datenvolumen $B_{code,ahb}$ des HTAs (Programmcode), mit bzw. ohne SSL auf der Transportschicht (TCP) festgehalten. Gelingt es dem Benutzer den ADA über eine breitbandige Datenverbindung zum Endgerät zu übertragen, z.B. über das Festnetz oder zentrale Datenübertragungspunkte mit hoher Übertragungskapazität (z.B. *Mediapoints*), so kann die Übertragungszeit für den HTA, dessen Mobilität eine flexible und personalisierte Dienstnutzung mit verschiedenen Endgeräten erlaubt, gering gehalten werden.

Datenvolumen $B_{code,ahb}$	ohne SSL	mit SSL
Homebanking Terminal Agent (HTA)	75,1 <i>kbyte</i>	80,6 <i>kbyte</i>

Tabelle 6.8: Datenvolumen des HTAs

6.5.2 Antwortzeiten

Zunächst wird nur die Antwortzeit des Dienstes betrachtet, d.h. die Verbindungsaufbau- und -abbauzeiten t_{cs} werden vernachlässigt, da diese von der dedizierten Hardware- und Softwarekonfiguration abhängen und sowohl bei der traditionellen, als auch bei der agentenbasierten Lösung gleichermaßen auftreten. In der herkömmlichen Lösung ist die Antwortzeit t_{hbc} das Zeitintervall zwischen dem Zeitpunkt, an dem der Benutzer einen oder mehrere Aufträge abgibt und dem Zeitpunkt, an dem die entsprechenden Antworten zurückkommen (siehe Abb. 6.12). Sie besteht folglich aus der Übertragungszeit t_{trans} und Bearbeitungszeit t_{proc} der HBCI-Nachrichten:

$$t_{hbc} = t_{trans} + t_{proc} \quad (6.40)$$

Für die Bearbeitung der HBCI-Nachrichten durch den Transaktionsserver der Bank wird beim AHB-Dienst keine Verbindung mit dem Benutzer benötigt. Beim AHB-Dienst wird in der Regel daher die Verbindung zwischen dem HTA und dem MCAP nach der Agentenmigration direkt abgebaut. Deshalb besteht die Antwortzeit t_{mig} des AHB-Dienstes lediglich aus den Zeiten, die für die Migration der DAs (siehe Abb. 6.13) anfallen:

$$t_{mig} = t_{sub} + t_{back} \quad (6.41)$$

Tab. 6.9 zeigt die gemessenen mittleren Antwortzeiten für die drei unterschiedlichen Dienstrealisierungen - klassisches HBCI (HBCI), HBCI mit der Kommunikationsplattform MCAP (AHB), sowie HBCI mit MCAP und SSL-Verschlüsselung (AHB+) für die Geschäftsfälle Saldenabfrage, Umsatzabfrage und Inlandsüberweisung in Abhängigkeit von der Auftragsanzahl aufgelistet. Dabei wurde die Verzögerungen durch den Auf- und Abbau der CSD-Datenverbindung t_{cs} nicht berücksichtigt. Es ist deutlich zu sehen, dass das agentenbasierte Kommunikationskonzept die Onlinezeit des Benutzers gegenüber der herkömmlichen Dienstanbietung (T_{hbc}) erheblich verkürzt. Mittels der gemessenen Antwortzeiten konnte auch die angenommene Transaktionsleistung für die Agentenmigration verifiziert werden.

Für eine Betrachtung der mittleren Dienstausführungszeit S_{ahb} muss beim agentenbasierten AHB-Dienst jeweils ein Verbindungsauf- bzw. -abbau mehr als beim traditionellen HBCI-Szenario berücksichtigt werden, da für jede Transaktion zwei Agentenmigrationen notwendig sind. Ferner werden für den Verbindungsaufbau zwischen zwei Voyager-Umgebungen mit SSL-Unterstützung zusätzlich noch etwa weitere neun Sekunden für den Aufbau der SSL-Sitzung benötigt (Kap. 6.3.1).

Bei der Untersuchung der traditionellen HBCI-Anwendung wurde festgestellt, dass die Bearbeitungszeit t_{proc} der Aufträge beim Bankserver wesentlich länger als die Übertragungszeit t_{trans} der HBCI-Nachrichten ist. Beim AHB-Dienst schlägt sich die lange Bearbeitungszeit t_{proc} aufgrund des zwischenzeitlichen Verbindungsabbaus nicht nieder. Hierin ist auch begründet, warum die mittlere Antwortzeit T_{mig} erheblich kürzer als T_{hbc} ist, obwohl in beiden Szenarien eine annähernd gleiche Datenmenge über eine CSD-Datenverbindung

Anzahl der Aufträge	Saldenabfrage		
	T_{hbc}	T_{mig} (AHB)	T_{mig} (AHB+)
0	–	9,3 s	10,1 s
1	80,0 s	10,2 s	11,0 s
2	160,8 s	11,0 s	11,9 s

Anzahl der Aufträge	Umsatzabfrage		
	T_{hbc}	T_{mig} (AHB)	T_{mig} (AHB+)
0	–	9,3 s	10,1 s
1	76,8 s	10,3 s	11,1 s
2	152,4 s	11,1 s	12,0 s

Anzahl der Aufträge	Überweisung		
	T_{hbc}	T_{mig} (AHB)	T_{mig} (AHB+)
0	–	9,3 s	10,1 s
1	82,0 s	10,2 s	11,3 s
2	162,8 s	11,0 s	12,3 s

Tabelle 6.9: Vergleich der Antwortzeiten der Bankanwendungen

übertragen wird (siehe auch Kap. 6.5.1). Eine weitere Erkenntnis ist, dass die Anzahl der Aufträge in der agentenbasierten Lösung keinen großen Einfluss auf die Antwortzeit der Anwendung hat, während bei der traditionellen Anwendung zwischen der Anzahl von Aufträgen und der Antwortzeit eine nahezu lineare Beziehung besteht. Der theoretische Fall 'kein Auftrag' beim AHB-Dienst wird betrachtet, um die reine Migrationszeit des MAS mit einem leeren Datenspeicher zu ermitteln. Eine vergleichbare Situation konnte bei der klassischen HBCI-Anwendung nicht hergestellt werden.

Beim AHB-Dienst wird beim erstmaligen Start zunächst der erforderliche ADA, d.h. der Programmcode B_{code} des HTAs übertragen, der nach der Übertragung auf dem Endgerät gespeichert werden kann. Beim nächsten Aufruf kann dann der lokale HTA verwendet werden. Tab. 6.10 zeigt die korrespondierenden mittleren Übertragungszeiten des Agenten $T_{code,ahb}$ über eine CSD-Verbindung mit und ohne SSL-Unterstützung.

Übertragungszeit $T_{code,ahb}$	ohne SSL	mit SSL
Homebanking Terminal Agent (HTA)	85 s	118 s

Tabelle 6.10: Übertragungszeit des HTAs über CSD

6.5.3 Dienstausführungszeit

Für die Berechnung der mittleren Dienstausführungszeit S_{ahb} wird nun auch die mittlere Verzögerungen für den Auf- und Abbau (Kap. 6.3.1) einer TCP/IP-Datenverbindung T_{cs} hinzugezogen.

$$S_{ahb} = \sum T_i = T_{cs} + T_{sub} + T_{cs} + T_{back} = T_{mig} + 2 * T_{cs} \quad (6.42)$$

Die gemessene mittlere Dienstaussführungszeit S_{ahb} liegt zwischen 62,2 s (ein Auftrag) und 68,4 s (acht Aufträge). Es wird deutlich, dass die Verbindungsaufbauzeit T_{cs} die Dienstaussführungszeit beim AHB-Dienst dominierend beeinflusst. So verlängert sich die Dienstaussführungszeit pro weiterer Überweisung im Mittel um 0,8 Sekunden, während bei der klassischen HBCI-Lösung mit einer Verdoppelung der Antwortzeit und folglich mit einer starken Zunahme der gesamten Dienstaussführungszeit gerechnet werden muss. Während die Dienstaussführungszeit und damit die Onlinezeit des Benutzers beim AHB-Dienst also nur unwesentlich steigt, wenn mehrere Bankaufträge gleichzeitig erteilt werden, so wird die Schere zwischen den beiden Bankanwendungen mit steigender Auftragsanzahl immer größer. Daraus folgt, dass der Einsatz von MAs um so effizienter ist, je mehr Aufträge dem DA übergeben werden.

Bei der Nutzung des AHB-Dienstes steht es dem Benutzer frei zu wählen, ob der MA in einer gesonderten Sitzung zurückgeholt wird oder die Rückkehr des MAs bei der nächsten Auftragserteilung ausreichend ist. Hierdurch kann die mittlere Dienstaussführungszeit für den Benutzer optimiert werden.

6.5.4 Kommunikationskosten

Für die Telekommunikationsteilnehmer erwächst aus der Tatsache, dass der AHB-Dienst die notwendige Online-Zeit des Benutzers für Banktransaktionen verringert, ein potentieller finanzieller Vorteil. Netzbetreiber legen für die Berechnung der Kommunikationskosten unterschiedliche Tarife und Gebühreneinheiten K_{pri} zu Grunde. Generell können die Kosten für den AHB-Dienst mit Gleichung 6.43 und die Anzahl der Gebühreneinheiten U_{sum} mit Gleichung 6.44 berechnet werden.

$$C_{ahb} = \left\lceil \frac{T_{sub}(\omega_1, \omega_2, B(\alpha)) + T_{cs}}{T_{takt}} \right\rceil K_{pri} + \left\lceil \frac{T_{back}(\omega_1, \omega_2, \gamma, B(\alpha)) + T_{cs}}{T_{takt}} \right\rceil K_{pri} \quad (6.43)$$

$$U_{sum} = \frac{C_{ahb}}{K_{pri}} = \left\lceil \frac{T_{sub}(\omega_1, \omega_2, B(\alpha)) + T_{cs}}{T_{takt}} \right\rceil + \left\lceil \frac{T_{back}(\omega_1, \omega_2, \gamma, B(\alpha)) + T_{cs}}{T_{takt}} \right\rceil \quad (6.44)$$

Selbst wenn man die Kommunikationskosten bei dem für den Benutzer ungünstigsten Gebührentakt, dem 60-Sekunden-Takt, betrachtet, kann man feststellen, dass mit Hilfe von MAs Kommunikationskosten eingespart werden können. Tab. 6.11 vergleicht die anfallenden Gebühreneinheiten U_{sum} für Banktransaktionen beim klassischen HBCI-Dienst und beim AHB-Dienst. Dabei fällt immer eine Einheit für den Transport des MAs zum MCAP und eine Einheit für dessen Rücktransport zum Endgerät an. Untersucht man kürzere Takteinheiten, so fällt der Vorteil des AHB-Dienstes noch deutlicher ins Gewicht.

Anzahl Banktransaktionen	1	2	3	4	5	6	7	8
Telefoneinheiten Klassisches HBCI	2	4	5	6	8	9	10	12
Telefoneinheiten AHB-Dienst	2	2	2	2	2	2	2	2

Tabelle 6.11: Gebühreneinheiten bei der Bankanwendung

Wenn der Benutzer für seine Bankaufträge, wie z.B. eine Inlandsüberweisung, keine direkte Bestätigung durch die Bank benötigt, sondern eine Benachrichtigung beim nächsten *virtuellen* Besuch der Bank ausreichend ist, dann reduzieren sich die Kosten um weitere 50 Prozent, da man auf den direkten Rücktransport des MAS verzichten kann und den MA beim nächsten Bankauftrag innerhalb der bestehenden Verbindung zurückholen kann.

6.5.5 Dienstausführungszeit bei einer GPRS-Verbindung

Der GPRS-Mobilfunkdienst ermöglicht Teilnehmern für digitale Datenpakete stets erreichbar zu sein, ohne dass man für diese Erreichbarkeit mit einer kommunikationszeitabhängigen Gebühr belastet wird. Dabei wird der Wechsel von *Standby*-Zustand in den Zustand *Aktiv* von der Mobilstation (Kap. 2.3.1) in rund 10 *ms* realisiert [Eri99b]. Es lassen sich daher direkt zwei deutliche Vorteile für die Nutzung von zukünftigen M-Commerce-Diensten über GPRS formulieren:

- Die Zeit, die ein Benutzer warten muss, bevor die eigentliche Datenübertragung einsetzt, wird durch die Verwendung einer GPRS- anstatt einer CSD-Kommunikationsverbindung erheblich reduziert.
- Dem Teilnehmer werden keine langen Verbindungsauf- und abbauzeiten mehr in Rechnung gestellt (Tab. 6.4).

Bei der Analyse des GPRS-Mobilfunkstandards für die beiden untersuchten M-Commerce-Dienste mussten die Übertragungszeiten simuliert bzw. gemittelt werden, da GPRS keine konstanten Datenraten bietet. Die beiden Dienste werden daher mit Hilfe des in Abschnitt 6.3.1 vorgestellten Modells in Abhängigkeit von der Verkehrslast und der Dienstgüte bewertet. Diese Analyse erlaubt einen Vergleich zwischen der Leistungsfähigkeit von MAS über leitungsorientierte (CSD) und paketorientierte (GPRS) Mobilfunkkanäle.

Sollte die Netzlast im GPRS-System erheblich steigen, so wird automatisch die Dienstgüte für die Teilnehmer reduziert, um so viele Teilnehmer wie möglich bedienen zu können. Daraus resultiert, dass der Unterschied zwischen den verschiedenen GPRS-Dienstgüten, der bei niedriger Netzlast noch recht gering ist, bei höherer Netzlast deutlich ansteigt. In den vorangegangenen Kapiteln wurde gezeigt, dass der AHB-Dienst der klassischen Homebanking-anwendung im Hinblick auf die Dienstausführungszeit und das zu übertragende Datenvolumen überlegen ist. Auffällig ist jedoch, dass der größte Anteil der Onlinezeit des Teilnehmers für den Aufbau der CSD-Datenverbindung benötigt wird. So muss der Benutzer im Falle einer Überweisung erheblich länger auf den Verbindungsaufbau als auf die Übertragung des MAS mit dem entsprechenden Bankauftrag warten. Da ein GPRS-Teilnehmer stets erreichbar ist, entfällt bei der Nutzung des AHB-Dienstes über GPRS die Verzögerung für den Kanalaufbau, vgl. Abb. 6.5. Es muss allerdings berücksichtigt werden, dass das Netz für den Teilnehmer beim Einbuchen ins Netz ein PDP-Kontext (ca. 1 Sekunde) und für jede Richtung einer Blockdatenübertragung ein TBF¹⁴-Kontext (downlink: 100 bis 250 *ms*, uplink: 200 bis 500 *ms*) eingerichtet werden muss.

Paketorientierte Systeme teilen die Informationsdaten in Pakete ein, wobei jedes Paket zusätzlich Informationen wie Adress- und Steuerungsdaten enthält. Dadurch wird die zu übertragende Datenmenge geringfügig erhöht. Herkömmliche GPRS-Mobiltelefone und Windows-Betriebssysteme unterstützen eine maximale TCP/IP-Segmentgröße (MTU¹⁵) von 1.500 *byte*. Für den Transport eines MAS mit zwei Überweisungsaufträgen wurden

¹⁴Temporary Block Flow

¹⁵Maximum Transmission Unit

durchschnittlich 4.945 *byte* TCP-Nutzdaten gemessen. In einem GPRS-System können diese MAs folglich in vier TCP/IP-Segmenten übertragen werden.

Wie in Kap. 6.5.3 beschrieben wurde, liegt die mittlere Dienstausführungszeit für zwei Überweisungsaufträge im CSD-Fall bei etwa 63 Sekunden. Diese Verzögerung ist relativ unabhängig von der Netzlast, da die Übertragungsrate nach Etablierung der Datenverbindung über CSD recht konstant ist. Basierend auf den Datenmengen aus Kap. 6.5.1 kann nun auch die durchschnittliche Dienstausführungszeit für die unterschiedlichen GPRS-Verkehrsmodelle berechnet werden. Abb. 6.15 stellt die Dienstausführungszeit für zwei Banküberweisungen graphisch dar. Es zeigt sich, dass bei einer Netzlast ρ bis 70 % lediglich im schlechtesten Fall (*worst case*) mit einer höheren Dienstausführungszeit gerechnet werden muss. Gelingt es dem Netzbetreiber jedoch nicht, Überlastsituationen zu verhindern, so kann eine GPRS-Verbindung durchaus zu höheren Verzögerungen als eine leitungsorientierte Verbindung (CSD) führen.

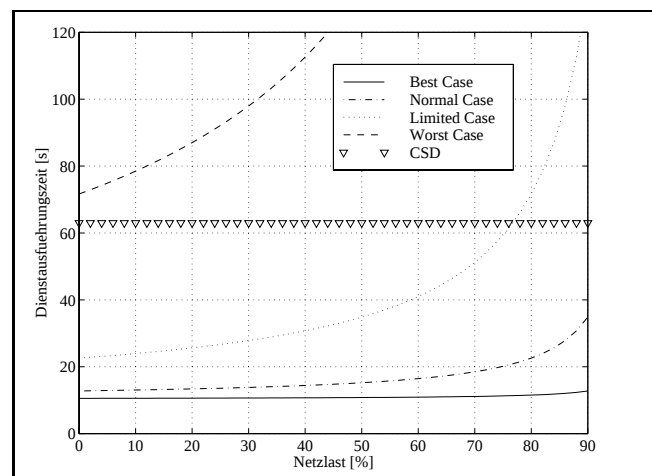


Abbildung 6.15: Dienstausführungszeit des AHB-Dienstes über GPRS

Zusammenfassend kann festgehalten werden, dass GPRS die Onlinezeit der Benutzer erheblich verringern kann. Doch auch, wenn mit höheren Übertragungsraten als bei CSD gerechnet werden kann, so sorgen die hohen Umlaufzeiten beim Nachrichtenaustausch dafür, dass die Transaktionsleistungen stark von der Anzahl der verschickten Nachrichten abhängt. Der Austausch einer Vielzahl von kleinen Nachrichten sollte daher vermieden werden. Daraus folgt auch, dass der Einsatz des klassischen HBCI-Protokolls nicht nur bei über CSD-, sondern auch bei einer GPRS-Mobilfunkkanäle nicht sinnvoll ist, da die nachrichtenintensive Konzeption von HBCI nicht mit den hohen Laufzeitverzögerungen von GPRS zusammenpasst. Das Agentenparadigma, nach dem Daten in einem MA zusammengefasst werden, bietet hier deutliche Vorteile. Der größte Vorteil entsteht für agentenbasierte Kommunikationskonzepte bei GPRS-Systemen dadurch, dass für die Übertragung von Datenpaketen auf den zeitintensiven Aufbau einer Datenverbindung (vgl. Tab. 6.4) verzichtet werden kann.

6.6 Leistungsbewertung des Such- und Bestelldienstes

Für die Leistungsanalyse des ASB-Dienstes wurde das Gesamtsystem in mehrere logische Teilsysteme zerlegt, siehe auch Kap. 6.3. Nachfolgend werden die Datenvolumina und

die Antwortzeiten des ASB-Dienstes mittels Messungen und eines verkehrstheoretischen Modells diskutiert.

6.6.1 Verkehrsmodell für den ASB-Dienst

Abb. 6.16 zeigt das den analytischen Berechnungen dieser Arbeit zugrundeliegende Gesamtmodell für den Kommunikationsfluss des ASB-Dienstes. Verschiedene Benutzer speisen ein GSM-Mobilfunknetz (1) mit Aufträgen. Die betrachteten Aufträge werden vom mobilen Endgerät in WBXML verfasst und vom Mobilfunknetz an ein WAP-Gateway (2) weitergereicht. Da der WAP-Gateway ein externer Netzknoten ist, der von verschiedenen Telekommunikationsnetzen Aufträge erhält, kann beim WAP-Gateway mit einem Markoff'schen Ankunftsprozess gerechnet werden. Der WAP-Gateway übersetzt die Aufträge von WBXML in WML und überträgt diese über das Internet (3) zu einem WAP-Server. Für das Internet wird eine konstante Verzögerung, die in Kap. 6.3.4 begründet wurde, angenommen. Der WAP-Server ist innerhalb der neu konzipierten Kommunikationsplattform MCAP (4) integriert. Der MCAP nimmt ASB-Aufträge von verschiedenen WAP-Gateways und andere M-Commerce-Aufträge (z.B. AHB-Aufträge) direkt von mobilen Endgeräten entgegen. Daher wird auch hier ein Markoff'scher Ankunftsprozess angenommen. Der MCAP setzt die ASB-Aufträge in mobile Agenten um und sendet diese über das Internet (5) zu Händlerrechnern (6). Verschiedene Händler werden im Modell durch dasselbe Verkehrsmodell abgebildet. Daher erfolgt im Modell eine Rückführung vom Händler (6) zurück zum Internet (5). Hat der mobile Agent den letzten Händler aufgesucht oder bricht er die Migration vorzeitig ab, so erfolgt der Rücktransport über das Internet (7) zum MCAP (8). Der MCAP setzt die zusammengetragenen Ergebnisse des mobilen Agenten in ein WML-Dokument um und speichert dieses im WAP-Server. Damit liegen die Ergebnisse nun für den Benutzer abrufbereit beim WAP-Server. Wie in Kap. 5.2.4 erläutert wurde, gibt es verschiedene Wege (erneuter Anruf, SMS, WML-Push, etc.) wie der Benutzer das Ergebnis erhält. In der untersuchten Realisierung wurde in der WAP-Seite ein Timer (9) gesetzt, der die Ergebnisse nach einer angemessenen Zeit vom WAP-Server abfragt. Die Ergebnisse werden dann über den WAP-Gateway (10) und das Mobilfunknetz (11) zum Benutzer gesendet.

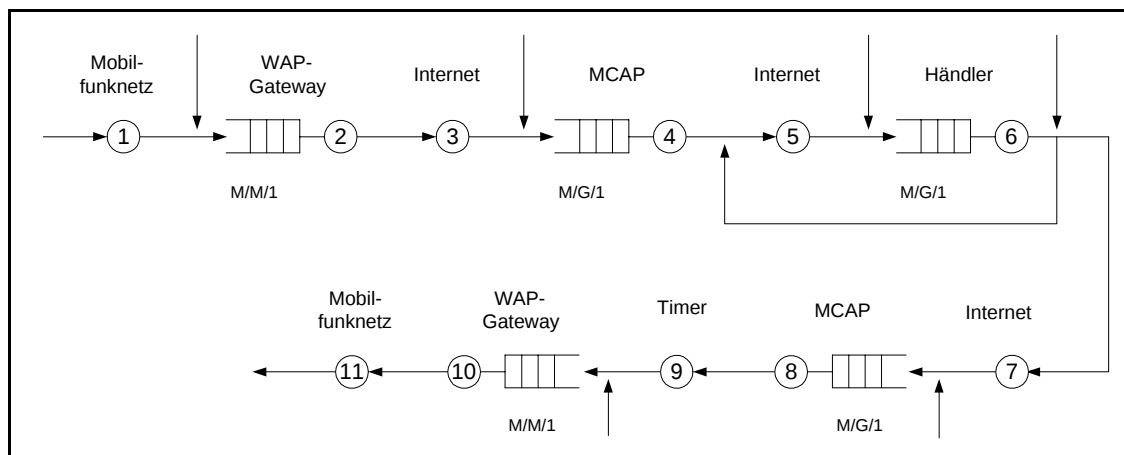


Abbildung 6.16: Verkehrsmodell für den mobilen Suchagenten

6.6.2 WAP-Kommunikation über CSD

In diesem Abschnitt wird die Kommunikation zwischen dem MCAP und dem mobilen Endgerät untersucht. Tab. 6.12 beschreibt in Anlehnung an Abb. 6.11 den Uplink, d.h. das Datenvolumen, welches vom Endgerät zum MCAP gesendet wird. Dabei wird zwischen der Größe der WTP-Nachrichten $B_{udp,up}$ (UDP-Nutzdaten), die vom Endgerät zum WAP-Gateway geschickt werden, und der Größe der HTTP-Nachrichten $B_{tcp,up}$ (TCP-Nutzdaten), die der WAP-Gateway zum MCAP sendet, unterschieden.

Vorgang	$B_{udp,up}$	$B_{tcp,up}$
<i>start</i>	70 <i>byte</i>	831 <i>byte</i>
<i>login</i>	91 <i>byte</i>	847 <i>byte</i>
<i>query</i>	154 <i>byte</i>	966 <i>byte</i>

Tabelle 6.12: WAP-Nutzdaten der Transportschicht (Uplink)

Der Downlink wird analog zu Tab. 6.12 durch Tab. 6.13 beschrieben. Es fällt auf, dass, ähnlich wie beim gewöhnlichen Webverkehr, das anfallende Datenvolumen auf dem Downlink erheblich größer ist als auf dem Uplink. Dies liegt daran, dass Benutzer für ihre Anfragen lediglich wenige Daten zum MCAP schicken, während der MCAP hingegen die Ergebnisse in umfangreicheren WML/PHP-Dokumenten zum mobilen Endgerät zurücksendet. Aufgrund der Tatsache, dass es sich hierbei um einen personalisierten Dienst handelt, ist es in der Regel auch nicht erforderlich, dass der Teilnehmer personenbezogene Daten bei der Dienstinutzung übermittelt, da diese dem MCAP schon bekannt sind.

Vorgang	$B_{udp,down}$	$B_{tcp,down}$
<i>start</i>	278 <i>byte</i>	1.011 <i>byte</i>
<i>login</i>	894 <i>byte</i>	2.306 <i>byte</i>
<i>query</i>	928 <i>byte</i>	2.333 <i>byte</i>

Tabelle 6.13: WAP-Nutzdaten der Transportschicht (Downlink)

Um die Effizienz des WAP-Gateways und des ASB-Dientes im besonderen zu prüfen, wurden auch die Nutzdaten der Protokolle WSP B_{wbxml} und HTTP B_{wml} auf dem Downlink gemessen, siehe Tab. 6.14. Während sich die WSP-Nutzdaten lediglich um 8 *byte* für die Kopfinformationen von WTP und WSP von der Summe der Nutzdaten auf der Transportschicht unterscheiden, ist der signifikante Unterschied zwischen den HTTP- und TCP-Nutzdaten in den umfangreichen Kopfinformationen des HTTP-Protokolls begründet.

Vorgang	B_{wbxml}	B_{wml}	Kodierungsrate ξ
<i>start</i>	270 <i>byte</i>	772 <i>byte</i>	65,0 %
<i>login</i>	886 <i>byte</i>	2.046 <i>byte</i>	56,7 %
<i>query</i>	920 <i>byte</i>	2.110 <i>byte</i>	56,4 %

Tabelle 6.14: WAP-Nutzdaten der Anwendungsschicht (Downlink)

Insgesamt werden beim ASB-Dienst zwischen 2 und 4 *kbyte* Daten über die Luftschnittstelle transportiert. Die geringe Datenmenge wird insbesondere durch die effiziente Kodierung

der WAP-Nachrichten erreicht. Abb. 6.17 zeigt die Kodierungsrate ξ in Abhängigkeit von der Größe B_{wml} der Dokumente, die als HTTP-Nutzdaten zum WAP-Gateway transportiert werden. Es wird gezeigt, dass die Kodierungsrate ξ für die Startseiten bekannter WAP-Adressen zwischen 25 und 65 % liegt. Die deutlichen Unterschiede liegen darin begründet, dass lediglich bekannte Tags und nicht der gesamte Seiteninhalt effizient kodiert werden können (Kap. 6.3.2). In Abb. 6.17 sind auch die Kodierungsraten ξ für verschiedene WAP-Nachrichten des ASB-Dienstes aufgetragen. Der Grafik ist zu entnehmen, dass die übertragenen Daten beim ASB-Dienst effizient kodiert werden.

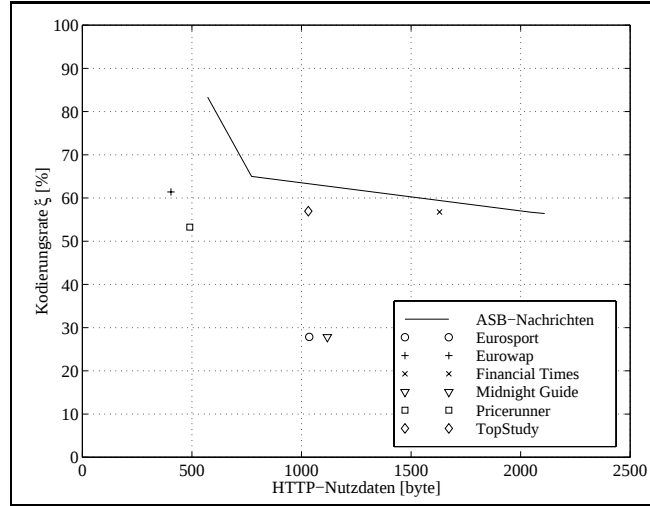


Abbildung 6.17: Kodierungsrate des WAP-Gateways für WML-Dokumente

Die Antwortzeit $T_{wap,csd}$ beschreibt die Dauer des WAP-Nachrichtenaustausches für ein Dokument (*start*, *login* oder *query*) zwischen dem mobilen Endgerät und der Kommunikationsplattform MCAP über eine CSD-Mobilfunkverbindung. Die wesentlichen Bestandteile der mittleren Antwortzeit $T_{wap,csd}$ sind die Verweilzeiten der Nachrichten auf der CSD-Mobilfunkstrecke V_{csd} und dem WAP-Gateway V_{wapgw} , sowie die Umlaufzeit T_{rtt} , die für den Transport der Nachrichten vom WAP-Gateway zum MCAP benötigt wird. Für WAP-Transaktionen bestehen die Verweilzeiten des Mobilfunkkanal und des WAP-Gateways jeweils zwei Anteilen, einen für den Uplink und einen für den Downlink. Daher wird auch die Verzögerung durch die Internetstrecke zweimal aufsummiert. Mit dem Parameter Δ_{wap} wird die mittlere Reaktionszeit des im MCAP integrierten WAP-Server von 150 ms berücksichtigt.

$$\begin{aligned}
 T_{wap,csd} &= V_{csd} + V_{wapgw} + 2 * T_{rtt} + \Delta_{wap} \\
 &= \frac{B_{udp,up} + B_{udp,down}}{b_{csd,wap}} + \theta + \eta * (B_{udp,up} + 8 \text{ byte}) + \theta + \eta * B_{wbxml} \\
 &\quad + 2 * T_{rtt} + \Delta_{wap} \\
 &= (B_{udp,up} + B_{udp,down}) * \frac{\eta}{b_{csd,wap}} + 2 * (\theta + T_{rtt} + 8 \text{ byte}) + \Delta_{wap} \quad (6.45)
 \end{aligned}$$

Abb. 6.18 zeigt die mittels Gleichung 6.45 berechneten mittleren Antwortzeiten T_{start} , T_{login} und T_{query} für die entsprechenden WAP/PHP-Seiten über CSD in Abhängigkeit vom Verkehrswert A_{bh} des WAP-Gateways. Die Antwortzeiten werden durch die

Verzögerungen auf der CSD-Mobilfunkstrecke und beim WAP-Gateway dominiert. Dabei wurden für die CSD-Mobilfunkstrecke und den WAP-Gateway die in Kap. 6.3 vorgestellten Modelle verwendet.

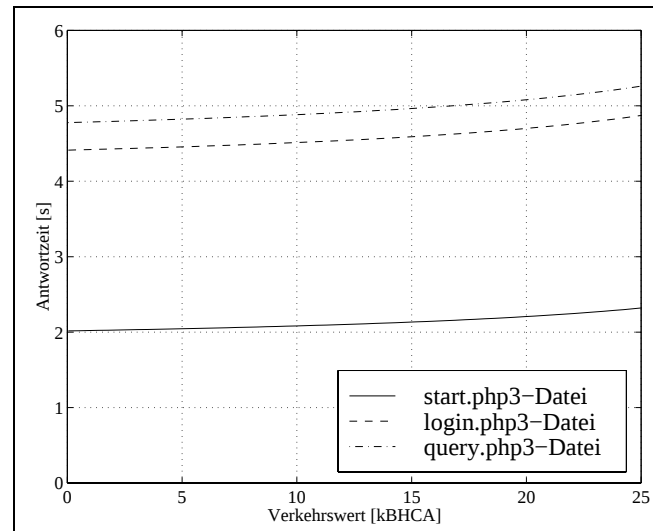


Abbildung 6.18: WAP-Antwortzeiten für CSD

Die Größe der WAP-Seiten und Nachrichten wird beim ASB-Dienst vor allem durch die Suchbeschreibung und das Suchergebnis beeinflusst, welche damit generell einen signifikanten Einfluss auf die Antwortzeit t_{wap} haben. Für den Abruf der WAP-Seiten über CSD werden je nach Dateigröße und Netzlast im Mittel zwischen 2 und 6 Sekunden benötigt. Die berechneten Werte wurden mittels umfangreicher Messungen verifiziert und bilden die realen Antwortzeiten t_{wap} sehr gut ab. Daraus folgt, dass die Transaktionsleistung $b_{csd,wap}$ eine sinnvolle Hilfsgröße zur Berechnung der Verweilzeit V_{csd} bei der WAP-Kommunikation über CSD ist. Der Einfluss der steigenden Netzlast bedingt durch eine höhere Ankunftsrate von Aufträgen ist eher gering, da für die WAP-Kommunikation über eine leitungsorientierte CSD-Verbindung genügend GSM-Kanäle mit jeweils relativ konstanten Transaktionsleistungen (Kap. 6.3.1) zur Verfügung stehen. Für die dennoch spürbar steigenden Antwortzeiten sind die Verweilzeiten der WAP-Nachrichten auf dem WAP-Gateway (Kap. 6.3.2) verantwortlich.

6.6.3 Datenvolumen

In diesem Abschnitt wird das ausgetauschte Datenvolumen zwischen der Kommunikationsplattform MCAP und den Händlerrechnern untersucht. Das von den stationären Suchagenten (SSA) übertragene Datenvolumen (TCP/IP-Verkehr) errechnet sich aus der Summe der SQL-Anfragen und -Antworten (Abb. 5.17). Abb. 6.19 zeigt den gemessenen TCP/IP-Verkehr für die SSAs in Abhängigkeit von der Anzahl der verfügbaren Händler. Im Mittel werden mit jedem Händler 1.529 byte ausgetauscht. Bei der angenommen Trefferwahrscheinlichkeit (Kap. 6.4.2) von 25 % zeigt sich, dass das Datenvolumen bei den Suchmethoden *ERSTES* und *DREI* ab einer bestimmten Händlerzahl nur noch geringfügig ansteigt. Dies liegt in der Tatsache begründet, dass bereits genügend Ergebnisse vorliegen und daher die restlichen Händler nicht aufgesucht werden. Bei der Suchmethode *BESTES* hingegen werden alle Händler besucht und das Datenvolumen steigt kontinuierlich weiter

an.

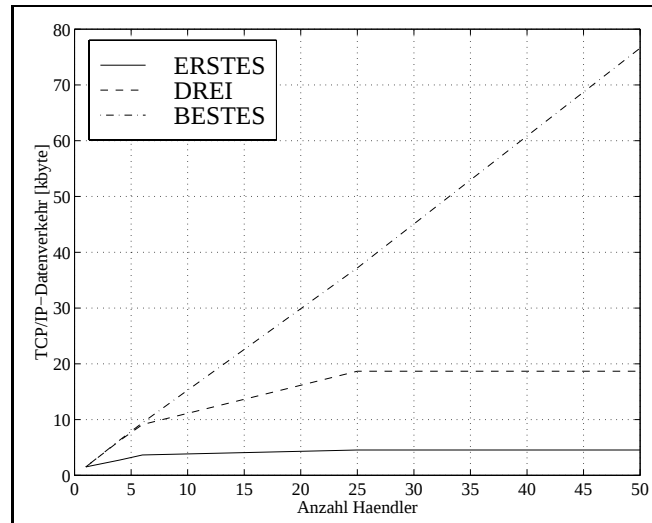


Abbildung 6.19: TCP-Datenverkehr für die statischen Suchagenten

Zum Vergleich der Datenvolumina zwischen SSAs und den mobile Suchagenten (MSA bzw. MSA+), welche für die Agentenmigration von der Kommunikationsplattform MCAP zum ersten Händler und vom letzten besuchten Händler zurück zum MCAP anfallen, wurden für die SSAs zwei unterschiedliche Trefferwahrscheinlichkeiten angenommen, die in Tab. 6.15 aufgelistet sind.

Trefferwahrscheinlichkeit	
P_{e1}	25 %
P_{e2}	10 %

Tabelle 6.15: Untersuchte Trefferwahrscheinlichkeiten beim ASB-Dienst

Abb. 6.20 bis 6.22 vergleichen nun die Datenvolumina der MSAs mit denen der SSAs für die Suchmethoden *ERSTES*, *DREI* und *BESTES* (Kap. 5.4.2). Es wird gezeigt, dass die Datenvolumina der MSAs sich in Abhängigkeit von der Suchmethode kaum verändern. Die Suchmethode entscheidet über die Anzahl der besuchten Händler, was lediglich bei der Nutzung von SSAs direkten Einfluss auf das Datenvolumen, welches der MCAP austauscht, hat. Es wird deutlich, dass die MSAs im Hinblick auf die zwischen der Kommunikationsplattform MCAP und den Händlern übertragene Datenvolumen im Vorteil sind, sobald eine Vielzahl von Händlern aufgesucht werden. Die Werte für die Suchmethode *SMART* wurde nicht aufgetragen, da sich dort für den SSA pro Händler schnell etwa 10 kbyte Daten ansammeln, so dass das Ergebnis für die SSAs deutlich aus dem Rahmen fällt. Dies zeigt, dass die SSAs, bedingt durch die Unfähigkeit, Daten vor Ort zu filtern, den MSAs auch bei komplexen Aufgaben deutlich unterlegen sind.

Für die Berechnungen der mittleren Antwortzeiten wurden die in Tab. 6.16 gezeigten gemessenen Daten verwendet. Dabei wird angenommen, dass der eigentliche Programmcode des MAs B_{code} auf den meisten (99 %) der Händlerrechner schon vorhanden ist, da die Händler regelmäßig von MAs aufgesucht werden. Daraus folgt, dass hauptsächlich der

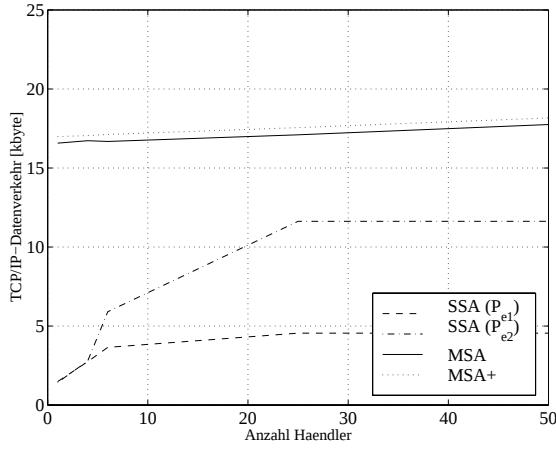


Abbildung 6.20: Datenvolumen bei Suchmethode ERSTES

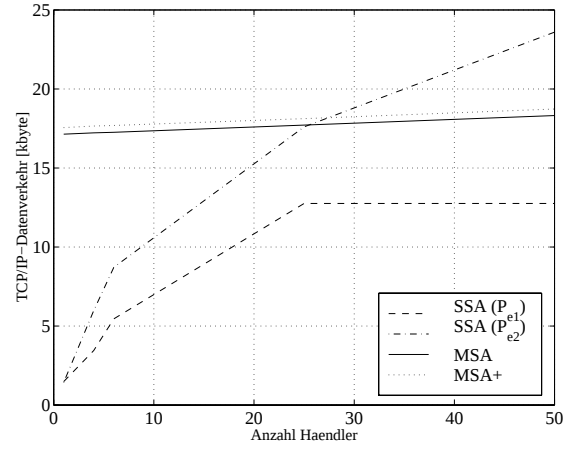


Abbildung 6.21: Datenvolumen bei Suchmethode DREI

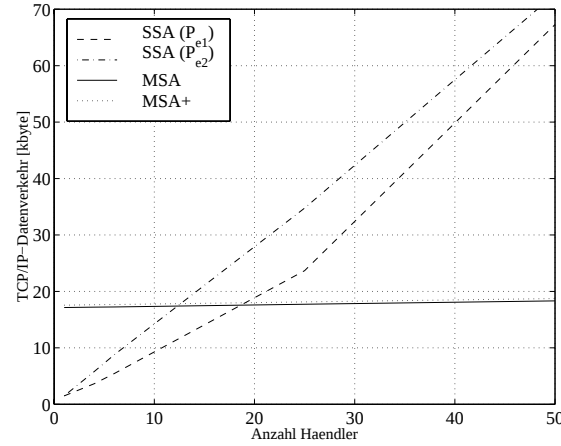


Abbildung 6.22: Datenvolumen bei Suchmethode BESTES

Datenspeicher B_{data} sowie die serialisierten Zustandsinformationen B_{state} zwischen dem MCAP und den Händlern übertragen werden.

Die Größe des Datenspeichers B_{data} wird durch das Suchergebnis und die Händlerliste bestimmt. Die Datenmenge eines Suchergebnisses B_{res} hängt vom gesuchten Produkt, den Suchkriterien und dem Wunsch nach Zusatzinformationen wie Verfügbarkeit, Lieferung, etc. ab. Tab. 6.17 zeigt die per Messung ermittelten durchschnittlichen TCP-Nutzdaten B_{res} für Suchergebnisse, welche in die verkehrstheoretische Analyse eingeflossen sind. Dabei ist die Größe eines Ergebnisses bei der Suchmethode *SMART* doppelt so groß wie bei den übrigen Suchmethoden, da eine Reihe von Suchkriterien abgefragt und übermittelt werden. Während bei der Suchmethode *ERSTES* nur ein Suchergebnis gefunden und übermittelt wird, werden bei den Suchmethoden *DREI* und *BESTES* drei Suchergebnisse übertragen.

Neben der Größe des Suchergebnisses muss auch berücksichtigt werden, dass der MCAP dem MA eine Händlerliste übergibt. Diese Händlerliste bestimmt die Migrationsstrategie

Parameter	Größe
$B_{code,msa}$	16.797 <i>byte</i>
$B_{code,msa+}$	18.797 <i>byte</i>
B_{cr}	100 <i>byte</i>
$B_{state,msa}$	8.132 <i>byte</i>
$B_{state,msa+}$	8.340 <i>byte</i>
P_{code}	0,01

Tabelle 6.16: MSA-Agentenparameter

ASB-Suchmethode	Suchergebnisgröße B_{res}
ERSTES, DREI und BESTES	285 <i>byte</i>
SMART	570 <i>byte</i>

Tabelle 6.17: Größe der Suchergebnisse

des MSAs, d.h. zu Beginn der Migration wird die Migrationsstrecke vom MCAP festgelegt. Der MCAP beobachtet dabei den Netzzustand durch Stichproben (Kap. 5.1.4). Eine dynamische netzzustandsabhängige Migration, bei der der MSA vor jeder Migration neu entscheidet, wohin er migriert, erfolgt nicht. Allerdings entscheidet der MSA selbständig anhand der vorliegenden Ergebnisse, ob er die Suche fortsetzt oder zurück zu seiner Heimat, dem MCAP, migriert. Tab. 6.18 zeigt die Datenvolumina B_{mlist} für diese Händlerliste. Eine Händleradresse ist dabei eine im Mittel 24 *byte* große URI¹⁶.

Anzahl Händler	1	4	6	25	50
B_{mlist} in [<i>byte</i>]	24	96	144	600	1200

Tabelle 6.18: Größe der Händlerliste B_{mlist}

6.6.4 Selektivität

Das mittlere Volumen des Ergebnisspeichers B_{data} des zurückkehrenden MSA kann angelehnt an Gleichung 6.36 durch Gleichung 6.46 berechnet werden, wobei n_{vm} für die Anzahl der besuchten Händler steht.

$$B_{data} = (1 - \gamma) * n_{vm} * \{B_{res} + B_{mlist}\} \quad \forall 0 \leq \gamma \leq 1 \quad (6.46)$$

Mit Gleichung 6.46 und den Messergebnissen der Datenvolumina B_{data} , B_{res} und B_{mlist} konnte die Selektivität γ in Abhängigkeit von der Anzahl der besuchten Händler n_{vm} ermittelt werden, siehe Abb. 6.23.

Die hohe Selektivität der Agenten bei den Händlern bewirkt, dass die zu transportierende Datenmenge B_{data} erheblich reduziert wird.

¹⁶Uniform Resource Identifier, generelles Adressierungsschema für Internetressourcen.

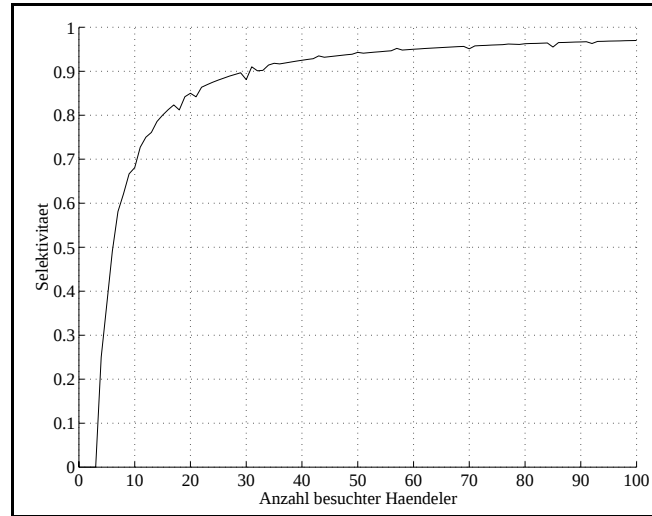


Abbildung 6.23: Selektivität γ eines mobilen Suchagenten in Abhängigkeit von der Anzahl der besuchten Händler

6.6.5 Antwortzeiten

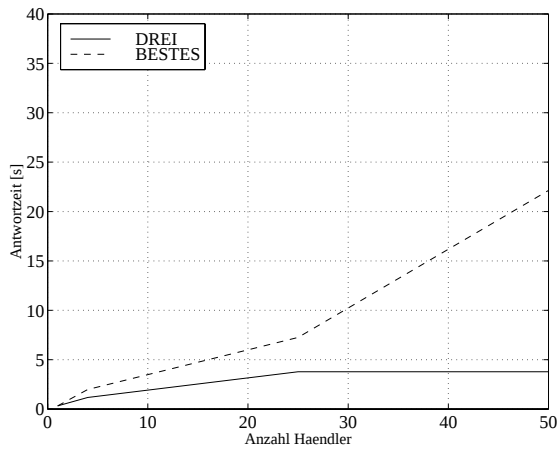
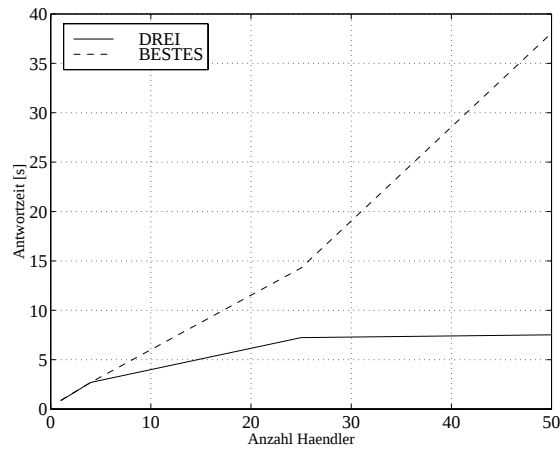
Anhand von Messungen konnten für den ASB-Dienst die in Tab. 6.19 dokumentierten mittleren Antwortzeiten für die Produktsuche T_{ps} (Abb. 6.11) bei Wahl der Suchmethode *BESTES* bei vier Händler ermittelt werden. Die Antwortzeiten verdeutlichen, dass in einem Laborsystem der statische Agent (SSA) und dementsprechend der klassische RPC-Ansatz den MSAs im Hinblick auf die Antwortzeit deutlich überlegen ist. Vorteilhaft erscheinen die MAs lediglich aufgrund des geringen Variationskoeffizienten der Produktsuchzeit T_{ps} . Allerdings haben diese quantitativen Messungen nur eine eingeschränkte Bedeutung, da sie nur für ein Laborsystem mit vier Händlern und geringen Verkehrswerten Gültigkeit besitzen. Außerdem spiegeln diese Messungen die qualitativen Vorteile der MSAs nicht wieder.

Agententyp	T_{ps}	$var(t_{ps})$	$\sigma(t_{ps})$	c_{ps}
MSA+	3.377 ms	43.825 ms ²	209 ms	0,097
MSA	2.698 ms	20.781 ms ²	144 ms	0,111
SSA	1.256 ms	6.377 ms ²	80 ms	0,512

Tabelle 6.19: Antwortzeiten beim ASB-Dienst

Die Abb. 6.24 und 6.25 zeigen die gemessenen Auswirkungen bei den Suchmethoden *DREI* und *BESTES* für den SSA bzw. den MSA, wenn bis zu 50 Händler aufgesucht werden. Es wird deutlich, dass bei der Suchmethode *DREI* sowohl beim SSA, als auch beim MSA schnell die Suche abgebrochen wird und relativ unabhängig von der Händlerzahl mit Antwortzeiten von wenigen Sekunden gerechnet werden kann. Die trotzdem ablesbare Zunahme der Antwortzeit bei steigender Händlerzahl ist in der größeren Händlerliste $B_{m\text{list}}$ begründet. Allerdings haben auch diese Messungen nur für die geringe Verkehrslast im Laborsystem Gültigkeit.

Ausgehend von den in Kap. 6.3 vorgestellten Modellen und Bedienzeiten wird die Antwortzeit t_{ps} nun mathematisch modelliert und kann mit den vorgenannten Messwerten

Abbildung 6.24: Antwortzeit T_{ps} (SSA)Abbildung 6.25: Antwortzeit T_{ps} (MSA)

überprüft werden. Die Abb. 6.26 bis 6.29 zeigen die mit dem in Abb. 6.16 dargestellten verkehrstheoretischen Modell berechneten Antwortzeiten bei der Produktsuche des ASB-Dienstes für die verschiedenen Suchmethoden in Abhängigkeit vom Verkehrswert A_{bh} , wenn 25 Händlern das gesuchte Produkt anbieten und die Trefferwahrscheinlichkeit bei 25 % liegt. Erneut wird deutlich, dass bei den Suchmethoden ERSTES, DREI und BESTES die Antwortzeiten für die beiden mobilen Agenten (MSA und MSA+) erheblich länger als für die stationären Agenten (SSA) sind. Besonders groß ist der Unterschied bei den mobilen Agenten mit XML-Schnittstelle (MSA+), denn für die XML-Konvertierung muss ein zusätzlicher Rechenaufwand berücksichtigt werden.

Da alle SSAs nach jeder Anfrage wieder von der Kommunikationsplattform MCAP bearbeitet werden, stellen bei der Nutzung von SSAs in der Regel nicht die Händlerplattformen, sondern der MCAP den Flaschenhals im Gesamtsystem dar. So kann ein MCAP in untersuchten Fällen unabhängig von der Anzahl der besuchten Händler und der Suchmethode lediglich Verkehrswerte bis zu ca. 11.500 *BHCA* für MSAs bzw. 9.000 *BHCA* für MSAs mit XML-Schnittstelle verarbeiten. Die Höchstgrenze an SQL-Anfragen, d.h. an SSAs, die der MCAP verkraftet, ist hingegen abhängig von der Anzahl der besuchten Händler. Man erkennt, dass der Verkehrswert bei dem der MCAP für die SSAs (gestrichelte Linie) in Sättigung geht von der Suchmethode abhängt. Bei der Suchmethode BESTES, bei der naturgemäß nicht nur ein Teil der Händler, sondern alle Händler aufgesucht werden, setzt die Sättigung des MCAPs schon bei weniger als 6.000 *BHCA* ein.

Noch deutlicher verändert sich der Graph für die SSAs im Fall der Suchmethode SMART. Aufgrund der relativ großen Datenmenge und des Einflusses der Selektivität γ sind die MAs in Abb. 6.29 bereits bei 25 Händlern den stationären Agenten nicht nur im Hinblick auf die Antwortzeit, sondern auch beim möglichen Verkehrswert A_{bh} überlegen. Dies beweist, dass der Einsatz von MAs besonders vorteilhaft ist, wenn komplexere Anfragen getätigt werden und die Ergebnisse bereits vor Ort (lokal) ausgewertet und gefiltert werden können. Ein einfacher Nachrichtenaustausch hingegen führt dazu, dass eine Kommunikationsplattform wie der MCAP verhältnismäßig schnell in Überlast gerät.

Abb. 6.30 zeigt den Trend im Detail. Hier sind die Verkehrswerte A_{bh} , die eine Last

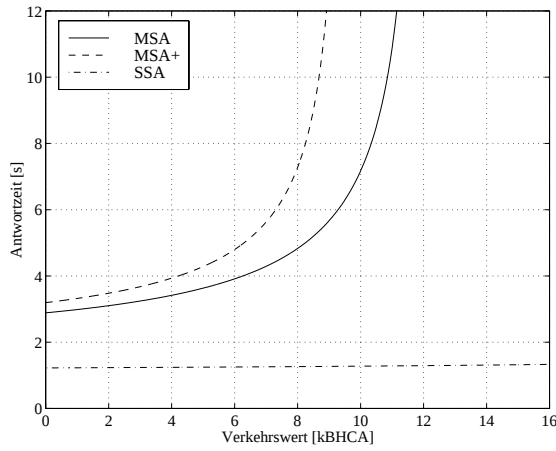


Abbildung 6.26: Suchmethode ERSTES

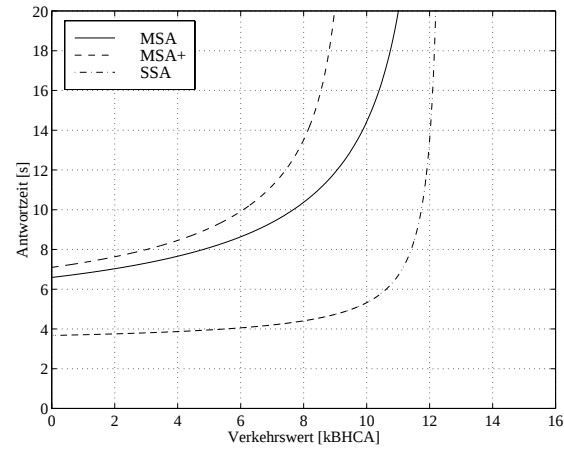


Abbildung 6.27: Suchmethode DREI

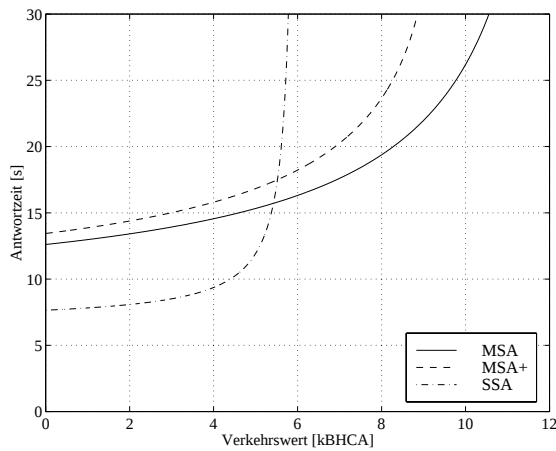


Abbildung 6.28: Suchmethode BESTES

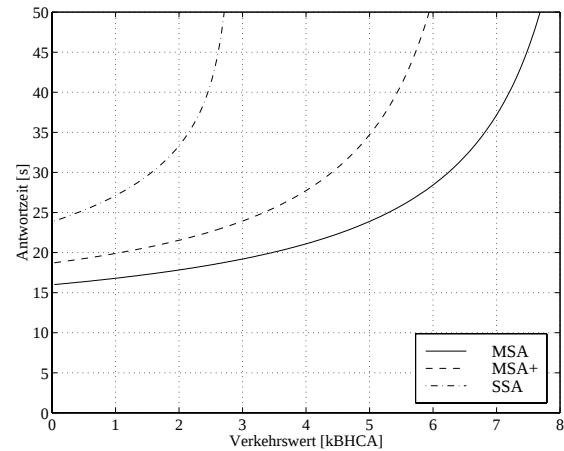


Abbildung 6.29: Suchmethode SMART

von 90 % im MCAP hervorrufen, in Abhängigkeit von der Anzahl der besuchten Händler aufgetragen. Während die Werte für die MSAs konstant bei 10.692 *BHCA* bzw. 8.586 *BHCA* liegen, sinkt die Anzahl der bedienbaren Aufträge beim SSA je mehr Händler besucht werden. Daraus resultiert, dass ab einer gewissen Anzahl von besuchten Händlern die MSAs deutlich im Vorteil sind. Ein weiterer Pluspunkt für die mobilen Agenten ist daher, dass die auftretende Prozessorlast des MCAPs leichter vorhersagbar ist, da sie in der Regel unabhängig von der Anzahl der besuchten Händler ist, die ohnehin vor einer Suche nur schwer prognostiziert werden kann. Die maximale Anzahl von Teilnehmern lässt sich mit Gleichung 6.28 berechnen. Nimmt man an, dass von 1.000 Teilnehmern acht Mobilfunkbenutzer in der Hauptverkehrsstunde eine Anfrage starten, so kann der MCAP im Falle der mobilen Agenten unabhängig von der Suche ca. 1,3 Millionen (MSA) bzw. 1,1 Millionen (MSA+) Teilnehmer versorgen. Für die SSAs lässt sich hingegen keine konkrete Aussage machen. Generell besteht die Möglichkeit, die Anzahl der aufgesuchten Händler durch Wahl der Suchmethoden *ERSTES* oder *DREI* gering zu halten.

Geht man von der in Tab. 6.20 dargestellten Verteilung für die unterschiedlichen Suchmethoden aus, so ist eine Voraussage für die mittlere Antwortzeit T_{ps} beim ASB-Dienst möglich, welche in Abb. 6.31 dargestellt ist. Es zeigt sich, dass die berechneten Ant-

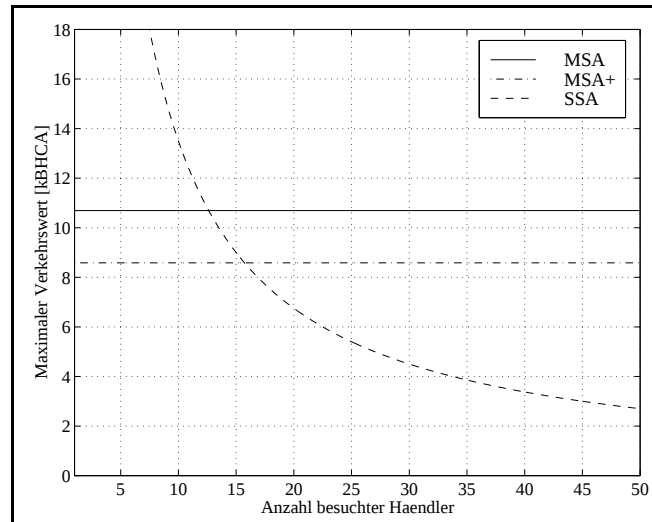


Abbildung 6.30: Verhältnis Ankunftsrate des MCAPs zur Händleranzahl

Suchmethode	Anteil
ERSTES	15%
DREI	30%
BESTES	35%
SMART	20%

Tabelle 6.20: Dienstmix

wortzeiten nur gering voneinander abweichen, und dass der ursprünglich im Laborsystem gemessene Suchzeitvorteil der SSAs in einem realen System durch die verschiedenen Suchmethoden und eine höhere Anzahl von Händlern kompensiert werden wird.

6.6.6 Dienstaussführungszeit

Nachdem in den vorangegangenen Abschnitten die mittlere Antwortzeit T_{ps} des ASB-Dienstes bei der Produktsuche analysiert worden ist, wird nun die mittlere Dienstaussführungszeit S_{asb} für die Produktsuche untersucht, die sich aus den in Gleichung 6.47 genannten Komponenten zusammensetzt.

$$S_{asb} = \sum T_i = T_{cs,wap} + T_{start} + T_{user,login} + T_{login} + T_{user,query} + T_{query} + T_{ps} \quad (6.47)$$

Abb. 6.32 zeigt, dass die Produktsuche über eine CSD-Datenverbindung inklusive der Benutzerinteraktionen beim Besuch von 15 Händlern (Suchmethode *BESTES*) zwischen 65 und 80 Sekunden dauern wird. Auch wenn dieser Wert durchaus akzeptabel klingt, so ist auffällig, dass ähnlich wie beim AHB-Dienst ein bedeutender Anteil der Dienstaussführungszeit S_{asb} für den Auf- und Abbau der Datenverbindung T_{cs} benötigt wird. Im Vergleich zum AHB-Dienst kann außerdem festgehalten werden, dass eine WAP-Datenverbindung deutlich schneller als eine TCP/IP-Datenverbindung aufgebaut wird.

In Abb. 6.33 wird die mittlere Dienstaussführungszeit in Abhängigkeit von der Anzahl besuchter Händler untersucht. Da im Verkehrsmodell des ASB-Dienstes für die Rechner der

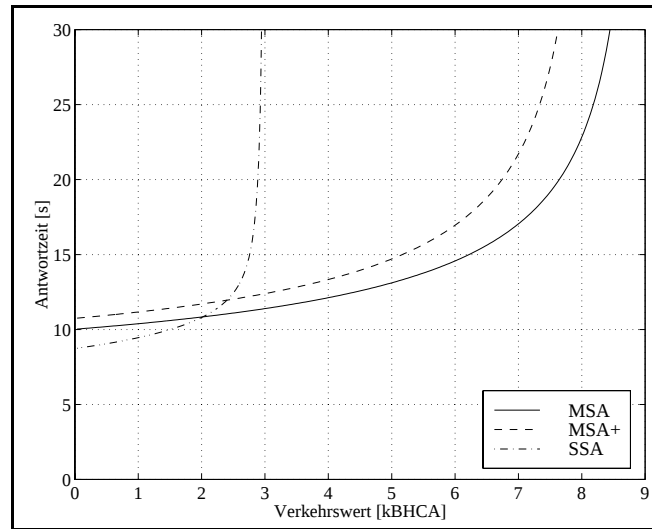


Abbildung 6.31: Mittlere Antwortzeit T_{ps} beim Dienstmix

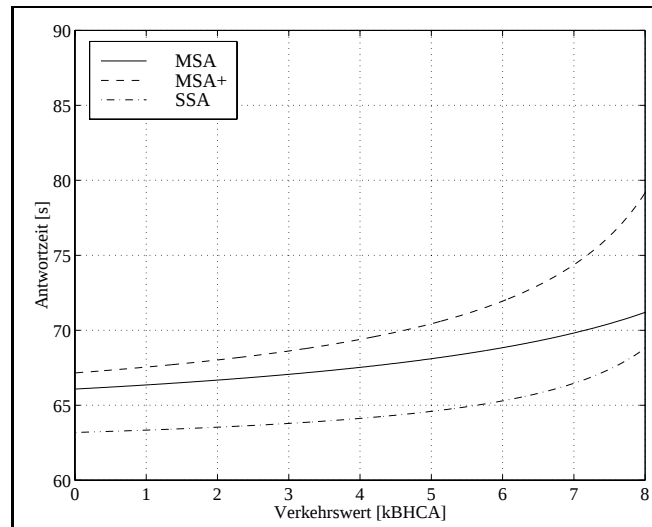


Abbildung 6.32: Mittlere Dienstaussführungszeit S_{asb} in Abhängigkeit vom Verkehrswert

Händler dieselbe Belastung angenommen ist, ist der Anstieg der Dienstaussführungszeit für die MSA linear.

Berücksichtigt man schließlich den Fall, dass der Teilnehmer den gesuchten Artikel nicht nur bestellen, sondern auch bezahlen möchte, so kann für eine agentenbasierte SET-Transaktion von einer weiteren Verzögerung von etwa 35 Sekunden ausgegangen werden [CAM98, Lim98, Hüc00].

6.6.7 WAP-Kommunikation über GPRS

In diesem Abschnitt werden die Auswirkungen auf die mittleren WAP-Antwortzeiten T_{wap} des ASB-Dienstes bei Verwendung einer paketorientierten Mobilfunkverbindung nach dem GPRS-Standard anstatt einer leitungsorientierten CSD-Übertragung untersucht. Beim ASB-Dienst werden WAP-Nachrichten zwischen dem mobilen Endgerät und dem MCAP für die Ein- und Ausgabe der Informationen ausgetauscht. Wie in Kap. 6.6.2 beschrieben

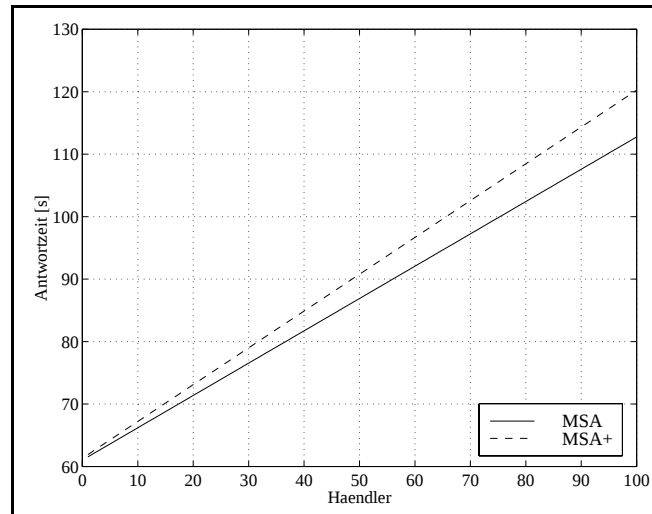


Abbildung 6.33: Mittlere Dienstaussführungszeit S_{asb} in Abhängigkeit von der Händlerzahl

wurde, dauert die Anfrage und die Übermittlung der drei WAP-Seiten (*start*, *login* und *query*) über CSD insgesamt über 10 Sekunden. Zusätzlich muss zum Aufbau der WAP-Verbindung über CSD eine Verzögerung von rund 16 Sekunden berücksichtigt werden. Die Nutzung des GPRS-Standards sollte es erlauben, die entsprechend Antwortzeiten zu verringern.

Die in diesem Kapitel vorgestellten Ergebnisse basieren auf Messungen mit dem GPRS-Simulator GATE (Kap. 6.2.3), sowie auf ersten Messungen mit handelsüblichen GPRS-Mobiltelefonen. Zunächst wurden die in Kap. 6.3.1 beschriebenen vier verschiedenen Übertragungsqualitäten in GATE eingestellt. Da die simulierten GPRS-Verweilzeiten sehr klein sind und daher die Antwortzeit $t_{wap,gprs}$ nicht mehr dominieren, werden diese Zeiten isoliert betrachtet. Tab. 6.21 zeigt die simulierten Verweilzeiten auf der GPRS-Mobilfunkstrecke V_{gprs} für die drei verschiedenen vom ASB-Dienst verwendeten WAP-Dokumente.

Vorgang	Best Case	Normal Case	Limited Case	Worst Case
<i>start</i>	245 ms	275 ms	281 ms	337 ms
<i>login</i>	431 ms	454 ms	461 ms	490 ms
<i>query</i>	428 ms	437 ms	479 ms	519 ms

Tabelle 6.21: Simulierte mittlere Verweilzeit V_{gprs} bei der WAP-Kommunikation

Die daraus resultierenden Transaktionsleistungen liegen zwischen 1.000 und 3.000 *byte/s*. Diese Durchsätze liegen zwar deutlich unter den theoretisch mögliche Übertragungsraten von GPRS, stellen im Vergleich mit der im Rahmen dieser Arbeit ermittelten CSD-Transaktionsleistung $B_{csd,wap}$ (279 *byte/s*) eine erhebliche Verbesserung dar. Die starken Schwankungen für die verschiedenen Dienstqualitäten und unterschiedlichen Nachrichten-größen zeigen auch, dass die Transaktionsleistung kein geeignetes Modell für die Beschreibung einer GPRS-Mobilfunkstrecke ist.

Erste reale Messungen mit einem GPRS-Mobiltelefon (Ericsson R520), welches vier bzw. einen GPRS-Zeitschlitz für den Down- bzw. Uplink unterstützt, konnten die durch Simu-

lation gewonnenen Ergebnisse nicht bestätigen. Tab. 6.22 zeigt die gemessenen mittleren Antwortzeiten T_{wap} bei der verbindungsorientierten WAP-Kommunikation für GPRS¹⁷ im Vergleich mit CSD. Bringt man von diesen Antwortzeiten die modellierten Verzögerungen des WAP-Gateways, der Internetstrecke und des WAP-Servers in Abzug, so ergeben sich aus verbleibenden Verweilzeiten V_{gprs} Transaktionsleistungen zwischen 300 und 500 *byte/s*. Ein für den Benutzer spürbarer Zeitvorteil tritt bei der WAP-Kommunikation gegenüber einer CSD-Verbindung lediglich bei größeren WAP-Seiten (z.B. *login* oder *query*) auf.

Vorgang	GPRS	CSD
<i>start</i>	1.882 <i>ms</i>	2.080 <i>ms</i>
<i>login</i>	2.804 <i>ms</i>	4.413 <i>ms</i>
<i>query</i>	3.025 <i>ms</i>	4.778 <i>ms</i>

Tabelle 6.22: Vergleich der mittleren Antwortzeiten T_{wap} bei CSD und GPRS

Die verglichen mit den möglichen Übertragungsraten und den Ergebnissen der Simulation relativ geringe Transaktionsleistung bei der WAP-Kommunikation mit heutigen GPRS-Mobiltelefonen ist insbesondere in den langen Verzögerungen für den Auf- und Abbau des TBF-Kontextes (Kap. 6.5.5) begründet, der für jede WAP-Nachricht auf dem Up- und auf dem Downlink neu initiiert werden muss. Da GATE im Hinblick auf Web-Verkehre und nicht auf WAP-Verkehre entwickelt worden ist, werden die TBF-Verzögerungen nicht ausreichend berücksichtigt und die in Tab. 6.21 dargestellten Verweilzeiten können bisher nicht erreicht werden.

Betrachtet man nun die Dienstausführungszeit des ASB-Dienstes, so macht sich der Wegfall der Antwortzeit für den Auf- und Abbau der Datenverbindung $T_{cs,wap}$ besonders deutlich bemerkbar. Abb. 6.34 zeigt mittlere Dienstausführungszeit $S_{asb,gprs}$ in Abhängigkeit von der Anzahl besuchter Händler.

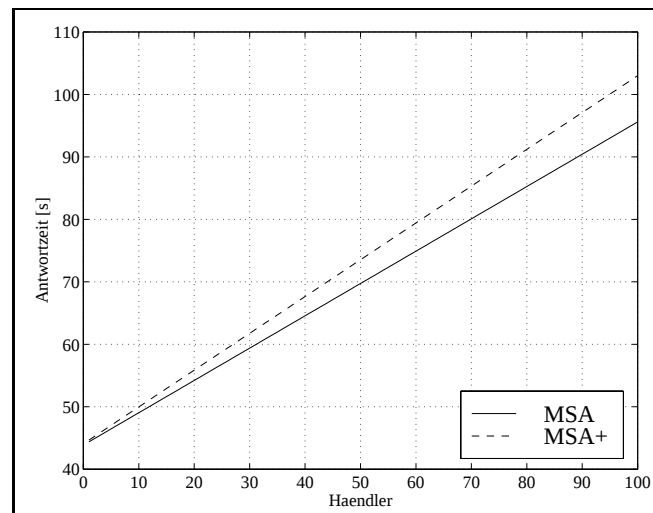


Abbildung 6.34: Mittlere Dienstausführungszeit S_{asb} bei GPRS

Neben der Zeitersparnis liegt ein weiterer Vorteil der GPRS-Nutzung darin, dass Mobil-

¹⁷Das verwendete GPRS-Netz (D2-Vodafone) nutzt das Kodierungsschema CS-2 (Kap. 2.3.4).

funknetzbetreiber die GPRS-Dienstnutzung hauptsächlich nach dem übertragenden Datenvolumen abrechnen. Dies ist besonders vorteilhaft für den prototypisch realisierten ASB-Dienst, da dieser lediglich geringe Datenmengen über die Mobilfunkstrecke austauscht.

Zusammenfassung und Ausblick

Bei den erfolgreichen und weitverbreiteten GSM-Mobilfunknetzen handelt es sich um geschlossene Kommunikationsnetze, die lediglich über das separate Vermittlungsnetz SS7 mit anderen Netzen verbunden sind. Parallel zur kontinuierlich steigenden Nachfrage nach Mobilfunkdiensten hat sich binnen weniger Jahre das Internet zum weltweit meistgenutzten, multimedialen offenen Kommunikationsnetz entwickelt. In der vorliegenden Arbeit wurde ein agentenbasiertes Kommunikationskonzept für Mobilfunknetze entworfen und bewertet, dass eine effiziente Nutzung des Internets von Mobiltelefonen aus und damit eine offene Dienstleistung ermöglicht.

Im Mittelpunkt dieses Kommunikationskonzeptes steht die agentenbasierte Kommunikationsplattform für M-Commerce-Dienste (MCAP). Nach einem Vergleich existierender Agentensysteme wurde für den neu entwickelten MCAP das Agentensystem Voyager ausgewählt. Voyager unterstützt den Einsatz von mobilen Softwareagenten, die von ihrem Benutzer mit einem Auftrag ausgestattet werden und anschließend eigenständig zwischen Netzknoten migrieren können. Damit erlaubt der MCAP eine Vielzahl von neuartigen Kommunikations- und Geschäftsmöglichkeiten. Beispielhaft wurden zwei massenmarktfähige E-Commerce-Dienste für Mobilfunkteilnehmer entworfen und implementiert, die einen Zugang zum Internet benötigen. Der Erfolg von Agentensystemen hängt insbesondere davon ab, dass die Systeme verschiedener Hersteller zusammenarbeiten können. Für die Kommunikationsplattform MCAP wurde daher ein sogenannter MASIF-Port konzipiert und implementiert, der eine Zusammenarbeit zwischen verschiedenen Agentensystemen basierend auf dem CORBA- und MASIF-Standard derart erlaubt, dass Dienste eines fremden Agentensystems bekannt gemacht werden können. Es hat sich gezeigt, dass eine darüber hinausgehende Zusammenarbeit zwischen verschiedenen Agentensystemen aufgrund deren unterschiedlichen Konzeptionen nur mit erheblichen Aufwand zu erreichen ist. Neben dem MASIF-Port unterstützt der MCAP auch unabhängig vom verwendeten Trägerdienst das Bankprotokoll HBCI, den WAP-Standard und aus Sicherheitsgründen das SSL-Protokoll.

Bei dem im Rahmen dieser Arbeit realisierten agentenbasierten Homebankingdienst (AHB) werden rund 5 *kbyte* große mobile Agenten auf der Luftschnittstelle eingesetzt, um Daten effizient zum MCAP zu übertragen. Der MCAP selbst nutzt das HBCI-Protokoll für die Kommunikation mit einem Bankinstitut. Die Personalisierung des AHB-Dienstes, sowie die Dienstportabilität wird durch die Übertragung eines allgemeinen Dienstagenten zum verwendeten mobilen Endgerät in relativer einfacher Art und Weise ermöglicht. Weiterhin wurde nachgewiesen, dass die Antwortzeiten des MCAPs gut mittels einer definierten Transaktionsleistung $b_{csd,tcp}$ für Nutzdaten der Transportschicht modelliert werden können. Dabei hat sich herausgestellt, dass sich die Antwortzeiten und Kommunikationskosten für Banktransaktionen durch die Übertragung eines mobilen Agenten im Vergleich zu einer klassischen Ende-zu-Ende HBCI-Lösung, bei der zahlreiche Nachrichten ausgetauscht werden müssen, deutlich verringern. Möchte der Benutzer mehrere Bankaufträge gleichzeitig tätigen, so steigt die Dienstaussführungszeit bei der Agentenlösung nur unwesentlich, während sie sich bei der klassischen HBCI-Lösung nahezu proportional zur Auftragsanzahl

erhöht. Schließlich wurde gezeigt, dass sich die entwickelte Lösung sehr gut für paketorientierte Mobilfunksysteme wie GPRS eignet, da die verwendeten mobilen Agenten bei diesen Systemen mittels weniger TCP/IP-Pakete und ohne lange Verzögerungen für den Aufbau einer Datenverbindung effizient übertragen werden können.

Mittels des neu entworfenen agentenbasierten Such- und Bestelldienstes (ASB) kann eine schnelle und komfortable Suche nach Konsumgütern und Dienstleistungen per Mobiltelefon im Internet durchgeführt werden. Um GSM-Teilnehmern eine bedienerfreundliche, personalisierte und dynamische Benutzerschnittstelle anbieten zu können, wurden der WAP-Standard und die Programmiersprache PHP eingesetzt, welche beide vom MCAP unterstützt werden. Weiterhin wurden für den ASB-Dienst drei verschiedene Dienstszenarien mit jeweils vier Suchmethoden konzipiert, implementiert und modelliert. Mit Hilfe des Dekompositionsverfahrens wurden alle bei der Dienstleistung involvierten Netzknoten analysiert und ein Verkehrsmodell für die mobilen Agenten aufgestellt, welches durch Messungen verifiziert wurde. Während sich in einem Laborsystem mit wenigen Händlern der Einsatz von mobilen Agenten aus Leistungsgründen nicht rechtfertigt, konnte anhand des Verkehrsmodells gezeigt werden, dass mobile Agenten überlegen sind, sobald komplexere Anfragen getätigt werden, eine Vielzahl von Händlern aufgesucht wird und die Ergebnisse bereits vor Ort beim Händler ausgewertet und gefiltert werden können. Der Datenspeicher der untersuchten mobilen Suchagenten zeichnet sich durch eine hohe Selektivität aus, wodurch die Datenmenge, die der MCAP behandeln muss, erheblich reduziert wird. Eine Reduktion der Datenmenge wird auch durch die ringförmige Kommunikation der mobilen Agenten im Vergleich zur sternförmigen Kommunikation statischer Ansätze erreicht. Es wurde nachgewiesen, dass dadurch die Prozessorlast des MCAPs verringert, so dass der MCAP beim Einsatz von mobilen Agenten mehr Teilnehmer und einen höheren Verkehrswert bedienen kann als bei der Verwendung von statischen SQL-Anfragen. Außerdem wird eine Lastunabhängigkeit von der besuchten Händleranzahl erreicht, d.h. die Prozessorlast wird auf alle involvierten Netzknoten verteilt. Die Summe der Untersuchungsergebnisse der beiden E-Commerce-Dienste unterstreicht den Nutzen des agentenbasierten Kommunikationskonzeptes für Mobilfunknetze.

Schließlich wurde gezeigt, dass die beim ASB-Dienst zu transportierende Datenmenge auf der Mobilfunkstrecke durch die Komponenten des WAP-Standards erheblich reduziert wird. So liegt der Kodierungsfaktor des WAP-Gateways ξ beim ASB-Dienst deutlich über 50 Prozent. Die Übertragung von WAP-Nachrichten über GSM-Datenverbindungen konnte mit Hilfe der eingeführten Transaktionsleistung $b_{csd,wap}$ gut abgebildet werden. Für die Untersuchung des GPRS-Datendienstes, der keine konstante Datenrate anbietet, wurden vier Übertragungsklassen definiert, denen verschiedene Dienstgüteparameter und Rahmenbedingungen zugeordnet worden sind. Die Analyse hat gezeigt, dass die Transaktionsleistung für die Beschreibung des Durchsatzes von WAP-Nachrichten bei GPRS ungeeignet ist, da die Verweilzeiten für die relativ kompakten WAP-Nachrichten (50 bis 1.500 *byte*) hauptsächlich durch nachrichtenunabhängige Faktoren wie Bedienzeiten des Web-/WAP-Servers und des WAP-Gateways, sowie die Auf- und Abbaupzeiten des TBF-Kontextes beeinflusst werden. Dennoch wurden deutliche Leistungsvorteile bei der Nutzung des GPRS-Dienstes für die WAP-Kommunikation festgestellt. So kann aufgrund des Wegfalls der langen Verzögerung für den Kanalaufbau die Dienstausführungszeit erheblich verkürzt werden. Vorteilhaft wird sich die Nutzung von GPRS auch bei den Kommunikationskosten auswirken, da GPRS-Dienste hauptsächlich nach dem übertragenen Datenvolumen abgerechnet werden und nicht nach der Onlinezeit.

Es bestätigte sich, dass die Agententechnologie einen sinnvollen Rahmen für die verteilte, flexible und informationsorientierte Dienstleistung in Mobilfunknetzen bietet, da die Kommunikation nach dem Agentenparadigma ein hohes Maß an Flexibilität besitzt und eine effiziente Ressourcennutzung im Hinblick auf Zeit, Datenmenge, Kosten und Prozessorleistung erlaubt. Um diese Vorteile in Mobilfunknetzen optimal nutzen zu können, wird jedoch ein gesonderter Dienstknoten benötigt, der die Vorteile des Agentenparadigmas mit Mobilfunkstandards wie WAP und GPRS synergetisch verknüpft. Dabei hilft der GPRS-Standard die Onlinezeit für mobile Datendienste erheblich zu verkürzen.

Ausblick

In dieser Arbeit wurden Fragestellungen zu mobilen Agenten und zukünftigen agentenbasierten Kommunikationskonzepten für Mobilfunksysteme diskutiert und bewertet. Bevor man jedoch den Einsatz von mobilen Agenten in Mobilfunknetzen sehen wird, müssen einige Themen noch intensiver behandelt werden. Drei der wesentlichen Herausforderungen sind:

- *Ausführungsunterbrechungen*

In Mobilfunksystemen tritt häufig die Situation ein, dass kurz- oder längerfristig keine Kommunikation zwischen der Mobilstation und dem Netz möglich ist, da keine Funkverbindung hergestellt werden kann. Daher müssen Probleme, wie verloren gegangene Agenten, Unterstützung für wartende Agenten, sowie netzzustandsabhängige Agentenmigration detaillierter untersucht werden.

- *Sicherheit*

Ein mobiler Agent, der als persönlicher Assistent eines Benutzers agiert, benötigt zur effizienten Erfüllung seiner Aufgaben umfangreiches Wissen über die Interessen und privaten Daten seines Benutzers. Es ist daher unabdingbar, dass die in einem Agenten gespeicherte Information sicher vor der Manipulation durch Dritte geschützt wird. Wenn auch die Vermutung vorherrscht, dass man in erster Linie Netzknoten vor böswilligen Agenten schützen muss, so wurde in der vorliegenden Arbeit auch gezeigt, dass dieser Schutz leichter zu erreichen ist als der Schutz des Agentendatenspeichers vor böswilligen Agentensystemen.

- *Zukünftige Kommunikationsnetze*

In dieser Arbeit wurden mobile Agenten für GSM- und GPRS-Mobilfunknetze untersucht. Ein wesentlicher Vorteil von mobilen Agenten ist, dass diese für verschiedene Trägerdienste und Kommunikationsnetze eingesetzt werden können. Daher sollte die Übertragung von mobilen Agenten auch für Kommunikationsnetze wie UMTS und WLAN intensiv analysiert werden.

Insgesamt wird der Bedarf an personalisierten und portablen E-Commerce-Diensten für nomadische Benutzer weiter ansteigen und somit wird auch die Bedeutung der Agententechnologie in den kommenden Jahren weiter zunehmen, so dass mit einem erhöhten Forschungsbedarf in diesem Themenbereich zu rechnen ist.

Agentensystem IBM Aglets Workbench

Aglets Workbench (AWB) ist eine JAVA-Programmierungsumgebung für mobile Agentensysteme, die im IBM Forschungslabor in Tokio, Japan entwickelt wurde [IBM01]. Die erste Version wurde 1996 fertiggestellt, die Version Beta 2 wurde im Dezember 1999 herausgegeben. Mobile Agenten werden in AWB durch sogenannte Aglets¹ verkörpert. Dabei ist ein Aglet ein autonomes JAVA-Objekt, das durch das Netz übertragen wird und auf jedem Rechner vordefinierte Aufgaben durchführen kann [OK97]. Die Laufzeitinstanz von Agenten wird generell durch den Programmcode, die Laufzeitdaten und Ausführungszustände beschrieben. AWB überträgt bei einer Agentenmigration allerdings nur den Programmcode und Laufzeitdaten. Um die Ausführungslogik kontinuierlich zu halten, muss ein Agent vor der Migration einen Zugangspunkt (eine Callback-Funktion) definieren, von dem aus nach der Migration das Agentenprogramm weiter ausgeführt wird.

AWB besteht aus zwei Programmierschnittstellen (API²) [OK97]:

- Aglet API
- Agent Transport and Communication Interface (ATCI)

Das Aglet API ist die Schnittstelle zwischen AWB und Anwendungen. Diese Schicht verbirgt die Implementationsdetails aller unteren Schichten und ist von der direkt unter ihr liegenden Aglet-Runtime-Schicht realisiert. ATCI ist für die Übertragung der Aglets zuständig, und wird durch das proprietäre Agent Transport Protocol (ATP) unterstützt. ATP ist ein Protokoll der ISO/OSI-Schicht 7 und unabhängig vom Transportprotokoll (z.B. TCP/IP).

Abb. A.1 zeigt anhand eines Beispiels die Übertragung eines Aglets. Im Aglet API ist eine Funktion `dispatch(URL url)` für die Übertragung eines Aglets definiert, die von einem Aglet oder einer Anwendung aufgerufen werden kann. Der Parameter `url` ist die Adresse des Agentenservers, zu dem das Aglet migrieren soll. Nachdem diese Funktion aufgerufen wird, wird zuerst das Aglet in der Runtime-Schicht in ein Bytearray umgewandelt und dann wird das Agletprogramm angehalten. Der Bytearray wird zur ATCI-Schicht weitergeleitet und in der ATP-Schicht in einen Bitstream umgewandelt. Dieser Bitstream wird dann über das Transportprotokoll (TCP/IP) zum Zielagentenserver übertragen. Dort wird der gleiche Prozess umgekehrt durchgeführt und das Agletprogramm fortgesetzt.

AWB unterstützt folgende drei Typen von Nachrichten [OK97] (siehe auch Abb. 3.9):

- Synchron-Nachrichten
- Future-Nachrichten
- One-Way-Nachrichten

Die wichtigsten Komponenten in AWB sind *Aglet*, *AgletContext*, *Aglet-Proxy* und *Nachrichten-Objekte*. Der *AgletContext* repräsentiert die lokale Ausführungsumgebung, in der Aglets erzeugt, übertragen, empfangen und zerstört werden können. Weiterhin kann ein

¹Kunstwort aus *Agent* und *Applet*

²Application Programming Interface

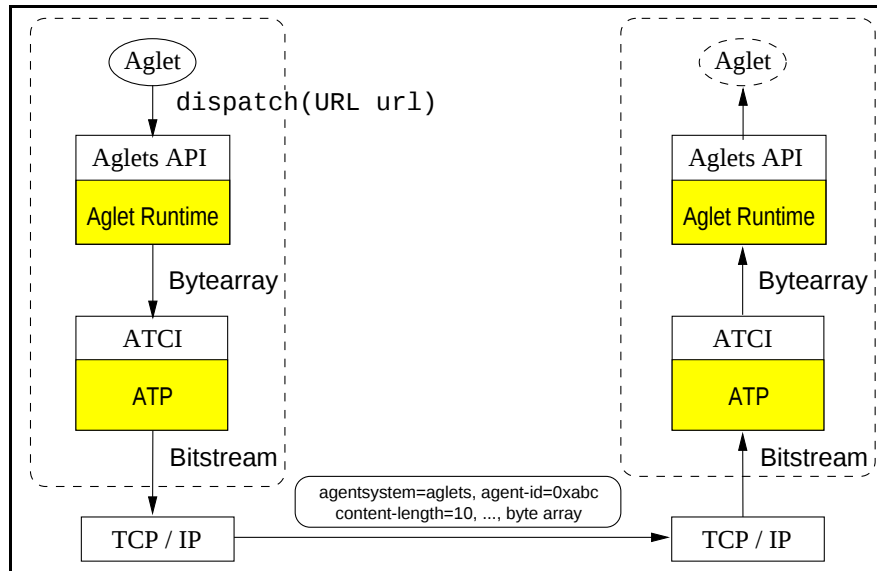


Abbildung A.1: AWB Systemarchitektur

Aglet beim *AgletContext* auch Informationen über das lokale Laufzeitsystem abfragen, siehe Abb. A.2.

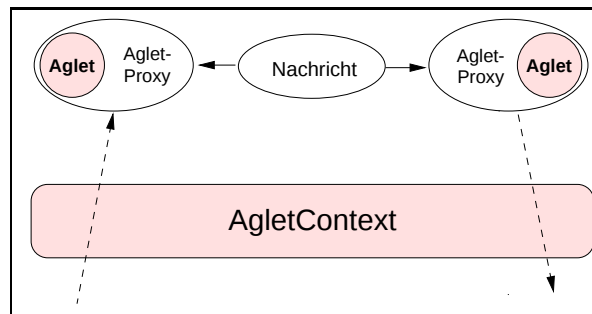


Abbildung A.2: AWB Objektmodell

Ein Aglet stellt anderen Objekten keine Referenz von sich zur Verfügung, statt dessen funktioniert die Kommunikation zwischen dem Aglet und der Außenwelt über einen *Aglet-Proxy*. Der *Aglet-Proxy* bietet einige allgemeine Funktionen, wie z.B. `dispatch()`, `clone()`, `dispose()`, etc., um das Aglet zu steuern. Benutzerdefinierte Funktionen müssen zuerst in ein Nachrichten-Objekt gepackt und dann über den *Aglet-Proxy* an das Aglet geschickt werden. In dem Aglet wird der Inhalt der Nachricht in einer Nachrichtenbearbeitungsfunktion verarbeitet. Verlässt ein Aglet eine Station, verlieren auch alle mit ihm verbundenen *Aglet-Proxies* ihre Gültigkeit.

Agentensystem Mitsubishi Concordia

Concordia [Mit01] ist eine Programmierumgebung für die Entwicklung und Verwaltung von mobilen Agentensystemen, welches im Rahmen dieser Arbeit in der Version 1.1.4 untersucht wurde. Es definiert eine Middleware-Schicht, die die Übertragung, Beständigkeit, Zusammenarbeit und Sicherheit von Agenten unterstützt [Mit98]. Innerhalb von Concordia ist ein Agent ein autonomes JAVA-Objekt, das durch das Kommunikationsnetz übertragen werden kann. Der Programmcode, die Daten und das Verhalten eines Agenten werden vom *Concordia-Server* verwaltet, d.h. nach der Erzeugung eines Agenten wird die Kontrolle an den Server übergeben. Alle Aktionen des Agenten werden nun vom Concordia-Server gesteuert, was auch bedeutet, dass ein Agent sich nicht selbst zu einem anderen Server schicken kann.

Der Concordia-Server bietet eine Kommunikationsinfrastruktur für die Agenten- und Datenübertragung. Concordia benutzt JAVA Object Serialization Facility (JOSF), ein Protokoll ähnlich wie JAVA RMI, um Agenten zu transportieren. Ein Concordia-Server wird durch den Rechnernamen identifiziert, daher kann auf jedem Rechner auch nur ein Concordia-Server gestartet werden. Beim Kommunikationsaufruf wird der Rechnernamen zunächst durch einen DNS-Server in die IP-Adresse des Concordia-Servers übersetzt.

Ein einfaches Concordia-Agentensystem besteht aus einem Concordia-Server und einem mobilen Agenten. Die folgenden weiteren Komponenten können in das System integriert werden:

- Administrations-Manager
- Sicherheits-Manager
- Beständigkeits-Manager
- Event-Manager
- Warteschlangen-Manager
- Dienstbrücke

Alle Komponenten sind in JAVA geschrieben. Wie Abb. B.1 zeigt, besteht ein Concordia Agentensystem aus einigen Serverobjekten, die auf einer oder mehreren JVMs laufen. Ein Agent wird durch den Aufruf einer Methode von einem Serverobjekt gestartet. Das System wird durch den Administrations-Manager konfiguriert und durch verschiedene Managerobjekte verwaltet. Eigene Dienste können mit Hilfe der Dienstbrücke in das System integriert werden.

Concordia unterstützt folgende Nachrichtentypen:

- Future-Nachrichten
- Multicast-Nachrichten
- Collaboration

Die ersten zwei Nachrichtentypen werden durch einen Eventmechanismus (Kap. ??) realisiert. Dabei werden die Events vom Event-Manager verwaltet. Ein Agent (selected event) oder eine Agentengruppe (group-oriented event) muss sich zunächst beim Event-Manager

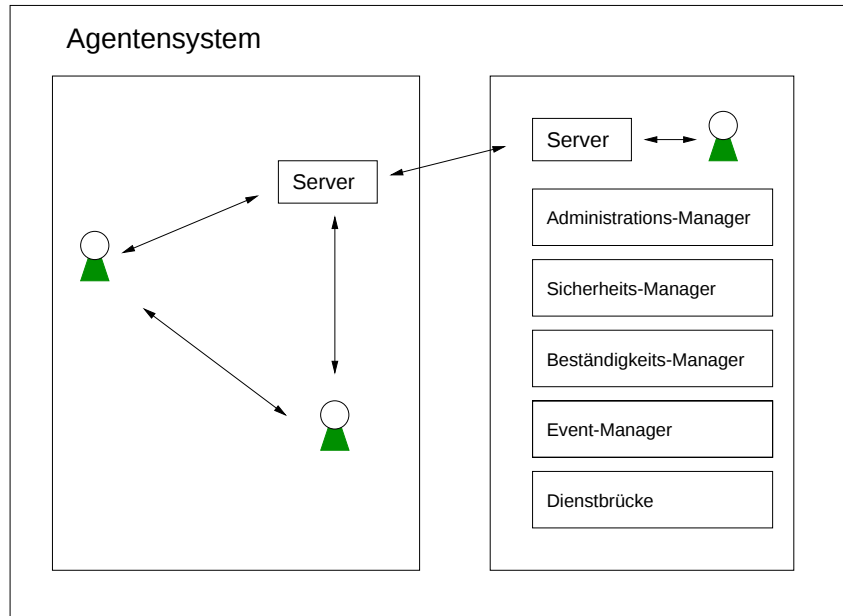


Abbildung B.1: Concordia Agentensystem

für bestimmte Eventtypen registrieren lassen, damit später Eventobjekte empfangen werden können. Wenn ein Programm einem Agenten oder einer Agentengruppe eine Nachricht schicken möchte, so generiert es zuerst ein Eventobjekt, das die Nachricht beinhaltet, und sendet es dann an den Event-Manager. Das Eventobjekt wird dann vom Event-Manager an den Empfänger oder die Empfängergruppe weitergeleitet.

Eine Collaboration-Nachricht wird für die Zusammenarbeit von Agenten eingesetzt, um komplizierte Probleme zu lösen [Mit98]. So wird eine Collaboration-Nachricht durch ein **AgentGroup** Objekt, in dem alle Agenten zusammenarbeiten werden, repräsentiert. Jeder Agent in der Gruppe erhält eine Referenz vom **AgentGroup** Objekt und kann seine Methoden via JAVA RMI aufrufen. In jedem Agenten wird eine Teilaufgabe definiert. Nachdem ein Agent seine Aufgabe erledigt hat, ruft er die Methode `collaborate()` in der **AgentGroup** auf, um seine Ergebnisse mitzuteilen. Dann wird die Ausführung dieses Agententhreads so lange blockiert, bis alle Agenten in der Gruppe diesen **Collaboration point** erreicht haben. Schließlich wird eine vordefinierte Callback Methode `analyzeResult` in jedem Agenten aufgerufen, um ihm die Ergebnisse der Gruppe zu übergeben.

HBCI-Dialog

HBCI arbeitet als synchrones, dialogorientiertes Protokoll, d.h. die Kommunikationspartner kommunizieren in einem Dialog, in dem der Sender die nächste Nachricht erst senden kann, wenn er eine Antwort vom Empfänger erhalten hat. Abb. C.1 zeigt den Ablauf eines HBCI-Dialogs.

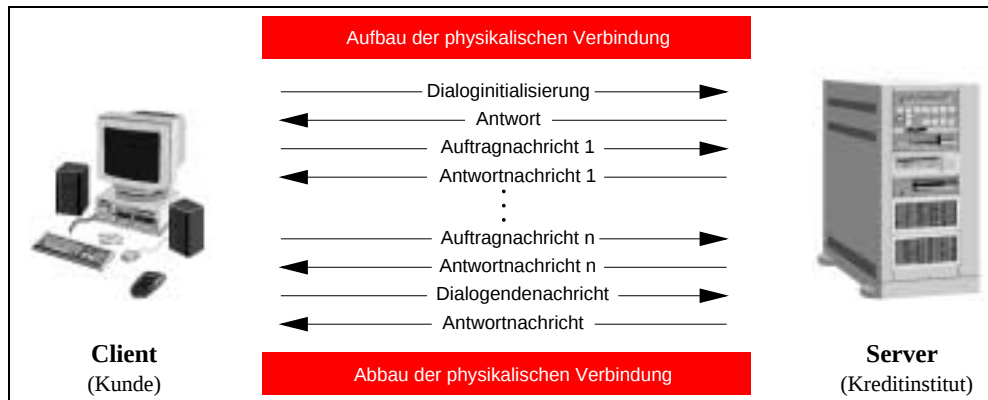


Abbildung C.1: Ablauf eines HBCI-Dialogs

Jeder HBCI-Dialog zwischen Client und Server beginnt mit einer Dialoginitialisierung, in der die Authentifizierungen der beiden Kommunikationspartner gegenseitig geprüft, die Verschlüsselungs- und Komprimierungsverfahren ausgehandelt und die Informationen über die Aktualität der Daten ausgetauscht werden. Nach erfolgreicher Initialisierung überträgt der Client Aufträge an den Server. Der Server bearbeitet diese Aufträge und sendet anschließend eine Antwort zurück an den Client. Wenn der Server alle Aufträge bearbeitet hat, wird der Dialog vom Client beendet.

Abb. C.2 zeigt die für die Kommunikation zwischen Client und Server vorgegebene Segmentabfolge innerhalb der HBCI-Nachrichten [Son98]. Jede Nachricht enthält dabei je ein Kopf- und ein Abschlussegment.

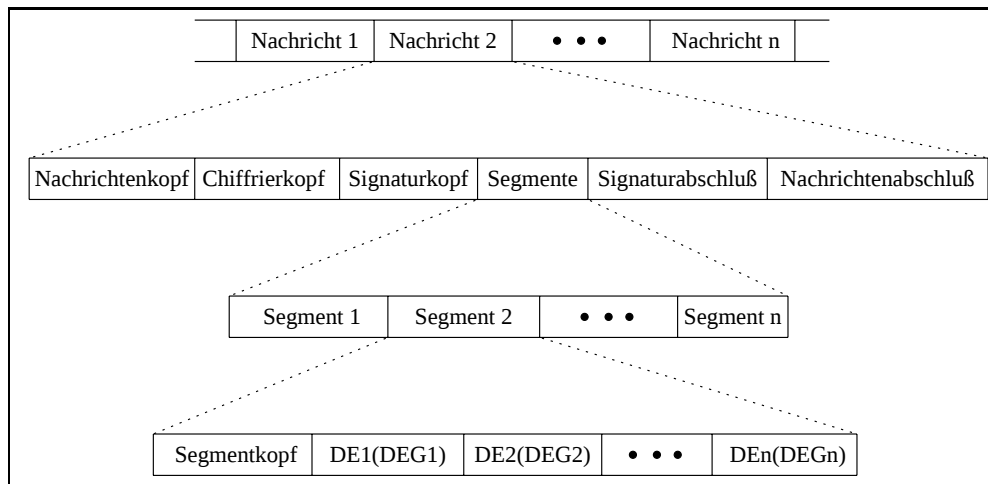


Abbildung C.2: Aufbau von HBCI-Nachrichten

Sicherheit von HBCI

Generell werden alle HBCI-Nachrichten bei der Übertragung verschlüsselt. Dabei werden sowohl symmetrische als auch asymmetrische Verfahren eingesetzt. Beim symmetrischen Verfahren verwenden die Kommunikationspartner den gleichen Schlüssel zum Ver- und Entschlüsseln. Es handelt sich dabei um nur den beiden involvierten Kommunikationspartnern bekannten Schlüssel (*secret keys*). Die häufigsten verwendeten symmetrischen Verfahren sind der *Data Encryption Standard* (DES) und der *Message Authentication Code* (MAC). DES ist ein Verschlüsselungsverfahren, das 1977 von der US-Regierung standardisiert wurde. Es verschlüsselt 64-bit Datenblöcke in Datenblöcke gleicher Länge mittels eines Schlüssels. MAC dagegen ist eine Art Prüfsumme einer Nachricht, die zur Authentifizierung herangezogen wird.

Ein asymmetrisches Verfahren verwendet jeweils Schlüsselpaare, die immer aus einem privaten Schlüssel (*private key*) und einem öffentlichen Schlüssel (*public key*) bestehen. Das zugrundeliegende Prinzip besteht darin, dass ein Kunde per Software ein persönliches Schlüsselpaar erzeugt und seine Aufträge mit seinem privaten Schlüssel signiert. Das Kreditinstitut kann mittels öffentlicher Schlüssel die elektronische Unterschrift auf Korrektheit prüfen. Der öffentliche Schlüssel beweist einerseits die Herkunft der Signatur eindeutig, muss aber andererseits nicht geheimgehalten werden, da mit ihm nur Signaturen überprüft, jedoch nicht erzeugt werden können. Entsprechend kann ein Kunde vertrauliche Daten mit dem öffentlichen Schlüssel des Kreditinstitutes verschlüsseln. Das bekannteste asymmetrische Verfahren ist Rivest-Shamir-Adleman (RSA), das nach den Namen seiner Erfinder benannt ist und sowohl zum Verschlüsseln als auch zum Authentifizieren verwendet werden kann. HBCI verwendet zwei Schlüsselpaare: Signierschlüssel und Chiffrierschlüssel.

Die HBCI-Spezifikation erlaubt dabei zwei Verfahren für die Verschlüsselung:

- das DES-DES-Verfahren (DDV) und
- das RSA-DES-Hybridverfahren (RDH)

Der Nachrichteninhalt wird in beiden Verfahren mittels einer Zufallszahl als Nachrichtenschlüssel durch das DES-Verfahren verschlüsselt. DDV verwendet den MAC als Signatur und verschlüsselt den Nachrichten-Chiffrierschlüssel mittels DES. In der HBCI Version 2.01 wird DDV beim Einsatz einer ZKA-Chipkarte verwendet.

RDH verwendet RSA zum Signieren der Nachrichten und zum Chiffrieren der Chiffrierschlüssel. In der HBCI-Version 2.01 wird eine Softwarelösung mit Diskette oder Festplatte als Speichermedium durch RSA-Verfahren unterstützt. Langfristig soll RDH als einheitliche Lösung eingesetzt werden. Der Unterschied zwischen DDV und RDH liegt hauptsächlich in den verschiedenen kryptographischen Verfahren mit unterschiedlichen Schlüsseln (symmetrisch und asymmetrisch). Die Verarbeitungsschritte sind identisch. Im Folgenden wird daher lediglich RDH beschrieben. Abb. D.1 zeigt die Verteilung der Schlüssel in RDH. Die beiden Kommunikationspartner (Kunde und Bank) benötigen insgesamt vier Schlüsselpaare (je zwei für eine Richtung). Gemäß des Standards werden für die Signatur

und Chiffrierung getrennte Schlüsselpaare eingesetzt.

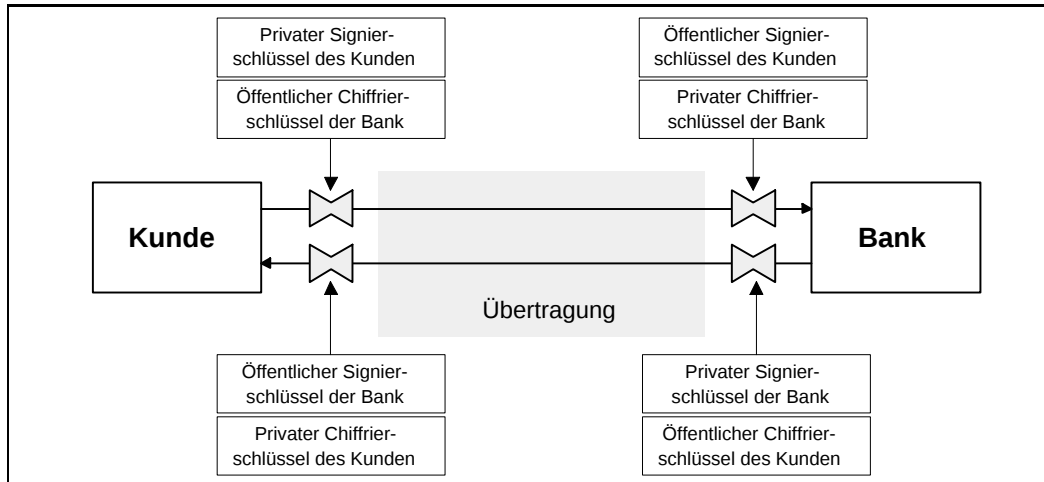


Abbildung D.1: Verteilung der Sicherheitsschlüssel beim HBCI-Protokoll

Um HBCI einzusetzen, müssen zunächst die öffentlichen Schlüssel und anderer vertraulichen Daten (Bank- und Kundenparameterdaten) ausgetauscht werden. Dazu gibt es zwei mögliche Wege: die Zusendung der Schlüssel auf Sicherheitsmedien (z.B. Diskette, Chipkarte) oder die Übertragung beim Erstzugang. Für den zweiten Weg müssen zusätzlich noch Initialisierungsbriefe von der Bank zum Kunden und umgekehrt geschickt werden, in denen die Hashwerte der Nachrichten mitgeteilt werden. Diese Briefe dienen zur Überprüfung, ob die Daten tatsächlich vom Partner stammen und die Übertragung korrekt ist. Beim DDV ist der zweite Weg ausgeschlossen, da die geheimen Schlüssel nicht übertragen werden dürfen.

ABBILDUNGSVERZEICHNIS

2.1	Telekommunikationsdienste	12
2.2	Die logische GSM-Architektur	16
2.3	Die logische CAMEL-Architektur	19
2.4	Agentenbasierte VHE-Architektur	21
2.5	Funktion des WAP-Gateways	22
2.6	Internet- und WAP-Architektur	23
2.7	Ressourcennutzung von GSM CSD und GPRS	25
2.8	Logische GPRS-Architektur	26
2.9	GPRS-Protokollstapel	28
3.1	Klassifikation von Softwareagenten	36
3.2	Mobile Agenten im Kommunikationsnetz	37
3.3	Systemarchitektur	38
3.4	Agententypen	39
3.5	Lebenszyklus eines mobilen Agenten	40
3.6	Sicherheitsaspekte bei Agentensystemen	41
3.7	Kommunikationsmechanismen verteilter Systeme	44
3.8	Vergleich RPC - Mobile Agenten	45
3.9	Kommunikation mit Hilfe von Nachrichten	46
3.10	Kommunikation durch Events	47
3.11	Erzeugen eines entfernten Objektes in Voyager	53
3.12	Space-Architektur	54
3.13	Grasshopper Agentensystem	55
4.1	Grenzen für die Zusammenarbeit zwischen Agentensystemen	59
4.2	Schematische Darstellung der Funktion eines CORBA ORBs	61
4.3	Zugriff auf die Schnittstellen eines MASIF-Systems	64
4.4	Netzknoten bei der agentenbasierten Dienstleistung	66
4.5	Verschiedene Endgeräte nutzen die gleichen Dienste	68
4.6	Einbinden des OrbixWeb Naming Service in Grasshopper	71
4.7	Grasshopper-Region mit MASIF-Schnittstellen	73
4.8	Voyager-Region mit MASIF-Port	73
4.9	Startprozess des MASIF-Ports	75
5.1	Agentensystemtypen	78
5.2	Stufen der Mobilität	79
5.3	Funktion des MCAPs	83
5.4	Architektur des MCAPs	84
5.5	Aufbau des MCAPs	85
5.6	HBCI-System	88
5.7	Klassische HBCI-Anwendung	89
5.8	Konzeption des AHB-Dienstes	90
5.9	Realisierung des AHB-Dienstes	92
5.10	Graphische Oberfläche des TAs	92
5.11	Graphische Oberfläche des HTAs	92

5.12	Ablauf des AHB-Dienstes	94
5.13	Nachrichtenfluss beim Start des Dienstes	95
5.14	Nachrichtenfluss bei der Auftragsbearbeitung	96
5.15	Heutiger M-Commerce-Dienst	97
5.16	ASB-Dienst mittels MCAP	97
5.17	Informationsfluss beim Einsatz von MAs (oben) und SQL (unten)	99
5.18	Mobiler Suchagent für den ASB-Dienst	100
5.19	Mobiler Suchagent mit XML-Schnittstelle	101
5.20	Stationärer Suchagent	102
6.1	Kommunikationszeiten	106
6.2	Warteschlange	107
6.3	Messungen mit GATE	112
6.4	CSD-Datenraten	113
6.5	Verbindungsauf- und abbauverzögerungen bei CSD	114
6.6	WAP-Kommunikation	117
6.7	WAP-Gateway als Inhaltskodierer/-dekodierer	118
6.8	IP-Datenpakete	119
6.9	Antwortzeit des WAP-Gateways in Abhängigkeit der WSP-Nutzdaten	120
6.10	Verkehrsmodell für das Aufsuchen von Händlern	124
6.11	Antwortzeiten beim ASB-Dienst	127
6.12	Klassische HBCI-Anwendung	128
6.13	Definition der Antwortzeiten beim AHB-Dienst	129
6.14	Vergleich der Datenvolumina B_{ahb} der Bankanwendungen	130
6.15	Dienstausführungszeit des AHB-Dienstes über GPRS	135
6.16	Verkehrsmodell für den mobilen Suchagenten	136
6.17	Kodierungsrate des WAP-Gateways für WML-Dokumente	138
6.18	WAP-Antwortzeiten für CSD	139
6.19	TCP-Datenverkehr für die statischen Suchagenten	140
6.20	Datenvolumen bei Suchmethode ERSTES	141
6.21	Datenvolumen bei Suchmethode DREI	141
6.22	Datenvolumen bei Suchmethode BESTES	141
6.23	Selektivität γ eines mobilen Suchagenten in Abhängigkeit von der Anzahl der besuchten Händler	143
6.24	Antwortzeit T_{ps} (SSA)	144
6.25	Antwortzeit T_{ps} (MSA)	144
6.26	Suchmethode ERSTES	145
6.27	Suchmethode DREI	145
6.28	Suchmethode BESTES	145
6.29	Suchmethode SMART	145
6.30	Verhältnis Ankunftsrate des MCAPs zur Händleranzahl	146
6.31	Mittlere Antwortzeit T_{ps} beim Dienstmix	147
6.32	Mittlere Dienstausführungszeit S_{asb} in Abhängigkeit vom Verkehrswert	147
6.33	Mittlere Dienstausführungszeit S_{asb} in Abhängigkeit von der Händlerzahl	148
6.34	Mittlere Dienstausführungszeit S_{asb} bei GPRS	149
A.1	AWB Systemarchitektur	156
A.2	AWB Objektmodel	156
B.1	Concordia Agentensystem	158

C.1	Ablauf eines HBCI-Dialogs	159
C.2	Aufbau von HBCI-Nachrichten	160
D.1	Verteilung der Sicherheitsschlüssel beim HBCI-Protokoll	162

TABELLENVERZEICHNIS

2.1	Historische Entwicklung von Intelligenten Netzen	14
2.2	Systemcharakteristika GSM 900	16
2.3	Kanalkodierungsschemata in GPRS	28
2.4	Prioritätenklassen	29
2.5	Verzögerungsklassen	30
2.6	Maximaler Durchsatz	31
2.7	Durchschnittlicher Durchsatz	31
6.1	Verteilungsfunktionen der Ereignisabstände	108
6.2	Abfertigungsstrategien von Warteschlangen	108
6.3	Transaktionsleistung bei CSD	115
6.4	Antwortzeit T_{cs} bei CSD	115
6.5	GPRS-Verkehrsmodelle	116
6.6	Bediendauern und Variationskoeff. des MCAPs und der Händlerrechner	120
6.7	Parameter bei der Agentenmigration	123
6.8	Datenvolumen des HTAs	131
6.9	Vergleich der Antwortzeiten der Bankanwendungen	132
6.10	Übertragungszeit des HTAs über CSD	132
6.11	Gebühreneinheiten bei der Bankanwendung	133
6.12	WAP-Nutzdaten der Transportschicht (Uplink)	137
6.13	WAP-Nutzdaten der Transportschicht (Downlink)	137
6.14	WAP-Nutzdaten der Anwendungsschicht (Downlink)	137
6.15	Untersuchte Trefferwahrscheinlichkeiten beim ASB-Dienst	140
6.16	MSA-Agentenparameter	142
6.17	Größe der Suchergebnisse	142
6.18	Größe der Händlerliste B_{mlist}	142
6.19	Antwortzeiten beim ASB-Dienst	143
6.20	Dienstmix	146
6.21	Simulierte mittlere Verweilzeit V_{gprs} bei der WAP-Kommunikation	148
6.22	Vergleich der mittleren Antwortzeiten T_{wap} bei CSD und GPRS	149

ABKÜRZUNGSVERZEICHNIS

A	Schnittstelle zwischen MSC und BSC (GSM)	BMF	Bundesministerium für Bildung und Forschung
A_{bis}	Schnittstelle zwischen BSC und BTS (GSM)	BP	Bearbeitungsprozess (AHB)
ACL	Agent Communication Language (FIPA)	BPD	Bankparameterdaten (HBCI)
ACTS	Advanced Communication Technologies and Services (EU)	BSC	Base Station Controller (GSM)
ADA	Allgemeiner Dienstagent	BSS	Base Station Subsystem (GSM)
ADT	Abstrakter Datentyp	BTS	Base Transceiver Station (GSM)
AHB	Agentenbasierter Homebanking Dienst	BTX	Bildschirmtext
AHB+	AHB-Dienst mit SSL-Sicherung	C	Schnittstelle zwischen HLR und GMSC (GSM)
AI	Artificial Intelligence	CA	Communicative Act <i>oder</i> Certification Authority
AIN	Advanced Intelligent Network	CAMEL	Customized Applications for Mobile network Enhanced Logic (ETSI)
AMS	Agent Management System	CAP	CAMEL Application Protocol
ANSI	American National Standards Institute	CCIR	Comité Consultatif International de Radio, International Radio Consultative Committee, seit 3/93: ITU-R
API	Application Programming Interface	CCITT	Comité Consultatif International de Télégraphique et Téléphonique, International Telegraph and Telephone Consultative Committee, seit 3/93: ITU-T
APM	Adaptive Profile Manager (CAMELEON)	CEPT	Conférence Européenne des Posts et Télécommunications, Conference of European Posts and Telecommunications
ASB	Agentenbasierter Such- und Bestelldienst	CGI	Common Gateway Interface <i>oder</i> Cell Global Identity (LAI+CI)
ASP	Application Service Provider	CI	Cell Identity (GSM)
AuC	Authentication Centre (GSM)	CLIP	Calling Line Identification Presentation (ISDN, GSM)
AWB	Aglets Workbench (IBM)	COM	Common Object Model
AXE10	Digitale Vermittlungsstelle der Firma Ericsson	CORBA	Common Object Request Broker Architecture (OMG)
B	Schnittstelle zwischen VLR und zugeordnetem MSC (GSM)	COS	Common Object Services (CORBA)
B2B	Business-to-Business	CPU	Central Processing Unit, Zentralprozessor
B2C	Business-to-Consumer	CRM	Customer Relationship Management
BA	Benutzeragent		
BDB	Bundesverband Deutscher Banken		
BDI	Beliefs, Desires and Intentions		
BER	Bit Error Rate, Bitfehlerrate		
BHCA	Busy Hour Call Attempts		

CS-x	IN Capability Set (ITU)	F	Schnittstelle zwischen MSC und EIR (GSM)
CSD	Circuit-Switched Data (GSM)		
CSE	CAMEL Service Environment	FCFS	First Come, First Serve
CSI	CAMEL Subscription Information	FDMA	Frequency Division Multiple Access
CTM	Cordless Terminal Mobility (ETSI)	FEC	Forward Error Correction
D	Deterministic, die involvierten Zeiten sind konstant <i>oder</i> Schnittstelle zwischen HLR und VLR (GSM)	FFS	Flexible Financial Service (CAMELEON)
		FIPA	Foundation for Intelligent Physical Agents
DA	Dienstagent	G	Generelle Verteilung <i>oder</i> Schnittstelle zwische zwei VLRs (GSM)
DAE	Distributed Agent Environment		
DAP	Dienstanbieterprozess (AHB)	GIOP	General Inter-ORB Protocol (CORBA)
DDV	DES-DES-Verfahren (HBCI)		
DES	Data Encryption Standard	GGSN	Gateway GPRS Support Node (GPRS)
DECT	Digital Enhanced Cordess Telecommunication (ETSI)	GMSC	Gateway MSC (GSM)
DF	Directory Facilitator	GMSK	Gaussian Minimum Shift Keying
DOC	Distributed Object Computing	GPRS	General Packet Radio Service (GSM)
DOT	Distributed Object Technology	GSM	Global System for Mobile communications (früher: Groupe Special Mobile)
DCOM	Distributed Common Object Model		
DES	Data Encryption Service	GUI	Graphical User Interface, Benutzerschnittstelle
DII	Dynamic Invocation Interface		
DOT	Distributed Object Technology	GUID	Global Unique Identifier (FIPA)
DSI	Dynamic Skeleton Interface	H	Hyperexponentiale Verteilung
DSS1	Digital Subscriber Signalling System No. 1	HBCI	Home-Banking Computer Interface
		HDLC	High Level Data Link Control (ISO)
DTD	Document Type Definition (XML)	HLR	Home Location Register (GSM)
DTMF	Dual Tone Multiple Frequency	HSCSD	High Speed Circuit Switched Data (GSM)
E	Erlang <i>oder</i> Schnittstelle zwischen zwei MSCs (GSM)	HTA	Homebanking Terminal Agent
ECBS	European Committee for Banking Standards	HTML	Hypertext Markup Language
		HTTP	Hypertext Transmission Protocol
EDI	Electronic Data Interchange	IA	Intelligenter Agent
EED	Ericsson Eurolab Deutschland GmbH	IDL	Interface Definition Language (CORBA)
EIR	Equipment Identity Register (GSM)	IEEE	Institute of Electrical and Electronics Engineers
ESIOP	Environment Specific Inter ORB Protocol (CORBA)	IIOP	Internet Inter-ORB Protocol (CORBA)
ETSI	European Telecommunications Standards Institute		
EU	Europäische Union	IN	Intelligent Network (ITU), Intelligentes Netz

INAP	IN Application Protocol	MCAP	M-Commerce Agentenplattform
IOR	Interoperable Object Reference	MExE	Mobile station application Execution Environment (GSM)
IP	Internet Protocol	MEZ	Mittel-Europäische Zeit
ISDN	Integrated Services Digital Network (ITU)	MS	Mobilstation (GSM)
ISO	International Standardization Organization	MSA	Mobiler Suchagent
ISP	Internet Service Provider	MSA+	MSA mit XML-Schnittstelle
IST	Information Society Technologies (EU)	MSC	Mobile service Switching Centre
ISUP	ISDN User Part (ITU)	MSISDN	Mobile Subscriber ISDN Number (GSM)
ITU	International Telecommunication Union	MSRN	Mobile Subscriber Roaming Number (GSM)
ITU-R	Funkbereich der ITU, früher CCIR	MTU	Maximum Transmission Unit <i>oder</i> Message Transfer Unit
ITU-T	Telekommunikationsbereich der ITU, früher CCITT	NCS	Network Computing System
JDK	JAVA Development Kit	NDB	Network Database
JOSF	JAVA Object Serialization Facility	NFS	Network File System
JVM	JAVA Virtual Machine	NSS	Network and Switching Subsystem (GSM)
KI	Künstliche Intelligenz	OMG	Object Management Group
KIF	Knowledge Interchange Format	OO	Object-Orientierung
KQML	Knowledge Query and Manipulation Language	ORB	Object Request Broker
LAI	Location Area Identifier (GSM)	OSI	Open System Interconnection
LAN	Local Area Network	PC	Personal Computer
LAPD	Link Access Protocol on the D-channel (ISDN)	PDA	Personal Digital Assistant
LAPD _m	LDAP über die <i>A_{bis}</i> -Schnittstelle (GSM)	PDCH	Packet Data Channel (GPRS)
LCFS	Last Come First Serve	PDE	Protokolldateneinheit
LDAP	Lightweight Directory Access Protocol	PDN	Packet Data Network
LLC	Logical Link Control	PDP	Packet Data Protocol (GPRS)
MA	Mobiler Agent	PDTCH	Packet Data Traffic Channel (GPRS)
MAC	Medium Access Control <i>oder</i> Message Authentication Code	PDU	Protocol Data Unit
MAF	Mobile Agent Framework (OMG)	PHP	PHP Hypertext Preprocessor (früher: Personal Home Page)
MAP	Mobile Application Protocol (GSM)	PIN	Persönliche Identifikationsnummer
MAS	Multi-Agenten-System	PLMN	Public Land Mobile Network, Öffentliches Mobilfunknetz
MASIF	Mobile Agent System Interoperability Facility (OMG)	PLR	Packet Loss Rate, Paketverlustrate
MAT	Mobile Agent Technology	PNO	Public Network Operator
		PPP	Point-to-Point Protocol

PSTN	Public Switched Telecommunication Network	TETRA	TErrestrial TRunked Radio, früher: Trans European Trunked Radio
QoS	Quality of Service	TINA	Telecommunications Information Network Architecture
RDH	RSA-DES Hybridverfahren	TLS	Transport Layer Security
RFC	Request for Comments	TMN	Telecommunications Management Network
RFI	Request for Information	TS	Time Slot
RFP	Request for Proposals	UA	User Agent, Benutzeragent
RLC	Radio Link Control	UDP	User Datagram Protocol
RLP	Remote Link Protocol (GSM)	UML	Unified Modelling Language
RMI	Remote Method Invocation	UMTS	Universal Mobile Telecommunications System
RPC	Remote Procedure Call	UPD	Kundenparameterdaten (HBCI)
RR	Round Robin	URI	Uniform Resource Identifier
RSA	Rivest-Shamir-Adleman Verfahren	URL	Uniform Resource Locator
RTT	Round-Trip Time, Umlaufzeit	USIM	Universal Subscriber Identity Module
RWTH	Rheinisch-Westfälische Technische Hochschule	VAB	Virtual Address Book (CAMELEON)
SAT	SIM Application Toolkit (GSM)	VAS	Value Added Service, Mehrwertdienst
SCF	Service Control Function (IN)	VHE	Virtual Home Environment (UMTS)
SCP	Service Control Point (IN)	VKI	Verteilte Künstliche Intelligenz
SDA	Spezielle Dienstagenten	VLR	Visitor Location Register (GSM)
SDF	Service Data Function (IN)	VS	Verteiltes System
SDP	Service Data Point (IN)	VMSC	Visited MSC (CAMEL)
SET	Secure Electronic Transaction	VPN	Virtual Private Network (IN)
SGML	Standard Generalised Markup Language (W3C)	VRMP	Voyager Remote Messenger Protocol
SGSN	Serving GPRS Support Node (GPRS)	VSSF	Visited Service Switching Function (CAMEL)
SIB	Service Independent Building block (IN)	W3C	World Wide Web Consortium
SIM	Subscriber Identity Module (GSM)	WAE	Wireless Application Environment (WAP)
SIRO	Service In Random Order	WAP	Wireless Application Protocol
SMG	Special Mobile Group (ETSI)	WBXML	WAP Binary XML (WAP)
SMS	Short Message Service (GSM)	WDP	Wireless Datagram Protocol (WAP)
SNDCP	Subnetwork Dependent Convergence Protocol (GPRS)	WSP	Wireless Session Protocol (WAP)
SQL	Structured Query Language	WTLS	Wireless Transport Layer Security (WAP)
SSA	Stationärer Suchagent	WTP	Wireless Transaction Protocol (WAP)
SSF	Service Switching Function (IN)	XML	eXtensible Markup Language
SSL	Secure Socket Layer	ZKA	Zentraler Kreditausschuss
TA	Terminal Agent, Endgeräteagent		
TAN	Transaktionsnummer		
TBF	Temporary Block Flow (GPRS)		
TCP	Transmission Control Protocol		
TDMA	Time Division Multiple Access		

FORMELZEICHEN

α	Agent
β	mittlere Bediendauer eines Auftrags, [s]
β_m	mittlere Bediendauer eines Händlerrechners (ASB), [s]
β_{mcap}	mittlere Bedienrate des MCAPs (ASB), [s]
β_{wapgw}	mittlere Bediendauer des WAP-Gateways, [s]
γ	Selektivität eines Agenten
Δ_{wap}	Verzögerung durch WAP/Web-Server, [s]
ϵ	Dienstrate, [1/s]
η	Steigung der WAP-Bedienzeitgleichung, [s/byte]
λ	mittlere Ankunftsrate, [1/s]
λ_0	mittlere Dienstnutzung, [1/s]
λ_{bh}	Benutzeranfragen in der Hauptverkehrsstunde, [1/s]
λ_{max}	maximale Ankunftsrate, [1/s]
μ	mittlere Bedienrate eines Prozessors, [1/s]
ρ	Auslastung, Verkehrslast
ρ_{mcap}	Verkehrslast des MCAPs
θ	Schnittpunkt der WAP-Bedienzeitgleichung mit der Ordinate, [s]
σ	Standardabweichung
σ_b	Standardabweichung der Bediendauer β
ω	Ort eines Agentensystems
Ω	Region einer Agentenwelt
Ω_{core}	Kernregion einer Agentenwelt
ξ	Kodierungsrate des WAP-Gateways
A_{bh}	Verkehrswert, [BHCA]
B	Datenvolumen, [byte]
B_{ahb}	Mittleres Datenvolumen eines DAs beim AHB-Dienst, [byte]
B_{back}	Größe des SDAs bei der Migration zum Endgerät, [byte]
B_{code}	Größe der Klassendatei, [byte]
B_{cr}	Größe der Anfrage nach der Klassendatei, [byte]
B_{data}	Größe des Datenspeichers eines Agenten, [byte]
B_{filter}	Gefilterte Datenmenge, [byte]
B_{hbci}	Übertragene Datenmenge bei der klassischen HBCI-Lösung, [byte]
$B_{hbci,req}$	Datenmenge der HBCI-Nachrichten vom Client, [byte]
$B_{hbci,res}$	Datenmenge der HBCI-Nachrichten zum Client, [byte]
B_{max}	Maximale Segmentgröße eines TCP/IP-Paketes, [byte]
B_{mig}	Gesamtgröße eines Agenten bei der Migration, [byte]
B_{mlist}	Größe einer Händlerliste, [byte]
B_{res}	Größe eines Suchergebnisses (ASB), [byte]
B_{state}	Größe der serialisierten Zustandsinformationen eines Agenten, [byte]
B_{sub}	Größe des SDAs bei der Migration zum MCAP, [byte]
B_{sum}	Gesamtdatenmenge, [byte]
B_{tcp}	TCP-Nutzdaten, [byte]

B_{udp}	UDP-Nutzdaten, [byte]
B_{wbxml}	WSP-Nutzdaten, [byte]
B_{wml}	HTTP-Nutzdaten, [byte]
C	Kommunikationskosten
K_{pri}	Preis pro Gebühreneinheit
K_{tcp}	Konstante, die von der TCP-Implementierung abhängt
L	mittlere Anzahl von Aufträgen im System
$M(\alpha)$	Agentenmigration als Vektor der besuchten Regionen
N	mittlere Warteschlangenlänge
P	Wahrscheinlichkeit
P_{code}	Wahrscheinlichkeit für die Übertragung von Programmcode
P_e	Trefferwahrscheinlichkeit
P_{plr}	Wahrscheinlichkeit für einen Paketverlust
S	mittlere Dienstaussführungszeit, [s]
T	mittlere Verzögerung, [s]
T_{back}	mittlere Übertragungszeit eines SDAs zum Endgerät, [s]
T_{code}	mittlere Übertragungszeit des ADAs zum Endgerät, [s]
T_{cs}	mittlere Zeit für den Auf- und Abbau einer Datenverbindung, [s]
$T_{cs,tcp}$	mittlere Zeit für den Auf- und Abbau einer TCP/IP-Datenverbindung, [s]
$T_{cs,wap}$	mittlere Zeit für den Auf- und Abbau einer WAP-Datenverbindung, [s]
T_{hbc}	mittlere Antwortzeit bei der klassischen HBCI-Lösung, [s]
T_{mig}	mittlere Migrationsdauer (hin und zurück) eines Agenten, [s]
T_{ps}	mittlere Produktsuchzeit, [s]
T_{rtt}	mittlere Umlaufzeit für ein TCP/IP-Paket, [s]
T_{set}	mittlere Zeit, die für die Durchführung der Bezahlung benötigt wird, [s]
T_{sub}	mittlere Übertragungszeit eines SDAs zum MCAP, [s]
T_{takt}	Gebührentaktung, [s]
$T_{user,login}$	mittlere Zeit, die der Benutzer zur Authentifizierung benötigt, [s]
$T_{user,query}$	mittlere Zeit, die der Benutzer zur Spezifikation seiner Suche benötigt, [s]
T_{wap}	mittlere WAP-Kommunikationsdauer zwischen Endgerät und MCAP, [s]
U_{sum}	Summe Gebühreneinheiten
V	mittlere Verweilzeit, [s]
V_{csd}	mittlere Verweilzeit auf einer CSD-Mobilfunkstrecke, [s]
V_{mcap}	mittlere Verweilzeit der Agenten beim MCAP, [s]
V_{place}	mittlere Verweilzeit eines Agenten bei einem Ort des Agentensystems, [s]
V_{wapgw}	mittlere Verweilzeit für WAP-Nachrichten beim WAP-Gateway, [s]
W	mittlere Wartezeit eines Auftrags in der Warteschlange, [s]
b	Übertragungsrate, Datenrate, Durchsatz, [bit/s]
b_{csd}	Transaktionsleistung für Nutzdaten der Transportschicht über CSD, [byte/s]
$b_{csd,tcp}$	Transaktionsleistung für TCP-Nutzdaten bei der Agentenmigration, [byte/s]
$b_{csd,wap}$	Transaktionsleistung für UDP-Nutzdaten von WAP-Nachrichten, [byte/s]
b_{gprs}	GPRS-Datenrate, [byte/s]
$b_{internet}$	Internet-Datenrate, [byte/s]
c_b	Variationskoeffizient der Bediendauer β
c_{ps}	Variationskoeffizient der Produktsuchzeit T_{ps}
m	Mobilitätsfunktion eines Agenten
n	Anzahl der Bediener im System
n_m	Anzahl Händler

n_{max}	Maximale Anzahl von Teilnehmern
n_{user}	Anzahl Teilnehmer im System
n_{vm}	Anzahl besuchter Händler
r_{bh}	Anfragen pro Teilnehmer in der Hauptverkehrsstunde
r_{user}	Anfragen pro Teilnehmer
t	Zeit, [s]
t_{back}	Übertragungszeit eines SDAs zum Endgerät, [s]
t_{cs}	Zeit für den Auf- und Abbau einer Datenverbindung, [s]
t_{hbc}	Antwortzeit bei der klassischen HBCI-Lösung, [s]
t_{mig}	Migrationsdauer (hin und zurück) eines Agenten, [s]
t_{proc}	Bearbeitungszeit von HBCI-Nachrichten, [s]
t_{ps}	Antwortzeit bei der Produktsuche, [s]
t_{sub}	Übertragungszeit eines SDAs zum MCAP, [s]
t_{trans}	Übertragungszeit von HBCI-Nachrichten, [s]
t_{wap}	Antwortzeit für den Abruf von WAP-Dateien, [s]

DANKSAGUNG



LEBENS LAUF

Name : Jens Hartmann

Anschrift : Hermannstr. 10, 41460 Neuss

Geburtstag und -ort : 6. Oktober 1970 in Düsseldorf

Familienstand : Ledig

Staatsangehörigkeit : Deutsch

Ausbildung

08/1977 - 07/1981 : Theodor-Fliedner-Schule, städt. Grundschule, Neuss

09/1981 - 05/1990 : Theodor-Schwann-Gymnasium, städt. Gymnasium, Neuss

10/1990 - 09/1991 : Grundwehrdienst

10/1991 - 04/1997 : Rheinisch-Westfälische Technische Hochschule Aachen
Studium der Elektrotechnik

10/1995 - 02/1996 : Norwegische Technische Hochschule, Trondheim
ERASMUS-Stipendiat

Beruf

seit 05/1997 : Ericsson Eurolab Deutschland GmbH, Herzogenrath
Forschungsabteilung

03/1998 - 02/2000 : Technischer Leiter des ACTS-Projektes CAMELEON

seit 03/2000 : Projektleiter des IST-Projektes ParcelCall

Freizeit

seit 09/1999 : Mitglied des Rates der Stadt Neuss
Vorsitzender des städt. Bauausschusses

LITERATURVERZEICHNIS

- [3GP00] 3GPP (3rd Generation Partnership Project). *The Virtual Home Environment (VHE)*. Technical Specification, Group Services and System Aspects, TS22.121, Version 3.3.0. Juni 2000.
- [APS99] Mark Allman, Vern Paxson, Richard Stevens. *RFC 2581: TCP Congestion Control*. <http://www.ietf.org/rfc/rfc2581.txt>, April 1999.
- [Bau97] Stephan Baucke. *Leistungsbewertung und Optimierung von TCP für den Einsatz im Mobilfunknetz GSM*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Informatik IV, 1997.
- [BdB00] BdB (Bundesverband deutscher Banken). *HBCI Homebanking Computer-Interface, Schnittstellenspezifikation Version 2.2*, Mai 2000.
- [BER01] Peter Braun, Christian Erfurth, Wilhelm Rossak. Performance evaluation of various migration strategies for mobile agents. In *GI/ITG-Konferenz Kommunikation in Verteilten Systemen (KiVS)*, Hamburg, 2001.
- [BFGH00] Helmut Becker, Wolfgang Froberg, Carmelita Görg, Jens Hartmann. Virtual Home Environment. In *FMC - Konvergenz von Fest- und Mobilfunknetzen*, Seite 95–102. VDE Verlag, Frankfurt, ISBN 3-8007 2536-3, 2000.
- [BGM⁺99] Brian Brewington, Robert Gray, Katsuhiro Moizumi, David Kotz, Georg Cybenko, Daniela Rus. Mobile agents in distributed information retrieval. In *Intelligent Information Agents*. Springer Verlag, 1999.
- [BHRS97] Joachim Baumann, Fritz Hohl, Kurt Rothermel, Markus Strasser. Mole - Concepts of a Mobile Agent System. *WWW Journal, Special Issue on Applications and Techniques of Web Agents*, 1997.
- [BK93] Marcos Bafutto, Paul J. Kühn. Capacity and Performance Analysis for Signaling Networks supporting UPT. In *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen*. Springer-Verlag, 1993.
- [Bol89] Gunter Bolch. *Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*. Teubner Verlag, Stuttgart, 1989.
- [Bra97] Jeffrey M. Bradshaw. *Software Agents*. The MIT Press, Menlo Park, Kalifornien, USA, 1997.
- [Bro66] *Brockhaus Enzyklopädie*. F.A. Brockhaus, Wiesbaden, 1966.
- [CAM98] ACTS CAMELEON. An Open Communication Environment Using Agent Technologies. *ACTS 341 - Technical Annex*, März 1998.
- [CAM99] ACTS CAMELEON. *Deliverable D09 - Performance assessment results*. November 1999.
- [CH97] Alper Caglayan, Colin Harrison. *Agent Sourcebook*. John Wiley & Sons Inc., New York, USA, 1997.
- [CHK95] David Chess, Colin Harrison, Aaron Kershenbaum. *Mobile Agents: Are They a Good Idea?* IBM Research Report, 1995.

- [Con98] Javier Conde. *Mobile Agents in JAVA*. European Organization for Nuclear Research, Document CERN/IT/RD45/12, Dezember 1998.
- [DH01] Torsten Dinsing, Jens Hartmann. XML-based freight information over mobile networks. In *IST Mobile Summit 2001*, Seite 440–445, Barcelona, Spanien, September 2001.
- [Döl98] Maik Döll. *Requirement analysis, concept and realisation of an agent-based platform-independent information retrieval system*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, Oktober 1998.
- [Elm01] Martin Elmcrone. *Design, implementation and evaluation of a WAP-based tracking and tracing system*. Diplomarbeit, RWTH Aachen, Informatik IV, Lehr- und Forschungsgebiet Verteilte Systeme, Juli 2001.
- [Eri99a] *WAP Gateway/Proxy 1.0 Description*. Ericsson Radio Systems, Internes Dokument, 1999.
- [Eri99b] *GPRS System Specification*. Ericsson Radio Systems, Internes Dokument, 1999.
- [Eri00] *GATE 2.0 User Guide*. Ericsson Radio Systems, Internes Dokument, 2000.
- [Eri01] *GPRS System Evaluation*. Ericsson Radio Systems, Internes Dokument, 2001.
- [ETS98a] ETSI (European Telecommunications Standards Institute). *General Packet Radio Service, Service Description Stage 1*. GSM Recommendation 03.60, Version 7.0.0. April 1998.
- [ETS98b] ETSI (European Telecommunications Standards Institute). *Mobile Application Part (MAP) specification*. GSM Recommendation 9.02, Version 6.2.0. November 1998.
- [ETS98c] ETSI (European Telecommunications Standards Institute). *General Packet Radio Service, Service Description Stage 2*. GSM Recommendation 03.60, Version 6.2.0. Oktober 1998.
- [ETS99a] ETSI (European Telecommunications Standards Institute). *Bearer Services supported by a GSM PLMN*. GSM Recommendation 02.02, Version 7.0.0. März 1999.
- [ETS99b] ETSI (European Telecommunications Standards Institute). *General on supplementary services*. GSM Recommendation 02.04, Version 8.0.0. März 1999.
- [Eve99] Rune Evensen. *Performance evaluation of agent-based applications for the cellular GSM mobile radio network*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, Mai 1999.
- [FB97] Warwick Ford, Michael S. Baum. *Secure Electronic Commerce*. Prentice Hall PRT, New Jersey, USA, 1997.
- [FG96] Stan Franklin, Art Graesser. Is It an Agent or Just a Program? A Taxonomy for Autonomous Agents. In *3rd International Workshop on Agent Technologies, Architectures and Languages*. Springer-Verlag, 1996.
- [FGS96] William Farmer, Joshua Guttman, Vipin Swarup. Security of Mobile Agents: Issues and Requirements. In *19th National Information Systems Security Conference (NISSC'96)*, 1996.

- [FH00] Wolfgang Frohberg, Jens Hartmann. Agentenkonzept. In *FMC - Konvergenz von Fest- und Mobilfunknetzen*, Seite 72–76. VDE Verlag, Frankfurt, ISBN 3-8007 2536-3, 2000.
- [FIP97] FIPA (Foundation for Intelligent Physical Agents). *FIPA 97 Specification*. Genf, Schweiz, Oktober 1997.
- [FIP98a] FIPA (Foundation for Intelligent Physical Agents). *FIPA 98 Specification, Part 11, Agent Management Support for Mobility*. Genf, Schweiz, Oktober 1998.
- [FIP98b] FIPA (Foundation for Intelligent Physical Agents). *FIPA 97 Specification, Version 2.0, Part 2, Agent Communication Language*. Genf, Schweiz, Oktober 1998.
- [For98] WAP Forum. Wireless Application Protocol Architecture Specification. April 1998.
- [For99] WAP Forum. WAP Push Architectural Overview, WAP-165. November 1999.
- [For00a] WAP Forum. WAP Push Access Protocol Specification WAP-164. Februar 2000.
- [For00b] WAP Forum. Wireless Session Protocol Specification WAP-203-WSP. Mai 2000.
- [Fun00] Funkschau. *GSM-Netze weltweit*. Nr. 16, Juli 2000.
- [Gen00] General Magic Inc. *General Magic Homepage*. <http://www.generalmagic.com/>, 2000.
- [Geu98] Eckhard Geulen. *Modelle zur Realisierung offener Dienste in zellularen Mobilfunknetzen nach dem GSM-Standard*. Verlag der Augustinus Buchhandlung, Aachen, ISBN 3-86073-625-6, Dissertation, RWTH Aachen, Juli 1998.
- [GF92] Michael Genesereth, Richard Fikes. *Knowledge Interchange Format Reference Manual - Version 3*, Stanford University, CSD Tech-Report Logic 92-1, 1992.
- [GH97] Eckhard Geulen, Jens Hartmann. Open Service Provisioning in GSM - What do we gain with CAMEL? In *2. European Personal Mobile Communications Conference (EPMCC'97) und 3. ITG Fachtagung 'Mobile Kommunikation', ITG-Fachbericht 145*, Seite 335–342. VDE-Verlag, Berlin, ISBN 3-8007-2307-7, September 1997.
- [GHN⁺97] Shaw Green, Leon Hurst, Brenda Nangle, Padraig Cunningham, Fergal Somers, Richard Evans. *Software Agents: A review*. Trinity College Dublin, Broadcom Eireann Research Ltd., Mai 1997.
- [GKN⁺96] Robert S. Gray, David Kotz, Saurab Nog, Daniela Rus, George Cybenko. Mobile agents for mobile computing. Dartmouth College, Computer Science, Hanover, USA, Mai 1996.
- [Gra01] Technische Universität Graz. *IAIK SSL Homepage*. <http://jcewww.iaik.tu-graz.ac.at>, Juni 2001.
- [Gun96] Martin Guntermann. *Universelle Benutzermobilität auf der Basis des Intelligen-ten Netzes*. Verlag der Augustinus Buchhandlung, Aachen, ISBN 3-86073-385-0, Dissertation, RWTH Aachen, Juni 1996.
- [Har98] Jens Hartmann. The Alignment of IN and GSM. In *5th International Conference on Intelligence in Networks (ICIN'98)*, Seite 99–104, Bordeaux, Frankreich, Mai 1998.

- [HB99] Jens Hartmann, Steffen Bretzke. Financial services for the future - mobile, flexible, and agent-based. In *2nd International ACTS Workshop: Advanced Services in Fixed and Mobile Telecommunications Networks, Singapur*, September 1999.
- [HEG⁺99] Jens Hartmann, Rune Evensen, Carmelita Görg, Peyman Farjami, Hai Long. Agent-based banking transaction and information retrieval - what about performance issues? In *European Wireless'99 - Mobile Kommunikation, ITG-Fachbericht 157*, Seite 205–210. VDE-Verlag, Berlin, ISBN 3-8007-2490-1, Oktober 1999.
- [Hew77] Carl Hewitt. Viewing Control Structures as Patterns of Passing Messages. *Artificial Intelligence*, Band 8(Nr. 3):Seite 323–364, 1977.
- [HGF98] Jens Hartmann, Carmelita Görg, Peyman Farjami. Agent Technology for the UMTS VHE Concept. In *ACM International Workshop on Wireless Mobile Multimedia (WOWMOM'98)*, Seite 48–57. ACM SIGMOBILE, Dallas, USA, Oktober 1998.
- [HH01] Jens Hartmann, Markus Hück. M-commerce services - what do we gain with mobile agents? In *3G Wireless 2001 - International on Third Generation Wireless and Beyond*, Seite 128–132, San Fransisco, USA, Mai 2001.
- [HK00] Jens Hartmann, Ralf Keller. Nomadic Communication - Impacts of Mobile Agents. In *Computing in Communications (CIC'2000), Las Vegas, USA*. CS-REA Press, ISBN 1-891512-63-7, Juni 2000.
- [HMB98] Lars Hagen, Thomas Magedanz, Markus Breugst. Impacts of Mobile Agent Technology on Mobile Communications System Evolution. *IEEE Personal Communication Magazin*, August 1998.
- [Hoh01] Fritz Hohl. *Sicherheit in Mobile-Agenten-Systemen*. Dissertation, Universität Stuttgart, 2001.
- [HP00] Jens Hartmann, Carsten Pils. Requirements for Disconnected Mobile Agents within Cellular Mobile Telecommunication Systems. In *AMOC 2000 - Asian International Mobile Computing Conference*, Penang, Malaysia, Oktober 2000.
- [HS99] Jens Hartmann, Wei Song. *Agent Technology for Future Mobile Networks*. San Diego, USA, März 1999. Second Annual UCSD Conference on Wireless Communications in cooperation with the IEEE Communications Society.
- [Hüc00] Markus Hück. *Performance evaluation of an agent-based e-commerce application for WAP-enabled mobile communication devices*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, August 2000.
- [IBM01] IBM Research Laboratory. *Aglets Workbench Homepage*. <http://www.trl.ibm.co.jp/aglets/>, Tokio, Japan, Juni 2001.
- [IH99] Leila Ismail, Daniel Hagimont. A Performance Evaluation of the Mobile Agent Paradigm. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Seite 306–313, 1999.
- [IKV01] IKV++ Technologies AG. *Grasshopper Homepage*. <http://www.grasshopper.de>, Berlin, Juni 2001.
- [INT01] INT Media Group, Inc. *SQL Tutorial*. <http://www.sqlcourse.com>, Juni 2001.
- [ITR01] *Internet Traffic Report*. <http://www.internettrafficreport.com>, Juli 2001.

- [ITU93] ITU (International Telecommunication Union). Principles of Telecommunication Services Support by an ISDN and the Means to Describe Them. 1993.
- [Jac90] Van Jacobson. *RFC 1144: Compressing TCP/IP Headers for Low-Speed Serial Links*. <http://www.ietf.org/rfc/rfc1144.txt>, Februar 1990.
- [Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.
- [KG99] David Kotz, Robert S. Gray. Mobile Agents and the Future of the Internet. In *ACM Operating Systems Review*, August 1999.
- [KH01] Birgit Kreller, Jens Hartmann. The field trial scenario of an inter-modal end-to-end and real-time tracking and tracing system. In *8th World Congress on Intelligent Transport Systems*, Sydney, Australien, Oktober 2001.
- [Kin90] Peter J. B. King. *Computer and Communication Systems Performance Modeling*. Prentice Hill, 1990.
- [Kre96] Birgit Kreller. *Mobile Softwareagenten für das effiziente Management der Lastverteilung in Telekommunikationsnetzen*. Diplomarbeit, Universität Magdeburg, 1996.
- [KZHW98] Ralf Keller, Guido Zavagli, Jens Hartmann, Fiona Williams. Mobile Electronic Commerce: GeldKarte Loading Functionality in Wireless Wallets. In *International IFIP/GI Working Conference: Trends in Electronic Commerce (TREC '98)*, Hamburg. ISBN 3-932588-24, Juni 1998.
- [Lan98] Danny B. Lange. Mobile Objects and Mobile Agents: The Future of Distributed Computing. In *European Conference on Object-Oriented Programming*, 1998.
- [LDD95] Anselm Lingnau, Oswald Drobnik, Peter Dömel. An HTTP-based Infrastructure for Mobile Agents. In *The 4th International WWW Conference*, Boston, USA, Dezember 1995.
- [Lem99] Rene Lemle. *Concept and Evaluation of Agent-based Architectures for the Virtual Home Environment (VHE) of Future Mobile Telecommunication Networks (UMTS)*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, September 1999.
- [Lim98] Suriawan Limantara. *Development and Performance Evaluation of an Electronic Payment Mechanism for Mobile Radio Networks*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, August 1998.
- [Lud00] Reiner Ludwig. *Eliminating Inefficient Cross-Layer Interactions in Wireless Networking*. <http://iceberg.cs.berkeley.edu/papers/Ludwig-Diss00/>, Dissertation, RWTH Aachen, April 2000.
- [Lül99] Arnd Lülldorf. *Untersuchung und Bewertung der CORBA Architektur für den Einsatz in zukünftigen Telekommunikationsnetzen*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, Mai 1999.
- [Mat01] MathWorks, Inc. *MATLAB Homepage*. <http://www.mathworks.com/products/>, Juni 2001.
- [Mit98] Mitsubishi Electric ITA, Horizon System Laboratory, Waltham, USA. *Mobile Agent Computing — A White Paper*, Januar 1998.

- [Mit01] Mitsubishi Electric ITA, Horiyon Systems Laboratory. *Concordia Homepage*. <http://www.meitca.com/HSL/Projects/Concordia/>, Juni 2001.
- [MJ00] Richard Murch, Tony Johnson. *Agententechnologie*. Addison-Wesley, ISBN 0-13-011021-3, 2000.
- [MLF96] James Mayfield, Yannies Labrou, Tim Finnin. Evaluation of KQML as an agent communication language. In *Intelligent Agents Vol. II*. Springer Verlag, 1996.
- [MPZ96] Thomas Magedanz, Radu Popescu-Zeletin. Towards 'Intelligence on Demand' - On the Impacts of Intelligent Agents on IN. In *4th International Conference on Intelligence in Networks*, November 1996.
- [MSMO97] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, Teunis Ott. The Macroscopic Behaviour of the TCP Congestion Avoidance Algorithm. *ACM Computer Communication Review, Band 27, Nr. 3*, Juli 1997.
- [Mül96] Jörg Müller. *The Design of Intelligent Agents*. Springer-Verlag, Heidelberg, 1996.
- [MV00] Falk Müller-Veerse. *Mobile Commerce*. Durlacher Research, Marktanalyse, 2000.
- [Net00] Netcraft. *Web Server Survey*. <http://www.netcraft.com/survey>, Oktober 2000.
- [Obj97] ObjectSpace Inc. *Voyager CORBA Integration Technical Overview*. Dallas, USA, 1997.
- [OK97] Mitsuru Oshima, Guenter Karjoth. *Aglets Specification (1.0)*. IBM, May 1997.
- [OMG98] OMG (Object Management Group). *Mobile Agent System Interoperability Facility (MASIF)*, Februar 1998.
- [PFTK98] Jitendra Padhye, Victor Firoiu, Don Towsley, Jim Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *ACM SIGCOMM*, 1998.
- [PH00] Carsten Pils, Jens Hartmann. The User Agent: An approach for service and profile management in wireless access systems. In *International Pacific Rim International Workshop on Intelligent Information Agents (PRIIA 2000)*, Melbourne, Australien, August 2000.
- [PHP01] PHP Group. *PHP Hypertext Preprocessor*. <http://www.php.net>, Juni 2001.
- [PWKH00] Carsten Pils, Michael Wallbaum, Birgit Kreller, Jens Hartmann. ParcelCall: An Open Architecture for Intelligent Tracing Solutions in Transports and Logistics. In *IST Mobile Summit 2000*, Galway, Irland, Oktober 2000.
- [Sch00] Jochen Schiller. *Mobilkommunikation*. Addison-Wesley, München, 2000.
- [Sea98] McGrath Sean. *XML by Example: Building E-Commerce applications*. Prentice Hall, 1998.
- [SM99] Akhil Sahai, Christine Morin. Mobile Agents for Location Independent Computing. In *ACM Symposium on Applied Computing (Special Track on Mobile Computing)*, San Antonio Texas, USA, März 1999.
- [Son98] Wei Song. *Anforderungsanalyse, Konzeption und Implementierung eines agentenbasierten Homebanking-Systems*. Diplomarbeit, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, August 1998.
- [SS97] Markus Schwehr, Markus Straßer. *A Performance Model for Mobile Agent Systems*. 1997.

- [SSK97] Srinivasan Seshan, Mark Stemm, Randy Katz. SPAND: Shared passive network performance discovery. In *USENIX Symposium on Internet Technologies and Systems*, 1997.
- [ST98a] Tomas Sander, Christian Tschudin. Protecting Mobile Agents Against Malicious Hosts. *LNCS on Mobile Agents and Security*, 1998.
- [ST98b] Tomas Sander, Christian Tschudin. Towards Mobile Cryptography. *IEEE Symposium on Security and Privacy*, Mai 1998.
- [Sun95] Sun Microsystems. *The Java Language Specification*, Oktober 1995.
- [T-O01] T-Online International AG. *T-Online*. <http://www.t-online.de>, Juni 2001.
- [Tel00] Rainer Tellmann. *Sichere Kommunikation in Agentensystemen*. Diplomarbeit, Universität Paderborn, September 2000.
- [Wal93a] Bernhard Walke. *Angewandte Informatik 2: Rechnerorganisation und Systemleistung, Skriptum zur gleichnamigen Vorlesung*. RWTH Aachen, Lehrstuhl für Kommunikationsnetze, 1993.
- [Wal93b] Bernhard Walke. *Kommunikationsnetze und Verkehrstheorie II, Verkehrstheoretische Modelle von Echtzeitsystemen und Kommunikationsnetzen, Skriptum zur gleichnamigen Vorlesung*. RWTH Aachen, Lehrstuhl für Kommunikationsnetze, 1993.
- [Wal97] Bernhard Walke. *Kommunikationsnetze und Verkehrstheorie I, Dienste und Protokolle digitaler Kommunikationsnetze, Skriptum zur gleichnamigen Vorlesung*. RWTH Aachen, Lehrstuhl für Kommunikationsnetze, 1997.
- [Wal00] Bernhard Walke. *Mobilfunknetze und ihre Protokolle, Band I & II*. B. G. Teubner, Stuttgart, ISBN 3-519-06430-8, 2000.
- [Whi94] James E. White. *Telescript technology: The foundation for the electronic marketplace – White paper*. General Magic, Inc., 1994.
- [WJ95a] Mike Wooldridge, Nick R. Jennings. *Intelligent Agents, Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin/Heidelberg, Deutschland, 1995.
- [WJ95b] Mike Wooldridge, Nick R. Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, Band 10(Nr. 2), 1995.