

Elektronische Verhandlungen in verteilten E-Business-Systemen

Dissertation zur Erlangung des Doktorgrades
am Fachbereich Informatik der Universität Hamburg

vorgelegt von
Andreas Bartelt
aus Rinteln

Hamburg 2004

Genehmigt vom Fachbereich Informatik
der Universität Hamburg

auf Antrag von Prof. Dr. W. Lamersdorf
und Prof. Dr. A. Rolf

Hamburg, den 29.04.2004

Prof. Dr. Ing. H. S. Stiehl (Dekan)

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit elektronisch unterstützten Verhandlungen als zunehmend wichtigem Teil der elektronischen Unterstützung der Abwicklung von Geschäftstransaktionen in Rechnernetzen (wie z.B. dem Internet). Verhandlungen sind – zumindest implizit – Teil einer jeden solchen Handelstransaktion. Dennoch wird im Bereich des Electronic Business eine angemessene technische (System-) Unterstützung für Verhandlungen bisher nur unzureichend berücksichtigt. Dabei ist zum einen vielfältiger Bedarf dafür gegeben (wie z.B. eine Untersuchung des Bereichs der elektronischen Beschaffung mit geeigneten E-Business-Modellierungsmethoden zeigt). Zum anderen stehen verschiedenartige Techniken bereit, die sich grundsätzlich für eine derartige Unterstützung durchaus eignen (z.B. vielfältige Möglichkeiten der Unterstützung elektronischer Verhandlungen durch lösungs- und prozessorientierte Systeme). Insbesondere bei den prozessorientierten Systemen stehen dabei vor allem die Verhandlungskommunikation sowie deren Standardisierung für (realistische) offene, verteilte Umgebungen im Mittelpunkt. Diese ist im Bereich Electronic Business eng mit den dort bereits vorhandenen Standards des ‚Business-Messaging‘ verknüpft. Durch die Nutzung sprechaktbasierter Systeme kann die Verhandlungskommunikation flexibel und zugleich formalisierbar gestaltet werden. Um der verteilten offenen Umgebung des Electronic Business gerecht zu werden, müssen Verhandlungssysteme zudem sowohl autonom sein als auch gleichzeitig in heterogenen Umgebungen interoperabel interagieren können. Dies kann – wie in der vorliegenden Arbeit gezeigt – durch eine Kombination von sprechaktbasierter Kommunikation und der Verwendung von Ontologien erreicht werden. Es muss dabei u.a. eine Verhandlungsentologie erstellt werden, die auf einem vorher entwickelten und mit relationaler Algebra formalisierten Basismodell für Verhandlungen beruht. Neben der Verhandlungsentologie werden auch für die Vermittlung zwischen den zahlreichen Standards des ‚Business-Messaging‘ Ontologien verwendet. Auf Ontologieebene werden letztendlich die Verhandlungskommunikation und das ‚Business-Messaging‘ miteinander integriert. Schließlich wurde im Rahmen der vorliegenden Arbeit die Realisierbarkeit der hier vorgeschlagenen Kon-

zepte anhand einer prototypischen Implementation auf der Basis einer dafür entwickelten Rahmenarchitektur erprobt und exemplarisch nachgewiesen. Dabei zeigte sich, dass insbesondere die Agententechnologie und Interoperabilität fördernde Technologien wie z.B. Web-Services dabei nutzbringend eingesetzt werden können.

Abstract

This work is concerned with electronically supported negotiations as an increasingly important part of the electronic support of business transactions in computer networks like the internet. Negotiations are – at least implicit – part of each such trade transactions. Nevertheless appropriate negotiation support systems are currently insufficient considered in the area of the electronic business. But there is manifold demand, like an exemplary investigation of the area of electronic procurement shows.

Especially for process oriented negotiation support systems, the negotiation communication as well as its standardization in open distributed environments is in the focus of the analysis. Seen from the domain of electronic business the negotiation communication is closely connected to established and innovative communication standards in this area. These should be integrated with innovative speech act based negotiation communication. Speech act based negotiation communication can be formalised while still being highly flexible. In order to be applicable in the distributed open environment of the electronic business, the negotiation systems must be both autonomous and interoperable in heterogeneous environments. This can be reached by a combination of speech act based communication and the use of ontologies. A negotiation ontology is created, based on a formal model with relational algebra. Business ontologies are used for mediation between the different B2B standards and for integration with the negotiation ontology.

To test the developed concepts and their realizability a prototype framework of a negotiation support system focused on communication was exemplary developed. Thereby agent technology and interoperability technologies like web services proved very beneficial.

Danksagung

Die Möglichkeit zu meiner Doktorarbeit gab mir Prof. Dr. Winfried Lamersdorf. Ich danke ihm für seine zielorientierte Begleitung bei gleichzeitiger Wahrung thematischer Gestaltungsmöglichkeiten. Ich danke Herrn Prof. Dr. Arno Rolf für die Begutachtung meiner Arbeit. Anfangs wurde meine wissenschaftliche Tätigkeit durch Prof. Dr. Hans-Jürgen Appelrath ermöglicht und weiter angeregt durch Prof. Dr. Stefanie Teufel sowie meine Kollegen am Institut OFFIS und an den Universitäten Oldenburg und Hamburg.

Ohne die zahlreichen Studenten und Studentinnen der Universitäten Hamburg und Oldenburg, mit denen ich im Verlauf meiner bisherigen wissenschaftlichen Tätigkeit meine Ideen diskutierte und die mich tatkräftig bei Implementierungsarbeiten unterstützten, wäre dieses Projekt in dem vorliegenden praktischen und theoretischen Umfang nicht möglich gewesen.

Hier sind zunächst Christian Gräfe und Andre Widhani zu nennen, die mit intensivem Engagement ihre Themen bearbeiteten. Weitere wichtige Beiträge erarbeiteten gemeinsam mit mir Afriyie Adwiraah, Andreea Barbu, Stefan Böge, Dimitri Bastian, Heiko Dittrich, Jan Knollmann, Wojciech Laka, Sven Offermann, Dirk Schilling, Björn Waldau und Axel Wriggers.

Dank gilt auch den weiteren, hier ungenannten Studenten und wissenschaftlichen Hilfskräften, die ich bei ihren Studien- und Diplomarbeiten und in Drittmittelprojekten anleiten durfte. Auch meine Lehrtätigkeiten prägten die Inhalte meiner Dissertation.

Besonderer Dank gilt meiner Frau Annette Sturm für ihre vielfältige ausdauernde Unterstützung und Flexibilität sowie meiner Tochter Sophie, die in der Promotionszeit zur Welt kam als einer andauernden Quelle der Lebensfreude.

Hamburg, im Oktober 2003

Eidesstattliche Erklärung

Erklärung an Eides statt nach §7(1)c der Promotionsordnung

Ich, Andreas Bartelt, geboren am 30.09.1967 in Rinteln, erkläre an Eides statt, dass ich meine vorliegende Dissertation selbst verfasst habe und dabei keine anderen als die angegebenen Hilfsmittel benutzt habe.

Wardenburg, den 15.11.2003

Bartelt

Inhaltsverzeichnis

Zusammenfassung	3
Abstract	5
Danksagung	6
Eidesstattliche Erklärung.....	7
Tabellenverzeichnis.....	13
Abbildungsverzeichnis.....	14
1 Einleitung	18
2 Electronic Business: Klassifikationen und Modellierungsmethoden	21
2.1 Charakteristika des Electronic Business	21
2.1.1 Electronic Business und Elektronische Märkte	22
2.1.2 Kriterien und Klassifikationen.....	26
2.1.3 E-Business-Modelle	29
2.1.4 Technologien für E-Business-Systeme	49
2.2 Modellierungsmethoden im Electronic Business	51
2.2.1 Modellierung von Geschäftsmodellen und Softwarelösungen im E-Business	52
2.2.2 E3-Value-Modellierung.....	69
2.2.3 Analyse der E-Business-Modellierungsmethoden.....	76
2.2.4 E-Business-Geschäftsprozessmodellierung	81
2.3 E-Procurement als typischer Bereich des E-Business	85
2.3.1 Beschaffung in Unternehmen	85
2.3.2 Elektronische Beschaffung	87

2.3.3	Systeme für die elektronische Beschaffung	93
2.3.4	Modellierungsmethoden im E-Procurement	99
2.3.5	Verhandlungsaspekte im E-Procurement	108

3 Elektronische Verhandlungen: Grundlegende Strukturen und Unterstützungssysteme 112

3.1	Charakterisierung von Verhandlungen	112
3.2	Klassifikation von Verhandlungen	114
3.2.1	Klassifikationskriterien	114
3.2.2	Klassifikationssichten	118
3.2.3	Verhandlungen im Electronic Business	119
3.3	Basismodell für Verhandlungen im Electronic Business	120
3.3.1	Basismodell.....	120
3.3.2	Ablauf einer Verhandlung.....	138
3.4	Verhandlungsstrategien	144
3.4.1	„Negotiation-Analysis“	145
3.5	Elektronisch unterstützte Verhandlungen	152
3.5.1	Komplexität der Verhandlungsform	152
3.5.2	Art der Unterstützungsfunktionalität.....	154
3.5.3	Verhandlungsunterstützungssysteme	160
3.5.4	Funktionen zur Verhandlungsunterstützung	168

4 Kommunikationsaspekte in Verhandlungssystemen.....170

4.1	Fallbeispiele für Verhandlungssituationen und Verhandlungskommunikation.....	170
4.1.1	Ein grundlegendes Fallbeispiel: Preisagentur Spürhund GbR	171
4.1.2	Freie Verhandlungen bei der Fahrzeugpool GmbH	174
4.1.3	Feste Protokolle bei den Stadtwerken Tüchtigburg	176
4.2	Business Messaging.....	178
4.2.1	Strukturen des Business Messaging	178
4.2.2	Klassifikation der Standards des Business Messaging... ..	181
4.2.3	Überblick über die B2B-XML-Standards	182
4.3	Modellierung von Verhandlungskommunikation	195
4.3.1	Grundlagen der Sprechakttheorie.....	195
4.3.2	Konstitutive Regeln für Sprechakte	200
4.3.3	Modellierungen auf Basis der Sprechakttheorie	203

5 Systemtechnische Infrastrukturen zur Durchführung von Verhandlungsinteraktionen 206

5.1	Ontologien als Infrastrukturen.....	207
5.1.1	Grundlagen von Ontologien	207
5.1.2	Entwurf von Ontologien	210
5.2	Verhandlungsontologie.....	212
5.2.1	Anpassung der Entwurfsmethoden auf die Verhandlungsontologie	212
5.2.2	Konzepte der Verhandlungsontologie	214
5.3	Vermittlung zwischen B2B-Standards	223
5.3.1	Ebenen der Vermittlung.....	223
5.3.2	Repräsentationstechniken von Konzepten im B2B Umfeld	226
5.4	Entwurf der B2B-Ontologie	230
5.4.1	Entwurfsmethoden für B2B-Ontologien.....	231
5.4.2	Exemplarische B2B-Ontologie	235
5.5	Integration von Verhandlungskommunikation und B2B-Messaging.....	241
5.6	Verhandelbare Verhandlungsprotokolle.....	247
5.6.1	Fallbeispiel für Bootstrapping	247

6 Implementation verhandlungsspezifischer Infrastrukturen und Systeme 251

6.1	Wichtige Technologien für die Realisierung verhandlungsspezifischer Systeme	251
6.1.1	Agententechnologie	252
6.1.2	Systemtechnische Interoperabilität mit Web-Services ..	259
6.2	Architektur einer Systeminfrastruktur zur Unterstützung von Verhandlungen	267
6.2.1	Die Grundarchitektur.....	267
6.2.2	Nachrichtenfluß im EPN-System	281
6.2.3	Architektur der B2B-Brokerkomponente	286
6.2.4	Implementation des Bootstrapping-Konzeptes.....	292
6.2.5	Realisierung von Verhandlungsunterstützungsfunktionen für den E-Business.....	296
6.2.6	Elemente systemtechnischer Kommunikation und Interoperabilität	300

7 Resümee und Ausblick 304

A Kommunikation.....309

A.1	Beispiele zur B2B-Kommunikation.....	309
A.1.1	EDIFACT-Beispiel	309
A.2	Kommunikation im EPN-System - Dialogbeispiele	311
A.2.1	Erfolgreiche Ausführung des Request-Protokolls.....	311
A.2.2	Request-Protokoll endet durch Refuse.....	312
A.3	WSDL-Dokument eines verhandlungsfähigen Webservices	313

B Implementationen 316

B.1	Zentrale Klassen und Schnittstellen des EPN-Frameworks...	316
B.1.1	Die Klasse epn.Boot	316
B.1.2	Die Klasse epn.system.EPNController	317
B.1.3	Die Klasse epn.system.mc.MessageCoordinator	318
B.1.4	Die Klasse epn.system.b2badapter.B2BAdapter	318
B.1.5	Die Schnittstelle epn.system.EPNAgent	318
B.1.6	Die Klasse epn.system.EPNAgentBehaviour	319
B.1.7	Die Klasse epn.system.SimpleEPNAgent.....	320
B.1.8	Die Schnittstelle epn.system.Constants	321
B.1.9	Die Klasse epn.util.RequestResponder	321

Literatur 324

Tabellenverzeichnis

Tabelle 2.1: Klassifikationskriterien für Geschäftsmodelle der Electronic Information.....	34
Tabelle 2.2: Weitere literaturbezogene Referenzmodelle mit Beispielen konkreter Ausprägungen	36
Tabelle 2.3: Referenzmodelle geschäftlicher Informationen.....	38
Tabelle 2.4: Phasen der Bedarfsdeckung.....	39
Tabelle 2.5: Vergleich der Modellelemente von e ³ -Value, E-BPMS und RSD	78
Tabelle 2.6: Vergleich wichtiger Eigenschaften von e ³ -Value, E-BPMS und RSD	80
Tabelle 2.7: Beschaffung indirekter und direkter Produkte (vgl. (Dolmetsch, 2000)).....	86
Tabelle 3.1: Einordnung und Strukturierung von Verhandlungsaktivitäten im Electronic Business	140
Tabelle 3.2: Beispiel zum Nash-Equilibrium	146
Tabelle 3.3: Beispiel zum Nash-Equilibrium (2 – Prisoners-Dilemma)	147
Tabelle 4.1: Einordnung und Vergleich gängiger B2B-Standards	192
Tabelle 6.1: Startparameter des EPN-Systems	271

Abbildungsverzeichnis

Abbildung 2.1: Klassifikation von Geschäftsmodellen nach Wirtschaftssubjekten und Verhalten der Akteure	28
Abbildung 2.2: Erlösquellen, vgl. (Skiera, 2000)	30
Abbildung 2.3: Informationszentriertes Phasenmodell	33
Abbildung 2.4: Beispiel EI-Phasenmodell	34
Abbildung 2.5: Digitale Bibliotheken	36
Abbildung 2.6: E-Portal	37
Abbildung 2.7: Beispiel für Phasenmodell einer Handelstransaktion	39
Abbildung 2.8: Phasenmodell für E-Shops	41
Abbildung 2.9: Online-Shop Szenario	41
Abbildung 2.10: Phasenmodell für E-Auctions	43
Abbildung 2.11: Phasenmodell für E-Tendering	44
Abbildung 2.12: Phasenmodell für E-Portals	45
Abbildung 2.13: Architekturskizze <i>PublicationPORTAL</i>	46
Abbildung 2.14: Phasenmodell für E-Malls	47
Abbildung 2.15: Phasenmodell für Shopping-Agents	48
Abbildung 2.16: Phasen und Workflows beim RUP	52
Abbildung 2.17: Business-Modelling-Workflow im RUP	54
Abbildung 2.18: Regelkreis zum Geschäftsprozessmanagement nach dem E-BPMS-Paradigma	57
Abbildung 2.19: Das im (Meta-)Geschäftsprozessmanagement-Werkzeug ADONIS® umgesetzte E-BPMS-Framework	58
Abbildung 2.20: Modellierungsebenen im E-BPMS Framework	59
Abbildung 2.21: Die sieben ‚Cornerstones‘ des RSD-Frameworks	62
Abbildung 2.22: RSD-Metamodell zur Erfassung von Networked- Enterprises	64
Abbildung 2.23: Konzepte in der Akteur-Domäne	65
Abbildung 2.24: Konzepte in der Rollen-Domäne	66
Abbildung 2.25: Metamodell für Verläufe, Rollen und Funktionen	67
Abbildung 2.26: Konzepte in der Funktionen-Domäne	67
Abbildung 2.27: Konzepte der Prozess-Domäne	69
Abbildung 2.28: Sichten bei der Entwicklung von E-Business-Lösungen	70
Abbildung 2.29: E ³ -Value-Metamodell / Ontologie	76
Abbildung 2.30: Einordnung E-Procurement	90
Abbildung 2.31: Anbieterzentriertes Beschaffungsmodell	90
Abbildung 2.32: Käuferzentriertes Beschaffungsmodell	91
Abbildung 2.33: Marktplatz/ ‚Broker‘ als Mittler der Akteure	92

Abbildung 2.34: Komponentenübersicht eines Desktop-Purchasing- Systems (Nenninger, 1999)	95
Abbildung 2.35: e ³ -Value-Modell E-Procurement	101
Abbildung 2.36: EP-Prozessmodell (Übersicht)	104
Abbildung 2.37: EP-Prozessmodell-Sourcing / Genehmigung	105
Abbildung 2.38: EP-Prozessmodell-Anforderung / Bestellung.....	106
Abbildung 2.39: EP-Prozessmodell-Lieferung / Empfang.....	107
Abbildung 2.40: EP-Prozessmodell-Rechnungsprüfung / Bezahlung ...	108
Abbildung 3.1: Wichtige Elemente des Basismodells.....	129
Abbildung 3.2: Zusammenhänge zwischen globalen und internen Protokollen sowie Strategien am Beispiel eines Akteurs	133
Abbildung 3.3: Angebots- und Einigungsmengen	134
Abbildung 3.4: Utility Set und Pareto-Optimalität (nach (Ehtamo, et al., 2002))	146
Abbildung 4.1: E ³ -Value Modell Fallbeispiel Spürhund GbR	172
Abbildung 4.2: Ablauf Fallbeispiel Spürhund GbR	173
Abbildung 4.3: Ablauf Fallbeispiel Fahrzeugpool GmbH.....	175
Abbildung 4.4: Ablauf Fallbeispiel Stadtwerke Tüchtigburg.....	177
Abbildung 5.1: Zentrale Konzepte und Klassen der Verhandlungsontologie	215
Abbildung 5.2: UBL-Adresselement mit Kontext.....	240
Abbildung 5.3: Zusammenhang zwischen Verhandlungs- und B2B- Nachrichten	243
Abbildung 5.4: Ausgehende Nachrichten im EPN-System	245
Abbildung 5.5: Eingehende Nachrichten im EPN-System.....	246
Abbildung 5.6: Beispiel für Bootstrapping am Fallbeispiel Stadtwerke Tüchtigburg	248
Abbildung 6.1: Thematische Struktur der FIPA-Standards.....	257
Abbildung 6.2: Übergänge von der abstrakten zu konkreten Architekturen	258
Abbildung 6.3: Aufbau einer SOAP-Nachricht (Oracle, 2002).....	262
Abbildung 6.4: Struktur eines WSDL-Dokumentes (Schmelzer, et al., 2002)	264
Abbildung 6.5: Ablauf Web-Services	266
Abbildung 6.6: Komponenten des EPN-Systems.....	268
Abbildung 6.7: Paketstruktur des Frameworks (Übersicht)	270
Abbildung 6.8: FIPA Request Interaction Protocol.....	273
Abbildung 6.9: Hierarchie der Ontologien.....	274
Abbildung 6.10: JOCEP.....	279
Abbildung 6.11: Zusätzliche Klasseneigenschaften unter JOCEP	280
Abbildung 6.12: Sequenzdiagramm: Business-Dokument trifft vor der Verhandlungsnachricht ein.....	285
Abbildung 6.13: Sequenzdiagramm: Verhandlungsnachricht trifft vor dem Business-Dokument ein.....	285
Abbildung 6.14: Datenfluss in der Brokerkomponente	286
Abbildung 6.15: Merging-PlugIn für Protege	291
Abbildung 6.16: Bildschirmausgabe für einen Agenten-Task beim Borda- Voting.....	294
Abbildung 6.17: Inhalt einer JAS-Transport-Message	295

Abbildung 6.18: SPNSS – Beispiele.....	298
Abbildung 6.19: SPNSS - Architektur.....	299
Abbildung 6.20: Ablauf der Kommunikation.....	300
Abbildung 6.21: Testclient des Web-Service für Verhandlungskommunikation.....	302

1 Einleitung

Electronic Commerce und Electronic Business haben sich in ihren Grundformen an der Schwelle zum 21. Jahrhundert flächendeckend etablieren können. Dabei stehen bisher einfache Anwendungen der Informationsversorgung und des direkten Verkaufs von Gütern und Dienstleistungen im Vordergrund. Weitreichende Defizite bestehen jedoch in der Tiefe der bisherigen Konzepte und der eingeschränkten Komplexität der Anwendungen.

Diese Defizite lassen sich gezielt identifizieren und bearbeiten:

Struktureller Gegenstand des Electronic Business ist die systematische Transition herkömmlicher Geschäftsbeziehungen auf das Medium des Internets sowie die Identifikation und technische Umsetzung neuer Geschäftsmöglichkeiten, die sich erst bei Anwendung neuer Technologien und Konzepte der Informatik ergeben.

Bei methodischer Betrachtung von Geschäftstransaktionen sowohl in konventionellen als auch in neuen elektronischen Geschäftsbeziehungen lässt sich jede Transaktion in die Phasen der Information, Vereinbarung und Abwicklung einteilen. Die Vereinbarungsphase bietet den beteiligten Parteien die Möglichkeit ihre meist unterschiedlichen Ausgangspositionen in einer Verhandlung einander anzunähern. Dies ist die Voraussetzung für das Zustandekommen vieler konventioneller Handelstransaktionen. Obwohl die Vereinbarungsphase für herkömmliche Geschäftsbeziehungen außerordentlich wichtig ist, wird sie im Bereich der elektronischen Geschäftsbeziehungen nur in sehr geringem Umfang technisch angemessen unterstützt.

Die aktuelle Forschung im betrachteten Gebiet fokussiert im Wesentlichen einerseits auf den automatisierten Einsatz von Auktionen unter vorrangiger Betrachtung multiattributiver Auktionen und andererseits auf webbasierte Verhandlungsunterstützungssysteme. Die isolierte Bearbeitung dieser Forschungsbereiche führt jedoch nicht zu effektiven Lösungen für elektronische Verhandlungen im E-Business.

Daher werden in dieser Arbeit unter Einbeziehung der genannten Forschung vor allem die offenen Problembereiche

- einer umfassenden Betrachtung von elektronischen Verhandlungen in Handelstransaktionen des Electronic Business und
- einer systemtechnischen Infrastruktur für die Verhandlungskommunikation im Electronic Business

untersucht und mit dem Ziel einer konstruktiven technischen Systemunterstützung bearbeitet.

Um dabei eine möglichst umfassende Betrachtung zu gewährleisten und so die Grundlage für den Entwurf einer adäquaten Verhandlungsinfrastruktur zu legen, bedarf es mehrerer umfangreicher Schritte, die beginnend im E-Business-Bereich über Verhandlungen zur Verhandlungskommunikation im E-Business führen:

Als erstes wird analysiert, an welcher Stelle im Verlaufe von anwendungsbezogenen Handelstransaktionen im Electronic Business welcher Bedarf an Verhandlungen besteht. Dazu ist der Bereich des Electronic Business entsprechend zu strukturieren. Bei der Strukturierung können adäquate Methoden für die E-Business und Verhandlungsmodellierung helfen, die recherchiert, analysiert und neu geschaffen werden müssen. Diese Modellierungsmethoden sollen auch die spätere Entwicklung von E-Business und Verhandlungssystemen unterstützen.

Aus diesen Analysen des Bereiches E-Business resultiert das spezielle Geschäftsmodell des Electronic Procurement als eine besonders interessante Anwendungsdomäne, auf die in dieser Arbeit ein besonderer Anwendungsschwerpunkt gelegt wird. Im E-Procurement, also der elektronischen Beschaffung von indirekten Gütern, sind Verhandlungen von besonderer Wichtigkeit. Es kann dabei zu multilateralen, multiattributiven Verhandlungen kommen, um sich über die Konditionen des Handels zu einigen. So geht es beispielsweise um eine Einigung über den Preis, den Zeitpunkt der Lieferung oder besondere Eigenschaften eines Gutes.

Der nächste wichtige Schritt ist die Klärung der Möglichkeiten zur Unterstützung von elektronischen Verhandlungen mit Verhandlungssystemen. Dazu wird ein generisches Basismodell für elektronische Verhandlungen im E-Business als Grundlage für die weitere nachfolgende Arbeit entwickelt. Bestehende lösungs- und prozessorientierte Ansätze und Systeme für die elektronische Verhandlungsunterstützung werden untersucht, adaptiert und weiterentwickelt.

Die Realisierung der notwendigen Kommunikation in Verhandlungssituationen ist einer der wichtigsten Aspekte bei den oben betrachteten Verhandlungssystemen. Eine in der Praxis bereits vorhandene Grundlage für

die interorganisationelle Kommunikation zwischen E-Business-Systemen ist das Business Messaging, das derzeit über XML-basierte B2B-Standards für Kommunikationsframeworks und Geschäftsdokumente realisiert wird. Es wird mit der Verhandlungskommunikation so verbunden, dass auf allen Kommunikationsebenen Kompatibilität zu den bestehenden Systemen gewährleistet ist.

Als Grundlage für die Realisierung der Verhandlungskommunikation ist zunächst nach geeigneten Modellierungsmöglichkeiten zu suchen. Dabei erwies sich die Sprechakttheorie als ein nützliches und grundlegendes Instrument, da sie den autonomen Interaktionscharakter der Verhandlungskommunikation adäquat abbildet.

Als eine weitergehende innovative Möglichkeit wird zum flexiblen Einsatz von Verhandlungsprotokollen das Bootstrapping-Konzept entworfen, das eine hochdynamische Konfiguration von Verhandlungsabläufen in E-Business-Systemen ermöglicht.

Die entwickelten Möglichkeiten zur Verhandlungskommunikation benötigen eine geeignete Infrastruktur zur konkreten Durchführung von Verhandlungen. Diese fehlt jedoch weitgehend in derzeitigen E-Commerce Systemen. Eine interoperable Verhandlungsinfrastruktur ist unabdingbare Voraussetzung für eine effektive und effiziente teilautomatisierte Unterstützung der Vereinbarungsphase im Electronic Business. Die angestrebte Verhandlungsinfrastruktur basiert auf der sprechaktbasierten Kommunikation, die innovative und für eine Verhandlungssprache adäquate Grundlagen bietet. Die sprechaktbasierte Kommunikation hat auch in der Agententechnologie Verwendung gefunden; die dortigen Lösungen sind jedoch in der Umgebung des Electronic Business nicht praktikabel umsetzbar. Daher wird im Kontext der verschiedenen B2B-Standards des Business Messaging eine E-Business Verhandlungssprache entworfen, umgesetzt und in B2B-Standards und E-Business-Systemen in einer heterogenen Kommunikationsumgebung integriert.

Damit soll die Arbeit durch eine umfassende Unterstützung der Verhandlungsphase bei Handelstransaktionen zur Vervollständigung von Systeminfrastrukturen des Electronic Business beitragen.

2 Electronic Business: Klassifikationen und Modellierungsmethoden

Eine Definition von Electronic Business ist: *„Electronic Business ist die elektronische Unterstützung von Wirtschaftstätigkeiten, insbesondere durch Internettechnologien“.*

Um den Bereich des Electronic Business tiefer zu erschließen, kann er nach verschiedenen Kriterien wie Anwendergruppen, Technologien oder der Art der unterstützten Wirtschaftstätigkeiten klassifiziert werden. Die vielfältigen Kriterien führen hin zu einer Modellbildung im Electronic Business, die notwendig ist, um Anwendungen und Systeme des Electronic Business zu planen, zu simulieren, umzusetzen, zu analysieren und evaluieren zu können. Die Modellierungsmethoden im E-Business befinden sich in einem frühen Entwicklungsstadium und werden als Basis für die weitere Arbeit im E-Business-Bereich eingehend erklärt. Die Anwendungen des Electronic Business lassen sich durch typische Modelle untersuchen und gestalten, wobei nach einem umfassenden Überblick und einer Fokussierung auf den Bereich des Electronic Commerce im Detail auf das Modell des E-Procurement eingegangen wird. Das E-Business-Modell des E-Procurement ist gut geeignet, um die Integration beteiligter Akteure zu untersuchen. Diese Integration, gerade *mit* und *für* Verhandlungen, macht Wirtschaftstätigkeiten erst möglich.

2.1 Charakteristika des Electronic Business

Zu den Grundlagen des Electronic Business zählen die Wirtschaftswissenschaften (Betriebswirtschaft, Volkswirtschaft und auch entsprechende juristische Aspekte), die Informatik, insbesondere mit den für die Realisierung wichtigen Internettechnologien, sowie Sicherheitsaspekte und elektronische Zahlungssysteme.

In diesem Abschnitt wird der Electronic Business begriffsklärend und konzeptionell mit den technischen Anforderungen und Umsetzungen vorgestellt.

2.1.1 Electronic Business und Elektronische Märkte

Die intensive kommerzielle Nutzung des World Wide Web begann Mitte der 90er Jahre. Zunächst wurde der Electronic Commerce mit seinem zentralen Geschäftsmodell des Online-Shops als bedeutende Anwendung gesehen. Im Laufe der Zeit erweiterten sich die Anwendungsmöglichkeiten und Betrachtungsweisen und es wurde intensiv nach neuen Erfolg versprechenden Geschäftsmodellen gesucht.

Der noch breiter angelegte Begriff des *Electronic Business* wurde erst 1998 überwiegend von IBM geprägt.

2.1.1.1 Begriffswelt des Electronic Business

Wenn von Electronic Business gesprochen wird, ist im Wesentlichen die Rede von der elektronischen Unterstützung von Wirtschaftstätigkeiten, insbesondere durch die Internettechnologien. Da jedoch der Begriff Electronic Business und die angrenzenden Begriffe intensiv als Schlagworte und als überladene Marketingkonzepte genutzt werden, ist die Bedeutung oft unterschiedlich und nur unscharf definiert. In dieser Arbeit soll Electronic Business als Oberbegriff stehen, der alle anderen Teilbereiche umfasst.

In der aktuellen Forschung werden bei der Betrachtung des E-Business unterschiedliche Akzente gesetzt.

So sieht Thome (Thome, 2002) *E-Business* als die „integrierte Ausführung aller digitalisierbaren Bestandteile ökonomischer Prozesse“. Mit *integriert* wird besonderer Wert auf eine medienbruchsfreie, rechnerbasierte und nach Möglichkeit automatisierte Reaktion auf eingehende Informationen und einer Ausgabe möglichst ebenso weiterverarbeitbarer Ergebnisse gelegt. Während die digitalisierbaren Bestandteile im bekannten Sinne und Umfang interpretierbar sind, wird mit ökonomischen Prozessen der Umfang des e-Business berechtigterweise recht weit gefasst. Als Ökonomische Prozesse werden alle Anstrengungen bezeichnet, bei denen Wirtschaftssubjekte Güter im Ausgleich für Gegenleistungen abgeben. Der Prozess des Auffindens eines geeigneten Partners wird als der aufwändigste Teil angesehen, der gestützt durch die elektronische Informationsverarbeitung wesentlich optimiert werden kann. Thome argumentiert des Weiteren, dass wesentliche Kostenersparnisse wiederum erst durch die oben genannte integrierte Ausführung erreicht werden können,

deren Voraussetzung eine organisatorische Anpassung und die Vereinheitlichung von Datenstrukturen über Unternehmensgrenzen hinweg seien.

Zwißler (Zwißler, 2002) bezeichnet *Electronic Business* als Oberbegriff unterschiedlicher Formen des Einsatzes elektronischer Kommunikations- und Kooperationsmechanismen zur Realisierung, Unterstützung und Optimierung von Geschäftsprozessen. So rückt sie die Betrachtung der elektronischen Kommunikations- und Kooperationsmechanismen stärker in den Vordergrund.

Merz et al. (Merz, 1999; Merz, 2002; Merz, et al., 1999) geben detaillierte Überblicke zum Gesamtthema des Electronic Commerce und Electronic Business:

- Softwareschnittstellen und Frameworks
- Elektronischer Datenaustausch
- Dokumentenformate
- Zahlungsverfahren
- Vertrauensinfrastrukturen
- Kryptographie
- Juristische Infrastrukturen
- Messstandards.

2.1.1.2 Elektronische Märkte

Elektronische Märkte sind ein Kernelement des Electronic Business. Sie ermöglichen die Abwicklung von Marktprozessen über elektronische Medien, insbesondere über das Internet mit seinem breiten Partizipationspektrum. Durch die Nutzung des neuen Mediums ergeben sich Charakteristika, von denen hier wichtige genannt werden.

Zunächst ist es für die auftretenden Effekte wichtig, inwieweit das elektronische Medium genutzt wird. Kommt es während der Abwicklung der Marktprozesse zu einem Wechsel des Mediums, beispielsweise bei der logistischen Auslieferung physischer Güter, so spricht man von einem *Medienbruch*. Je weniger oder geringer intensiv die Medienbrüche auftreten, umso stärker wirken sich die Effekte Elektronischer Märkte aus.

Eine besondere Möglichkeit, die sich durch die Nutzung Elektronischer Märkte ergibt, ist die *Disintermediation*. Dabei handelt es sich um den Wegfall von einzeln ausgeprägten Stufen der Wertschöpfungskette. Ermöglicht wird dies prinzipiell durch eine erhebliche Reduktion oder den vollständigen Wegfall des für eine Stufe nötigen Aufwandes. Ein eminentes Beispiel dafür sind die in den letzten Jahren entstandenen Direktbanken, die völlig auf den Filialbetrieb verzichten. Funktionen wie die Erfassung von Überweisungsträgern sind weitgehend ins Internet und auf die Kunden verlagert und die notwendigen Grundfunktionen, wie bei-

spielsweise die Bereitstellung von Geldautomaten, werden anderweitig eingekauft.

Unter dem Gesichtspunkt der *Effizienz* ergeben sich weitere wichtige Aspekte internetbasierter Märkte im Vergleich zu herkömmlichen Märkten. Hier geht es nach Smith et al. (Smith, et al., 1999) insbesondere um die folgenden vier Effizienz-Dimensionen:

- Preishöhe ('Price Levels')
- Preissensitivität (,Price Elasticity')
- Preisanpassungskosten (,Menu Costs')
- Preisverteilung (,Price Dispersion').

Preishöhe: In klassischen ökonomischen Wohlfahrtsmodellen wird die Effizienz optimiert, wenn alle Wohlfahrt erhöhenden Handelstransaktionen durchgeführt werden. Dazu müssen in den hier betrachteten Märkten die Preise den Grenzkosten der Verkäufer entsprechen. Dies kann wiederum erreicht werden, wenn nur möglichst geringe Suchkosten für die Käufer entstehen (Bakos, 1997). Und gerade hier greifen elektronische Märkte; sie ermöglichen einen schnellen Überblick über Produktangebote und Preise und reduzieren die Informationsasymmetrie zwischen Käufern und Verkäufern erheblich.

Preissensitivität: Die Preiselastizität oder -sensitivität ist ein Indikator dafür, wie stark Käufer auf Preisänderungen reagieren. In effizienten Märkten reagieren Käufer schneller und intensiver auf verschieden hohe Preise. Durch geringe Such- und Wechselkosten können elektronische Märkte zu hoher Preissensitivität beitragen. Jedoch kann eine gut realisierte Kundenbindung und gezielt abgestimmte Produktinformation dieser Preissensitivität entgegenwirken.

Preisanpassungskosten: In einem effizienten Markt müssen die Verkäufer ihre Preise flexibel anpassen können. Eine Preisanpassung verursacht jedoch Kosten, sei es allein durch die Neuauszeichnung von Preisen an den Waren oder durch Mitteilungen an die Kunden. Elektronische Märkte senken in der Regel diese Preisanpassungskosten durch die Unterstützung durch die kostengünstige Informationstechnologie.

Preisverteilung: Das Bertrand-Model des Preiswettbewerbs geht von einem extremen Gesichtspunkt auf die Markteffizienz aus. Produkte sind homogen, die Käufer sind genau über alle Preise und Produkte informiert, die Suchkosten sind damit gleich Null usw. Dies würde dazu führen, dass sich für ein bestimmtes Produkt nur ein Preis etabliert. Doch die Märkte sind nicht so ‚ideal‘ wie das Bertrand-Model annimmt und so treten für das gleiche Produkt verschiedene Preise auf. Als ein wichtiger Hauptgrund dafür gelten die Suchkosten. Da die Suchkosten jedoch von elektronischen Märkten gesenkt werden, ist hier mit einer geringeren

Preisverteilung zu rechnen. Mehrere Untersuchungen zeigen jedoch andere Ergebnisse (Smith, et al., 1999). Als Gründe werden dafür unter anderem ein noch nicht voll entwickelter Markt, verschieden vertrauensvolle Verkäufer oder auch die verschieden stark ausgeprägte Aufmerksamkeit der Käufer genannt. Eine endgültige Entwicklung ist hier noch nicht abzusehen.

Die Effizienz elektronischer Märkte kann durch den Einsatz von Verhandlungssystemen weiter intensiviert werden. Da die beteiligten Parteien in der Verhandlung gezielt Informationen austauschen können Informationsasymmetrien abgebaut werden. So kann letztendlich eine effiziente Preishöhe erreicht werden. Verhandlungen ermöglichen ebenfalls die sinnvolle Umsetzung von Preissensitivität, da Preise auch mit dem ursprünglichen Anbieter verhandelt werden können. Die Preisanpassungskosten werden weiter gesenkt, da sie idealerweise nur den Verhandlungskosten entsprechen müssen. Verhandlungen können in unterschiedlicher Weise zur Preisverteilung beitragen. Einerseits ermöglichen sie individuelle subjektiv bewertete Preise, andererseits können für homogene Produkte wie im Bertrand-Model angesprochen, gleiche Preise leichter erreicht werden.

Zur *Realisierung* Elektronischer Märkte sind entsprechende Konzepte, Infrastrukturen und Standards sowie Technologien notwendig, über die in Abschnitt 2.1.4 ein Überblick gegeben wird. In Hauptteil dieser Arbeit wird jedoch insbesondere die nötige technische Infrastruktur Elektronischer Märkte im Hinblick auf alle auftretenden Verhandlungsaspekte vertieft.

2.1.1.3 Vorteile des Electronic Business

Durch den Einsatz des elektronischen Mediums, insbesondere des Internets, zur Planung und Durchführung von Geschäftsabläufen und -kommunikation ergeben sich wesentliche Vorteile.

Ein dominierender Vorteil ist die *Kostensenkung*, bei der Durchführung der Geschäftsabläufe. Sie basiert in großen Teilen auf der Ersetzung realer Dokumente durch elektronische Dokumente, die kostengünstiger übertragen, verarbeitet und verwaltet werden können. Auch der Automatisierungsgrad der Abläufe kann somit oft erhöht werden.

Ein weiterer wesentlicher Vorteil ist die *hohe Geschwindigkeit*, mit der Geschäftsabläufe durchgeführt werden können. Dies führt unter anderem zu einer verkürzten Reaktionszeit und einem möglicherweise höheren Durchsatz mit Steigerung der Leistungsfähigkeit.

Weitere Vorteile sind (vgl. (Zwißler, 2002)):

- Eine vergrößerte *Robustheit* der Systeme und Abläufe, da eine automatische Verarbeitung und Überprüfung häufig weniger fehleranfällig ist als eine manuelle Bearbeitung.
- Eine höhere *Transparenz*, da durch die digitale Datenhaltung die Vorgehensweisen und Entscheidungsgrundlagen auf Wunsch leicht offen gelegt und im Sinne von MIS/EIS analysiert werden können.
- Eine verbesserte *Interaktivität*, da elektronische Medien schnelle und direkte Reaktionen erlauben.
- *Explizite Mehrwerte* können sich ergeben, wenn durch die Verknüpfung unterschiedlicher Daten und Verarbeitungsschritte neue innovative, bisher nicht verfügbare Funktionalitäten realisiert werden können.

2.1.2 Kriterien und Klassifikationen

Die Klassifikation von Gegebenheiten und Aktivitäten im Electronic Business anhand verschiedener Kriterien dient der Analyse des Themenbereichs und ermöglicht dadurch eine gezielte Bearbeitung der für die Thematik dieser Arbeit relevantesten Bereiche.

Um einen übersichtlichen Einstieg zu geben, sollen zunächst mit der *Electronic Information*, dem *Electronic Commerce* und der *Electronic Cooperation* (Bartelt, et al., 2000b) einige Schwerpunktbereiche mit hoher Aktualität herausgearbeitet werden.

Der Teilbereich der *Electronic Information* (EI) (Bartelt, et al., 2001c) bezieht sich auf den elektronisch unterstützten *Austausch von Informationen*, also auf alle Vorgänge im Electronic Business, die primär der Informationsvermittlung dienen. Wichtige Beispiele aus diesem intensiv bearbeiteten Bereich sind Digitale Bibliotheken (Bartelt, et al., 2001a) und die inzwischen breit diversifizierten Web-Portale.

Ein weithin bekannter Teilbereich des Electronic Business ist der *Electronic Commerce*, der elektronisch unterstützte *Handel* (Bartelt, et al., 2000b). An ihm sind Anbieter und Nachfrager, gegebenenfalls auch Mittler beteiligt; Waren oder Dienstleistungen werden für Gegenwerte, in der Regel Geld, ausgetauscht.

Ein weiterer Teilbereich des Electronic Business ist die *Electronic Cooperation* (ECoop); die elektronisch unterstützte *Kooperation* von Unternehmen sowohl auf organisatorischer als auch auf technischer Ebene.

Diese Schwerpunktbereiche stellen jeweils ein unterschiedliches Kriterienspektrum in den Vordergrund, so dass es durchaus zu Überschneidungen kommen kann. Je nach Sichtweise auf die Evidenz der einzelnen

Kriterien sind dann spezielle E-Business-Aktivitäten mehr dem einen oder anderen Bereich zuzuordnen oder es sind noch weitere Bereiche zu definieren. Eine ähnliche wie die hier vorgestellte Einteilung findet sich auch bei Zwißler (Zwißler, 2002), die jedoch vereinfachend von Funktionsbereichen spricht. Zusätzlich wird von Zwißler der Bereich *Electronic Finance* eingeführt, der in dieser Arbeit weitgehend in die Electronic Information und den Electronic Commerce integriert wurde. Im Bereich Electronic Cooperation fokussiert sie die ganz andere Ebene von Infrastrukturaspekten des Business Messaging und geht nicht auf eigentliche Kooperationsprinzipien und Theorien der Electronic Cooperation, wie beispielsweise virtuelle Unternehmen (Sieber, 1998), ein.

Zur detaillierten Klassifikation bietet sich eine reichhaltige Auswahl an weiteren Kriterien an. Beispielhaft seien hier genannt:

- nach den beteiligten wirtschaftlichen und gesellschaftlichen Gruppen (*Anwendergruppen*) und ihren jeweiligen Konstellationen:
Anwendergruppen können Endkunden (Customer), der Geschäftsbereich (Business), Mitarbeiter eines Unternehmens (Employees), die öffentliche Verwaltung (Administration), Regierung (Government), Bürger (Citizens) usw. sein. Die typischsten Ausprägungen von Konstellationen sind B2B (Business-to-Business) und B2C (Business-to-Consumer).
- nach den beteiligten Wirtschaftssubjekten in ihren Rollen (*Wirtschaftssubjekte*):
Welche Wirtschaftssubjekte, wie Anbieter, Nachfrager oder Mediatoren sind beteiligt?
- nach aktivem oder passivem Verhalten der beteiligten Akteure (*Initiative*):
Dieses Kriterium bezieht sich darauf, ob die Akteure initiativ auf die anderen Akteure zugehen.
- nach der Art der gehandelten Güter:
Physische Produkte oder immaterielle Dienstleistungen, physische oder digitale Güter, Variantenprodukte, Einzelprodukte, etc.
- nach der Art der unterstützten Wirtschaftstätigkeiten (*Wirtschaftstätigkeiten*):
Hier kann detailliert nach Produktion, Dienstleistung oder verschiedenen Branchen usw. klassifiziert werden
- nach den unterstützten Phasen einer Handels- bzw. Markttransaktion (*Phasen*):
Hier sind im Wesentlichen die Handelsphasen der Bedarfsidentifikation, der Vermittlung von Produkten, der Vermittlung von Händlern, der Vereinbarung inkl. Verhandlung, der Abwicklung und des Services und Evaluation angesprochen, die ähnlich von

Guttman et. al. (Guttman, et al., 1998) vorgestellt und vom Autor in (Bartelt, et al., 2000b) zur Klassifikation verwendet werden (siehe auch Abschnitt 2.1.3.4).

- nach den verwendeten Technologien (*Technologien*):

Auch dieses Kriterium kann noch weiter unterteilt werden, z.B. nach verwendeter Middleware, Programmiersprachen, Betriebssystemen oder Kommunikationsplattformen, wie Internet, VAN, usw.

Eine Klassifikation nach dem Kriterium des aktiven oder passiven Verhaltens der beteiligten Akteure (*Initiative*) und nach dem Kriterium der beteiligten Wirtschaftssubjekte wird in Abbildung 2.1 veranschaulicht.

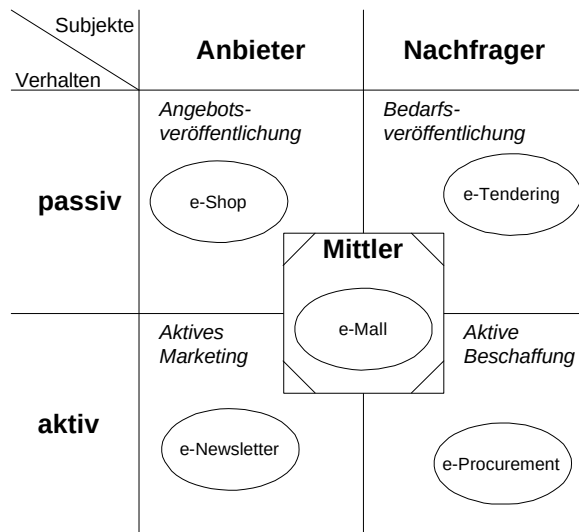


Abbildung 2.1: Klassifikation von Geschäftsmodellen nach Wirtschaftssubjekten und Verhalten der Akteure

Die erste Gruppe der von Anbietern getragenen Geschäftsmodelle ist die der *Angebotsveröffentlichung*. Bei diesen Modellen verhalten sich die Anbieter weitgehend passiv gegenüber den Nachfragern, veröffentlichen ihre Angebote und warten auf Anfragen und Bestellungen. Der Angebotsveröffentlichung kann insbesondere das Geschäftsmodell des E-Shops zugeordnet werden.

Bei den Geschäftsmodellen des *aktiven Marketings* treten die Anbieter aktiv an die Nachfrager heran. Sie stellen ihnen Informationen über Produkte, Preise oder andere Angebote zur Verfügung. Diese Art der Werbung erfolgt oft über E-Mail. Die negativen Auswüchse unerwünschter Zusendungen sind hinreichend als Spamming-Problem bekannt. Es gibt jedoch auch viele positive Anwendungen, wie einen aktiven Cross-

Selling Mechanismus oder die vom Kunden erwünschte Benachrichtigung über neue Preise. Der E-Newsletter ist ein typisches Beispiel des aktiven Marketings.

Bei den Geschäftsmodellen der *Bedarfsveröffentlichung* handelt es sich um von Nachfragern getragene Modelle, bei denen die Nachfrager ihre Bedarfe veröffentlichen und Angebote der Anbieter entgegennehmen. Durch die bisher relativ hohen Kosten dieses Vorgehens waren die Modelle bei geringer Marktmacht nicht praktikabel. Jedoch sind solche Geschäftsmodelle aufgrund der im Electronic Business durchschlagenden Kostenersparnisse inzwischen Erfolg versprechend. Für die organisatorische Bündelung der Nachfrager kann es sinnvoll sein, eine Mittlerinstanz einzusetzen und zu nutzen. Die grundlegende Idee beruht jedoch auf der Bedarfsveröffentlichung durch die Nachfrager. Ein Beispiel für ein Geschäftsmodell der Bedarfsveröffentlichung ist das E-Tendering.

Bei der *aktiven Beschaffung* recherchieren die Nachfrager und nehmen eine aktive Kommunikation mit den Anbietern auf. Diese Modelle werden von den Nachfragern getragen. Wichtige Beispiele sind E-Procurement und Shopping Agents.

Die Geschäftsmodelle für *Mittler* sind derzeit nicht eindeutig definiert. Die meisten der schon vorgestellten Geschäftsmodelle lassen sich mit vielen Ausprägungsmöglichkeiten auch durch Mediäre betreiben. Mit den Anfängen des Electronic Commerce entwickelte sich auch verstärkt das Konzept der Disintermediation, das die Kontaktherstellung zwischen Anbietern und Nachfragern ohne Mittler beinhaltet. Der Prozess der Zusammenführung kann jedoch über Mittler zum Teil immer noch kostengünstiger durchgeführt werden als ein direkter Suchprozess. Auch können dem Prozess Mehrwerte hinzugefügt werden. Für Mittler existieren verschiedene Modelle wie 3rd-Party-Marketplace, OTC-Handelssysteme oder Online-Börsen. Neben Marktplätzen sind die E-Mall und die E-Auction weitgehend als Mittler etabliert.

Diese Klassifikation wird vom Autor in (Bartelt, et al., 2001b) näher ausgeführt.

2.1.3 E-Business-Modelle

Konkrete Aktivitäten im Electronic Business lassen sich durch Geschäftsmodelle, wie Timmers sie für elektronische Märkte schon 1998 vorstellte (Timmers, 1998), charakterisieren. Dabei wird ein Geschäftsmodell als eine Architektur beschrieben, die sich aus Produkten, Dienstleistungen, Informationsflüssen und einer Beschreibung der beteiligten Akteure und ihrer Rollen zusammensetzt. Dazu kommt eine Be-

schreibung der möglichen Vorteile der Akteure und die Benennung der Einkunftsquellen des Geschäftsmodells (siehe auch Abschnitt 2.1.3.1.1).

In diesem Abschnitt sollen typische Modelle des E-Business anhand der von Timmers vorgeschlagenen und durch den Autor (Bartelt, et al., 2000b) erweiterten Methodik vorgestellt werden. Besonderes Augenmerk liegt auf den für die informationstechnische Umsetzung relevanten Faktoren. Eine weitere Vertiefung des Themas E-Business-Modellierung folgt in Abschnitt 2.2.

Zunächst werden allgemeine Modellierungsaspekte betrachtet und danach die Bereiche *Electronic Information* und *Electronic Commerce* vorgestellt.

2.1.3.1 Allgemeine Modellierungsaspekte

Es gibt einige Elemente der Modellierung von Geschäftsmodellen im Electronic Business, die für alle Teilbereiche zutreffen. Dazu zählen die Erlösmodelle und auch die Sichtweise der Funktionsmodule für E-Business-Modelle.

2.1.3.1.1 Erlösmodelle

Die Erlösquellen sind ein integraler Bestandteil eines Geschäftsmodells im E-Business (vgl. (Timmers, 2000)) und werden vom Autor ausführlich in (Bartelt, et al., 2001c) diskutiert. Sie bestimmen die direkten Einkünfte und bilden die Basis der Wirtschaftlichkeit eines Geschäftsmodells.

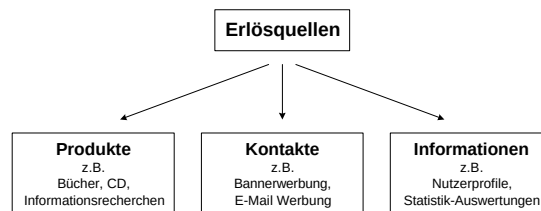


Abbildung 2.2: Erlösquellen, vgl. (Skiera, 2000)

Erlöse lassen sich im Internet nicht nur über den direkten *Verkauf* von Produkten und Dienstleistungen erzielen, sondern es gibt weitere Erlösquellen, wie die *Kontakt-* und die *Informationsvermarktung* (vgl. (Skiera, 2000), siehe Abbildung 2.2). Die direkte Erlösquelle über den Verkauf der Produkte ist in manchen Bereichen des Electronic Business nur schwer zu realisieren, da aufgrund geringer Produktwerte oder starker Konkurrenz oft nur ein geringer Marktpreis durchzusetzen ist (Suchmaschinen, Zeitversetzte Börsenkurse, etc.). Das bis heute weitgehend unge löste Problem praktikabler Micro-Payments im Internet macht in der Folge die direkte Erlösquelle über den Verkauf praktisch unmöglich. So

wird zu anderen Erlösquellen wie der Kontaktvermarktung gegriffen. Eine Möglichkeit ist das Schalten von Bannerwerbung. Die Akteure im Internet können also die *Kontakte*, die sie zu ihren Kunden haben, direkt nutzen. Des Weiteren entstehen durch die Besucher von Web-Seiten *Informationen*, wie beispielsweise Nutzerprofile, die ebenfalls als Erlösquelle genutzt werden können. Die Erfassung und Verwendung ist natürlich nur im Rahmen der datenschutzrechtlichen Vorgaben und insbesondere der Einwilligung der betreffenden Personen möglich. Die vorgestellten Quellen für Erlöse können auf unterschiedlichste Art und Weise genutzt werden. Einige Beispiele sind Lizenzgebühren, Pay-per-View Transaktionsgebühren oder vergütete Serviceleistungen. Die Erlösquellen und -formen lassen sich auf fast alle Geschäftsmodelle anwenden. Der tatsächliche Einsatz und die Erlösverteilung unter den Akteuren hängen jedoch vom konkreten Geschäftsmodell ab und sind eng mit der Preisbildung für das eigene Angebot und einem Angebot-Nachfrage-Equilibrium verknüpft.

Ein Problemfeld bei der Erlösrealisierung speziell im Bereich der Electronic Information ist die zum Teil weit verbreitete gesellschaftliche Ansicht, dass Informationen frei verfügbar sein müssten. Besonders ausgeprägt ist dies im Bereich wissenschaftlicher Informationen zu beobachten, wo jedoch auch oft eine entsprechende öffentliche Teil-Finanzierung bei der Produktion und Verteilung wissenschaftlicher Informationen zur Verfügung steht und diesen Anspruch stützt.

2.1.3.1.2 Funktionsmodule für E-Business-Modelle

In den Geschäftsmodellen des Electronic Business lassen sich Funktionsmodule identifizieren, die in anderen Modellen wieder verwendet werden können. Sie sind weiterhin zum Teil durch Module, die dem gleichen Typ entsprechen, austauschbar. Beispielsweise kann das Modul der Preisbildung durch statische Preise, ein Rabattsystem oder gar einen Auktionsmechanismus gestaltet werden. Durch Austausch und auch durch Kombination solcher Funktionsmodule werden neue Geschäftsmodelle geschaffen. Die softwaretechnische Realisierung dieser Funktionsmodule erfolgt gegebenenfalls durch Komponenten. Dieser Vorgang soll kurz am Beispiel eines E-Catalogs / E-Shops erläutert werden. Ein E-Catalog (E-Katalog) ist eine reduzierte Form eines E-Shops, der sich auf die Funktionen der Verkaufsförderung beschränkt. Mögliche Module eines E-Katalogs sind die Katalogfunktion, Produktpräsentation, Suchfunktion und Lieferbarkeitsauskunft. Ein E-Shop wird zusätzlich zum Geschäftsmodell des E-Katalogs weitere spezielle Module enthalten, zum Beispiel einen Warenkorb, eine Zahlungsabwicklung, eine Logistik-Verfolgung oder verschiedene Sicherheitsfunktionen (z.B. Verschlüsselung). Module

können verpflichtend enthalten sein, um ein spezielles Geschäftsmodell auszubilden, oder sie ergänzen ein Modell nur optional.

2.1.3.2 Charakteristika der Electronic Information

Die *Electronic Information* (EI), der elektronisch unterstützte Austausch von Informationen, bezieht sich auf alle Vorgänge im Electronic Business, die primär der Informationsvermittlung dienen (Bartelt, et al., 2001c). Wichtige Beispiele aus diesem intensiv bearbeiteten Bereich sind Digitale Bibliotheken und die inzwischen breit diversifizierten Web-Portale.

“It’s not a matter of giving people more information, but how to package that information, how to organize and disseminate it.”

Borgman 1999

Christine L. Borgman (Borgman, 2000; Borgman, 2001), die sich seit Jahrzehnten mit elektronischen Informationen, Digitalen Bibliotheken und der Informationsflut des Internets auseinandersetzt, stellt die Probleme bei der Vermittlung von Informationen an die Benutzer in den Mittelpunkt. Angesichts des Übergangs vom Dienstleistungszeitalter in das Informationszeitalter ist das Erfassen von Geschäftsmodellen, Nutzungskonzepten und internetbasierten Technologien des Electronic Business, die Informationen für die Benutzer bereitstellen, besonders wichtig.

Durch die *Geschäftsmodelle* der *E-Information* werden Geschäftstätigkeiten im informationszentrierten Bereich des E-Business konzeptuell beschrieben. Die Electronic Information benötigt jedoch auch Infrastrukturen und somit *Infrastruktur-Dienste* wie elektronische Zahlungssysteme und Sicherheitsmechanismen. Des Weiteren sind konkrete *Technologien* zur Umsetzung einzelner Geschäftsmodelle notwendig, wie allgemein im E-Business (siehe dazu auch Abschnitt 2.1.4).

Wie oben schon angesprochen, soll die von Timmers (Timmers, 1998) vorgestellte Modellierung als Ausgangspunkt für die weiteren Betrachtungen dienen.

Die Geschäftsmodelle der Electronic Information gehören zur Informationsökonomie und beziehen sich immer auf Informations-Produkte oder Informations-Dienstleistungen. Zur Modellierung können daher die Charakteristika der Electronic Information genutzt werden, wie beispielsweise der grundlegende Publikationsprozess. In der herkömmlichen Form sind daran als Akteure *Autoren*, *Verlage* und *Editoren*, in der physischen Produktion die *Setzerei*, *Druckerei* und *Binderei*, in der Vertriebskette die *Großhändler* und *Buchhandlungen* und letztendlich die *Leser* als Konsumenten beteiligt. Bei elektronischen Publikationen ist dieser Prozess fle-

xibler geworden. Die bisherigen Akteure sind veränderten Anforderungen unterworfen und es kommen neue Formen von Akteuren wie internet-basierte Content Provider hinzu. Diese Veränderungen ziehen sich durch den gesamten Publikationsprozess von der Informationserzeugung bis zur Informationsnutzung. Sie sind jedoch im Rahmen von Disintermediationseffekten im mittleren Teil des Publikationsprozesses besonders stark ausgeprägt. Aus Sicht der Benutzer können so neue nützliche Dienstleistungen und Geschäftsmodelle entstehen. Um diese zu konkretisieren und die Erstellung entsprechender Softwarelösungen zu fördern, werden adäquate Modellierungsmethoden benötigt, die jedoch derzeit nur in geringem Maße vorhanden sind.

Für den Bereich *Electronic Information* wird nun ein Phasenmodell vorgestellt, das Informationsobjekte in den Mittelpunkt stellt. Dabei werden die besonderen Anforderungen, die bei der Nutzung von Informationsobjekten auftreten, durch spezifische Phasen berücksichtigt.

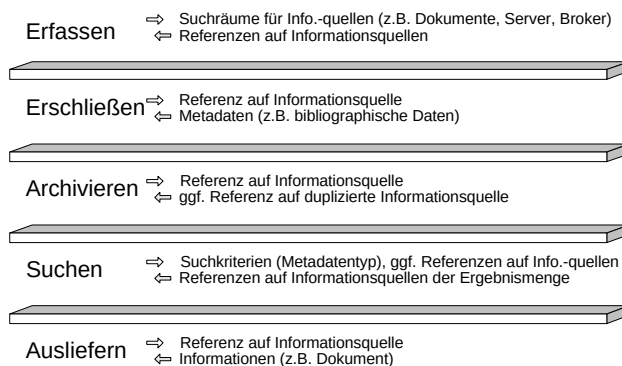


Abbildung 2.3: Informationszentriertes Phasenmodell

Beim *Erfassen* bzw. Sammeln von elektronischen Informationen werden die Informationsquellen zusammengetragen. Durch die *Erschließung* wird der Inhalt und Gehalt der einzelnen Informationsquellen durch Metadaten beschrieben. Die *Archivierung* dient dazu, einen späteren Zugriff auf die Informationen zu garantieren; gegebenenfalls durch das Anlegen einer persistenten Kopie. Bei der Durchführung des *Suchprozesses* wird entweder mit bestimmten Kriterien nach Informationen gesucht oder es erfolgt ein ‚Stöbern‘ (vgl. ‚Information Discovery‘ (Arms, 2000)) nach nützlichen Informationen. In der Phase der *Auslieferung* werden die Informationen schließlich dem Nutzer geliefert und vollständig zugänglich gemacht. Eine weitere mögliche Phase wäre die der *Relevanz-Rückmeldung*. Da sie nur relativ selten eingesetzt wird, wird sie hier jedoch nicht aufgeführt.

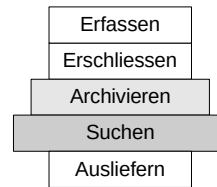


Abbildung 2.4: Beispiel EI-Phasenmodell

Wie intensiv ein Geschäftsmodell eine dieser Phasen abdeckt, wird in Grafiken (siehe Abbildung 2.4), die in der Folge für die Geschäftsmodelle aufgeführt werden, dadurch gekennzeichnet, wie breit und wie dunkel eingefärbt der Balken für die jeweilige Phase ist. Dies soll einen Anhaltspunkt geben, wie umfangreich und wie qualitativ hochwertig eine Phase in einem Geschäftsmodell behandelt wird. Der Durchlauf durch diese Phasen muss nicht immer unbedingt sequentiell erfolgen, es ist auch das Überspringen einer Phase möglich; beispielsweise wie der des Archivierens, wenn Informationen immer ad hoc erstellt werden. Gegenstand des informationszentrierten Phasenmodells sind nicht die Meta-Daten, die zur Erfüllung der Aufgaben eines Geschäftsmodells benötigt werden, sondern die Informationen, an denen der Nutzer interessiert ist.

2.1.3.3 E-Business-Modelle der Electronic Information

Die vielfältigen E-Business-Modelle der Electronic Information erfordern die Entwicklung weiterer Klassifikationen. Dazu ergänzend werden im Folgenden auch exemplarisch einzelne Referenzmodelle im Detail vorgestellt.

2.1.3.3.1 Klassifikation von Geschäftsmodellen der E-Information

Zur besseren Einordnung sollen verschiedene Kriterien vorgestellt werden, mit denen sich die *Geschäftsmodelle* der Electronic Information klassifizieren lassen. Tabelle 2.1 zeigt dazu mögliche Kriterien

Tabelle 2.1: Klassifikationskriterien für Geschäftsmodelle der Electronic Information

Domäne	Literatur
	Internet
	Geschäftliche Information
Informations-Phasen	Erfassen, ..., Ausliefern
Erlösquellen und -formen	Produkte
	Kontakte
	Informationen
Art	Nachrichten, Literatur, etc.

Inhalte	Informatik, Kunst, etc.	
Multimedia	Text, Audio, etc.	
Erzeugungszeit der Informationen	ad hoc	
	gespeichert	
Kontaktinitiative	PULL (aktiv)	
	PUSH (passiv)	vorher angefragt
		gespeichert

Aus diesen vielfältigen Klassifikationsmöglichkeiten, die noch weitergeführt und verfeinert werden können, werden zur Einordnung der Geschäftsmodelle vorwiegend die oben vorgestellten *Phasenmodelle* und eine *domänenorientierte* Einteilung gewählt. Letztere Einteilung dient zur Grobgliederung der Referenzmodelle der Electronic Information in drei Gruppen: Die Modelle in der Gruppe *Literatur* befassen sich mit dem Angebot von geprüften und anerkannten Informationen. In der Gruppe *Internet* sind Modelle enthalten, die sich vorwiegend mit den frei verfügbaren Informationen im Internet befassen. Bei den *geschäftlichen Informationen* stehen Informationen zur Durchführung von Geschäftstätigkeiten im Mittelpunkt der Betrachtung.

Orientiert an einem *Referenzmodell* können Ausprägungen realer Geschäftsmodelle leichter erstellt werden. Sie können für die konkrete Umsetzung an vorgefundene Szenarien angepasst und bei Bedarf mit anderen Modellen kombiniert werden. Im Folgenden werden einige etablierte und zukunftssträchtige Referenzmodelle der Electronic Information vorgestellt.

2.1.3.3.2 Literaturbezogene Referenzmodelle

Bei den literaturbezogenen Referenzmodellen werden nur qualitativ hochwertige, geprüfte und anerkannte Informationen angeboten. Die Nutzer erhalten als Ergebnis der Phase des *Suchens und Stöberns* meist bibliographischen Referenzen und gegebenenfalls auch die Informationen selbst. Die Auslieferung erfolgt je nach Geschäftsmodell durch die physische Zustellung des Originals oder einer Kopie an den Nutzer, durch physischen Zugriff des Nutzers, oder durch Auslieferung in digitaler Form.

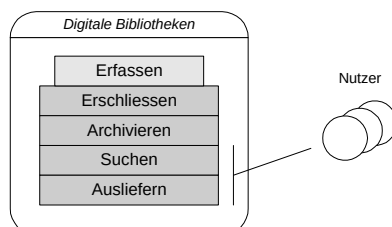


Abbildung 2.5: Digitale Bibliotheken

Ein wichtiges Referenzmodell sind *Digitale Bibliotheken* (Abbildung 2.5). Im Gegensatz zu konventionellen Bibliotheken halten digitale Bibliotheken als Volltextdatenbanken ihre gesammelten Informationen in elektronischer Form vor. Wurde vom Benutzer im Archiv der digitalen Bibliothek ein relevanter Eintrag gefunden, so kann die Publikation ohne Medienbruch direkt digital ausgeliefert und weiterverarbeitet werden. Bei den Realisierungen digitaler Bibliotheken sollten im Phasenmodell die meisten Phasen sehr gut abgedeckt werden. Wichtige Vorteile einer Digitalen Bibliothek sind der schnelle und flexible Zugriff auf Informationen, die guten Recherchemöglichkeiten mit mächtigen Anfragesprachen auf Volltexten, erweiterte Präsentationsmöglichkeiten und reduzierte Kosten (vgl. (Endres, et al., 2000)). Weitergehende Aspekte und spezielle Fragen der technischen Realisierung auch verteilter digitaler Bibliotheken werden ebenfalls vom Autor in (Bartelt, et al., 2001a) eingehender diskutiert. Ein weiteres neueres Werk zu Implementationsaspekten kommt von Witten und Bainbridge (Witten, et al., 2002). Ein bekanntes Beispiel für eine Digitale Bibliothek ist das Gutenberg Projekt (gutenberg.aol.de/), das seit 1994 Texte von mehr als 350 deutschen Klassikern sammelt. Bedeutende Digitale Bibliotheken im Bereich der Informatik wurden von der ACM (www.acm.org/dl/) und IEEE (computer.org) aufgebaut.

Kataloge sind Nachweisdatenbanken über die Bestände einer Bibliothek. Beispiel: Library of Congress (www.loc.gov/catalog/). *Meta-Kataloge* bieten als Föderationsdienst die Möglichkeit, auf mehreren Katalogen gleichzeitig zu recherchieren. Beispiel: Karlsruher Virtueller Katalog (www.ubka.uni-karlsruhe.de/kvk.html). Tabelle 2.2 nennt jeweils mit Beispielen weitere Modelle.

Tabelle 2.2: Weitere literaturbezogene Referenzmodelle mit Beispielen konkreter Ausprägungen

Verbundkataloge	www.gbv.de
Internet-Buchhandlungen	e-Shops (Bartelt, et al., 1999), eBooks
E-Journals	www.electronicmarkets.org
Tele- und Printmedien	www.tagesschau.de
Individualbücher und CD	www.buch.de , deshima.de
Books-On-Demand	www.bod.de
Writing-On-Demand	www.microserve.net/~sk/
Zitatnachweise	Citeseer.nj.nec.com
Hyper-Referenzen	www.crossref.org

2.1.3.3.3 Internetbezogene Referenzmodelle

In diesem Abschnitt werden Referenzmodelle vorgestellt, die sich vorwiegend mit den frei verfügbaren Informationen im Internet befassen. Bei diesen Internetbezogenen Referenzmodellen erhalten die Nutzer als Ergebnis der Phase des *Suchens und Stöberns* meist eine URL als Referenz auf die gesuchte Information.

Suchmaschinen dienen zum Auffinden vorher indizierter Web-Seiten durch meist sehr einfach formulierte Anfragen. Hingegen bedienen sich *Meta-Suchmaschinen* zur Durchführung der eigentlichen Suche anderer Suchmaschinen und bieten somit ebenfalls wie die schon vorgestellten Meta-Kataloge eine Föderationsfunktionalität an. *Web-Verzeichnisse* bzw. *Themenkataloge* sind Sammlungen von Verweisen auf Web-Sites, die zumeist manuell von Redakteuren gepflegt werden. Ein *Benachrichtigungsdienst* informiert seine Benutzer über die Veröffentlichung von Informationen, die zu dem vorher von ihm spezifizierten oder automatisch aufgrund seines Verhaltens ermitteltem Interessensprofil passen.

E-Portale (Abbildung 2.6) unterstützen den Nutzer beim Zugang zu den vielfältigen und unübersichtlichen Informationsangeboten im Internet, insbesondere durch ihre Möglichkeiten des Suchens und Stöberns.

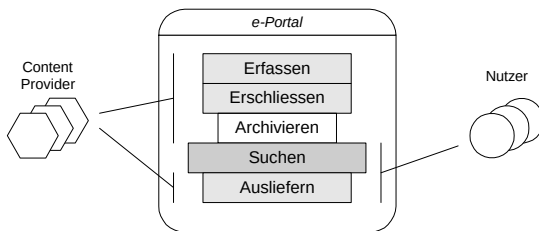


Abbildung 2.6: E-Portal

Im Kontext der Electronic Information kann ein Portal definiert werden als *eine feste Anlauf- und Rückkehrstelle im Web, die auf die persönlichen Informationsbelange des jeweiligen Nutzers ausgerichtet ist*. Der spezielle Lösungsansatz von Portalen bei der Bewältigung der Informationsvielfalt besteht dabei (1) in der Personalisierung ihrer Informationsangebote für den jeweiligen Nutzer, (2) in der Ausrichtung an bestimmten Nutzergruppen, wie privaten Nutzern, den Mitarbeitern einer Firma, oder an Lokalkriterien sowie (3) in der Schwerpunktbildung bei der Informationsauswahl (Expertise für einen bestimmten Bereich). Dieser Schwerpunkt kann sich in einer ‚horizontalen‘ Ebene, beispielsweise funktional und über viele Branchen hinweg, oder auf ‚vertikale‘ Weise eingeschränkt auf eine Branche ausbilden. Ein weiteres Element ist die reichhaltige Auswahl an vorbereiteten Informationen, aus denen der Nutzer auswählen kann.

Da Portale jedoch viele Charakteristika des Electronic Commerce aufweisen, werden sie weitergehend unter diesem Gesichtspunkt in Abschnitt 2.1.3.5.3 im Detail betrachtet.

2.1.3.3.4 Geschäftliche Referenzmodelle der E-Information

Die in diesem Abschnitt vorzustellenden Referenzmodelle wurden aus Geschäftsmodellen gebildet, bei denen im weiteren Sinne geschäftliche *Informationen* betrachtet werden. Unter der Sichtweise des Electronic Commerce sind sie zum Teil ebenfalls den Geschäftsmodellen des Electronic Commerce (siehe (Bartelt, et al., 2000b) und Abschnitt 2.1.3.5) zugeordnet. Tabelle 2.3 nennt in übersichtlicher Weise einige Modelle und Bereiche mit kurzen Anmerkungen.

Tabelle 2.3: Referenzmodelle geschäftlicher Informationen

E-Advertising		Werbung und Marketing als Informationsverbreitung; adXML
Angebotsvergleiche		Weiterentwicklung zu <i>Shopping Agents</i> [4]
E-Art		Kunst als Kommunikation und Information; herkömmlich / multimedial
Content-Provider		Beispiele unter www.iSyndicate.com
Intermediäre		als Händler für Vermittlungsinformationen
Finanzdienstleistungen	E-Banking	Home-Banking, HBCI, ...
	E-Finance	Finanzinformationen
	E-Brokering	Aktienhandel über das Internet
	E-Billing	Rechnungsausgleich online
Staat und Gesellschaft [1]	E-Administra.	Verwaltungsangelegenheiten
	E-Governme.	Regierungsaufgaben
	E-Democracy	Voting, Parteienarbeit, ...

2.1.3.4 Charakteristika des Electronic Commerce

Dem Bereich des Electronic Commerce kommt seit Beginn der kommerziellen Nutzung des Internets besondere Bedeutung zu, da dieser sich als erstes intensiv entwickelte und seitdem einen wesentlichen wirtschaftlich quantitativen Anteil am gesamten Electronic Business bildet.

Ein besonderes Charakteristika der Modelle des Electronic Commerce ist die Einteilung einer Handels- bzw. Markttransaktion in verschiedene Phasen, die sich auch als geeignet zur Analyse der elektronischen

Unterstützung von Geschäftsmodellen erwiesen hat (Bartelt, et al., 2000b).

Der Prozess der Bedarfsdeckung des Kunden lässt sich in verschiedene Phasen einteilen. Ein einfaches Grundmodell geht hier von den drei Phasen der Information, Verhandlung und Abwicklung aus (Merz, et al., 1999; Schmid, et al., 1995). Die hier verwendete detaillierte idealtypische Einteilung in sechs Phasen (siehe Tabelle 2.4) orientiert sich an dem Vorschlag von Guttman et al. (Guttman, et al., 1998).

Tabelle 2.4: Phasen der Bedarfsdeckung

Nr.	Phase	Abk.
1.	Bedarfsidentifikation	BI
2.	Produktvermittlung	PV
3.	Händlervermittlung	HV
4.	Verhandlung	Vh
5.	Kauf und Lieferung	KL
6.	Produkt-Service und Evaluation	SE

Die Unterteilung dieser Phasen stellt wie jedes Modell eine Vereinfachung dar. Einzelne Phasen können sich im konkreten Fall überschneiden oder spezielle Formen annehmen. Für die Modellierung der Geschäftsmodelle des Electronic Commerce kann gezeigt werden, welche Phasen ein Modell typischerweise und in welcher Form unterstützt. Dadurch werden die durch das Geschäftsmodell wahrgenommenen Aufgaben verdeutlicht.

Zum besseren Verständnis sind im Folgenden die durch einen e-Shop unterstützten Phasen in einer Grafik (siehe Abbildung 2.7) dargestellt. Die Breite der einzelnen Blöcke steht für den Grad der Unterstützung, den ein Geschäftsmodell für eine bestimmte Phase erbringt. Eine abgestufte Grauschattierung ist zusätzlich hinterlegt. Dieser Grad reicht von nicht oder nur geringfügig unterstützten Phasen (hier BI, HV und SE) über im Wesentlichen abgedeckte Phasen (hier PV und Vh) bis hin zu den besonders intensiv unterstützten Phasen (hier KL).

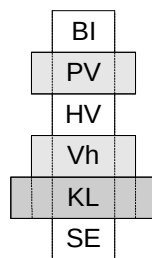


Abbildung 2.7: Beispiel für Phasenmodell einer Handelstransaktion

Das Phasenmodell kann neben der Förderung des Verständnisses eines Geschäftsmodells außerdem dazu beitragen, die richtigen Geschäftsmodelle aufgrund eines speziellen Bedarfs in einzelnen Phasen auszuwählen oder Defizite im E-Business-Bereich des Unternehmens aufzudecken.

Eine ausführliche Charakterisierung von Geschäftsmodellen des Electronic Commerce anhand des Phasenmodells durch den Autor ist in (Bartelt, et al., 2001b) zu finden.

2.1.3.5 E-Business-Modelle des Electronic Commerce

Ausgehend von den E-Shops als Vorreiter haben sich im Laufe der letzten Jahre vielfältige E-Business-Modelle im Electronic Commerce herausgebildet. Ein aktuelles und bedeutsames Modell ist die elektronische Beschaffung von Gütern und Dienstleistungen durch Unternehmen, das so genannte E-Procurement. Da es in dieser Arbeit noch eine zentrale Rolle einnehmen wird, findet es später noch besondere Beachtung.

Im Folgenden werden wesentliche Geschäftsmodelle des Bereiches Electronic Commerce, die bereits eine breite wirtschaftliche Relevanz erreicht haben, charakterisiert, um zu einem Gesamtverständnis beizutragen. Die vorgestellten E-Business-Modelle sind:

- E-Shops
- E-Auctions
- E-Portals

2.1.3.5.1 E-Shops

Ein E-Shop (Synonym 'Online-Shop') hat zwei Hauptfunktionen für ein Unternehmen, das Waren und Dienstleistungen über das Internet vertreibt: Diese sind Verkaufsförderung (Marketing) und Bestellabwicklung.

Das Geschäftsmodell eine E-Catalogs kann als eine Vorstufe zum E-Shop angesehen werden (vgl. auch Abschnitt 2.1.3.1.2 zu Funktionsmodulen). Bei einem E-Katalog werden Informationen und Auskünfte über die angebotenen Güter dem Kunden möglichst komfortabel, möglicherweise personalisiert, zur Verfügung gestellt.

Bei einem E-Shop wird typischerweise die Handelsphase des Kaufs und der Logistik (KL) zusätzlich intensiv unterstützt (siehe Abbildung 2.8).

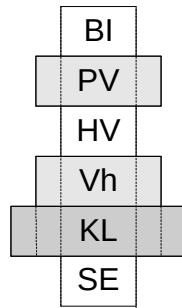


Abbildung 2.8: Phasenmodell für E-Shops

Abbildung 2.9 zeigt ein Szenario, in dem Online-Shops eingesetzt werden und wichtige Fragen, die bei der Einführung von Online-Shops zu beachten sind.

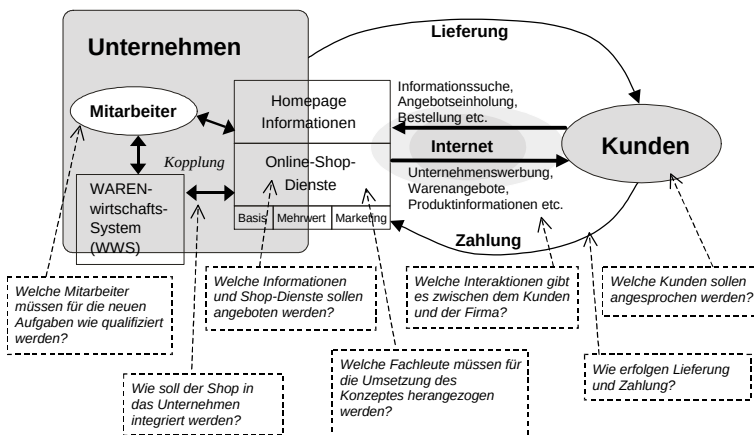


Abbildung 2.9: Online-Shop Szenario

Ein Online-Shop kann vielfältige Elemente enthalten, die für die Kunden verschiedene Arten von Dienstleistungen (Services) erbringen. Diese lassen sich den verschiedenen Dienstklassen 'Basis-Dienste', 'Mehrwert-Dienste' und 'Marketing-Dienste' zuordnen.

Basis-Dienste sind Funktionen, die für die Funktion des Online Shops unbedingt erforderlich sind. Sie gliedern sich in *primäre* und *sekundäre Basis-Dienste*. Die *primären Basis-Dienste* ergeben sich aus den Dienstleistungen und Produkten, die ein Unternehmen in seinem Kerngeschäftsfeld anbietet. Sie befassen sich typischerweise mit den Stufen der Information, der Bestellungsabwicklung, des Versands und der Zahlungsabwicklung. Zum Beispiel könnte ein Einzelhändler, der Musik CD vertreibt, die Kunden über die verfügbaren CD informieren und ihre Bestellungen entgegennehmen. Die *sekundären Basis-Dienste* sind weite-

re unabdingbare Dienste eines Online-Shops, wie die wichtigen Sicherheitsfunktionalitäten¹ oder die Verwaltung einer Benutzersitzung.

Mehrwert-Dienste werden erst durch den Online-Shop ermöglicht und dienen direkt den Basis-Diensten. Sie bieten dem Kunden neue zusätzliche Funktionalitäten an, beispielsweise eine schnelle, komfortable Suchfunktion. Die folgende Auflistung enthält Vorschläge für mögliche Mehrwert-Dienste:

- Suchfunktion,
- Mitglieder-/ Kundenverwaltung,
- One-to-One Produktlisten / Favoriten,
- Lieferbarkeitsauskunft,
- Bestellverfolgung (Order Tracking),
- Lieferungsverfolgung (Logistic Tracking) etc.

Die *Marketing-Dienste* fördern den ökonomischen Erfolg des Online-Shops. Sie unterscheiden sich von den Mehrwert-Diensten dadurch, dass sie *nicht* direkt den Basis-Diensten dienen. Beispiele hierfür sind:

- zusätzliche Hintergrundinformationen für Kunden,
- Sonderrabatte für Online-Shop-Bestellungen,
- allgemeine Community-Funktionen
- Preisausschreiben zur erhöhten Kundenbindung etc.

Ein typisches Beispiel für einen E-Shop ist bei der Firma JPC (www.jpc.de) zu finden. Es handelt sich um das klassische Verkaufsfeld von CD, Videos und Büchern, das sich gut für den Versandhandel eignet und so die Logistik erleichtert.

Für die technische Realisierung von Online-Shop-Systemen stehen inzwischen Standardsoftwarelösungen zur Verfügung. Welche Strukturen und Architekturen diese Systeme haben, wurden vom Autor und unter Anleitung des Autors untersucht und weiterentwickelt (Bartelt, et al., 2000c; Widhani, 2002).

Weiterführende Aspekte sowie das Geschäftsmodell eines Online-Shops werden vom Autor auch noch ausführlich an anderer Stelle beschrieben (Bartelt, et al., 1999; Bartelt, et al., 2000c; Widhani, et al., 2002).

2.1.3.5.2 E-Auctions

Die typische Ausprägung einer E-Auction (Synonym ‚Online-Auktion‘) kann als spezieller E-Shop angesehen werden, der die besondere Charakteristik der dynamischen Preisbildung über Auktionen besitzt.

Online-Auktionen stellen eine elektronische Umsetzung von robusten Marktmechanismen dar, wie sie aus traditionellen Auktionen bekannt

¹ Beispielsweise kann durch die Nutzung des [https](https://)-Protokolls eine sichere Verbindung aufgebaut werden. Dies ist für das Vertrauen der Kunden in den Online-Shop wichtig.

sind. Eine Auktion wird oft als Marktform gewählt, wenn die zu verkaufenden Artikel in eine der folgenden beiden Kategorien fallen: (1) Artikel, deren Preise sich schnell und extrem ändern. Grund dafür kann die Abhängigkeit von Verfügbarkeitszeitpunkt und -menge (z.B. Fisch oder Blumen), die unterschiedliche Wertschätzung der Kunden (z.B. bei Kunstwerken) oder die Inhomogenität des Artikels (z.B. bei Vieh) sein. (2) Artikel, für die es keinen Referenzpreis gibt (z.B. Erdgas-, Erdöl- und Mineralienabbaurechte).

Bei Auktionen treffen die Kunden und Verkäufer aufeinander und versuchen nach fest vorgeschriebenen Regeln und Protokollen einen optimalen Preis für Artikel durch Bieten festzulegen. Der Grad der elektronischen Unterstützung kann dabei von einer elektronischen Gebotsabgabe bis zur vollständig automatisierten Durchführung einer Auktion reichen. In herkömmlichen Systemen beschränkt sich jedoch die elektronische Unterstützung auf ein zentrales System, auf das die Benutzer durch die Abgabe ihrer Gebote zugreifen können. Typische unterstützte Auktionsarten sind beispielsweise 'English', 'Dutch' oder 'Vickrey'. Reck (Reck, 1997) stellt eine ausführliche Taxonomie für Auktionsarten vor. Reduziert man eine E-Auction auf die Preisbildung, ist dieses Funktionsmodul wiederum gut mit vielen anderen Geschäftsmodellen kombinierbar. Aufgrund des oftmals spannenden Ablaufs wecken Online-Auktionen im B2C Bereich großes Interesse. Groß angelegte Auktionssysteme im C2C-Bereich sind in den Bereich der Mittler im Electronic Commerce einzuordnen, da hier viele Anbieter ihre Waren über das System des Mittlers in Auktionen versteigern lassen. Ein typisches Beispiel dafür ist eBay (www.eBay.de). Im B2B-Bereich werden Auktionen auch als Form des Dynamic-Pricing und zum Aufbau von Marktplätzen eingesetzt (Beispiel: Fastparts, www.fastparts.com).

Abbildung 2.10 zeigt das Phasenmodell für E-Auctions, das insbesondere die Eigenschaft der dynamischen Preisbildung in Form der Verhandlungsphase hervorhebt.

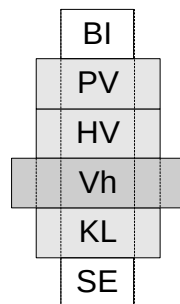


Abbildung 2.10: Phasenmodell für E-Auctions

Auktionen werden nicht nur in den Bereichen der Mittler oder Angebotsveröffentlichung, sondern auch im Bereich der Bedarfsveröffentlichung verwendet. Ein Beispiel ist das E-Tendering.

Konventionelle Ausschreibungen sind nur sinnvoll für Nachfrager, die ein großes Auftragsvolumen in Menge oder Wert nachfragen. Bei der eminenten Senkung der einzelnen Transaktionskosten durch die neuen Technologien des Electronic Commerce wird nun das Verfahren des E-Tendering (der Online-Ausschreibung) auch für kleine Auftragsvolumina sinnvoll. Ein Nachfrager ‚versteigert‘ dabei seinen Auftrag unter mehreren Anbietern. Damit sich diese allein auf Nachfrager gestützten Systeme durchsetzen, sind noch weitere Entwicklungen einer entsprechenden Infrastruktur nötig. Auf Mittler gestützte Beispiele dieser Idee sind der Deal-Assistant (www.Dealassist.com) und eWanted (www.eWanted.com), die Bedarfsveröffentlichung mit umgekehrten Auktionen im C2C-Bereich betreiben. Die Händlervermittlung (HV) und Verhandlungsphase (Vh) werden durch E-Tendering besonders intensiv unterstützt (siehe Abbildung 2.11).

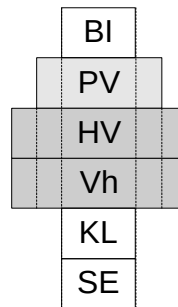


Abbildung 2.11: Phasenmodell für E-Tendering

Es gibt inzwischen etliche Standardsoftwarelösungen für Online-Auktionen. Ein Beispiel ist die Software von Open-Site (www.opensite.com), einen Unternehmen, das inzwischen von Siebel aufgekauft wurde.

Wissenschaftlich wurde das Thema bisher in mehreren bedeutenden Projekten betrachtet:

- Michigan Internet AuctionBot²
- eAuctionHouse³
- Electronic Fishmarket Auction⁴, agentenorientiert

Unter Betreuung des Autors entstand ein wissenschaftlicher Prototyp für ein Online-Auktionssystem mit dem Namen OAsys (Dittrich, 2000),

² <http://auction.eecs.umich.edu>

³ <http://ecommerce.cs.wustl.edu/emediator>

⁴ <http://www.iiia.csic.es/Projects/fishmarket>

bei dem insbesondere auch die Interoperabilität mit anderen E-Commerce-Systemen Beachtung fand.

2.1.3.5.3 E-Portals im Electronic Commerce

E-Portals wurden in Abschnitt 2.1.3.3 bereits unter dem Gesichtspunkt der Electronic Information und mit Fokus auf die Verarbeitung und das Angebot von Informationen kurz vorgestellt.

Im Kontext des Electronic Commerce kann ein E-Portal definiert werden als eine feste Anlauf- und Rückkehrstelle im Web, die auf die persönlichen Informationsbelange des jeweiligen Nutzers im Hinblick auf Informationen zu Produkten und Handelspartnern ausgerichtet ist. Wie auch in der Electronic Information besteht der spezielle Lösungsansatz von Portalen bei der Bewältigung der Informationsvielfalt (1) in der Personalisierung ihrer Informationsangebote für den jeweiligen Nutzer, (2) in der Ausrichtung an bestimmten Nutzergruppen sowie (3) in der Schwerpunktbildung bei der Informationsauswahl (Expertise für einen bestimmten Bereich).

In Abbildung 2.12 wird deutlich, dass die E-Portals des Electronic Commerce einfachen Zugang zu Handelspartnern, Business- und Produkten anbieten. Siehe dazu auch das Phasenmodell für E-Portals in Abbildung 2.12. Sie stellen so eine Vorstufe zu Marktplätzen dar. Marktplätze ermöglichen dann zusätzlich die Durchführung und Abwicklung einer Handelstransaktion.

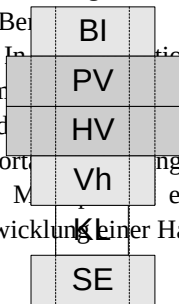


Abbildung 2.12: Phasenmodell für E-Portals

Spezielle Ausprägungen des Geschäftsmodells E-Portal sind: Consumer-Portale, Branchen-/Fach-Portale, Intranet-Portale und Enterprise-Information-Portale (EIP). Durch Portale können auch viele der schon vorgestellten Geschäftsmodelle der Electronic Information oder des Electronic Commerce in einem einzigen Portal integriert werden. Das E-Portal kann von der E-Mall (siehe Abschnitt 2.1.3.5.4) dahingehend abgegrenzt werden, dass ein E-Portal keine gemeinsamen E-Commerce Funktionalitäten in Bezug auf die Phase Kauf und Logistik beinhaltet.

Zum Konzept der E-Portals gehört die Präsentation von Inhalten und Funktionen in abgegrenzten dynamischen Web-Bereichen, den so genannten Portlets.

Die Realisierung von Portalen erfordert den Einsatz komplexer Web-Technologien und die Anbindung heterogener Informationsquellen. Verschiedene Anbieter, darunter Viador, Oracle und BEA, versuchen Plattformen und Standardsoftware für den Betrieb von Portalen zu schaffen.

Als Beispiel für ein E-Portal soll kurz auf das PublicationPORTAL eingegangen werden, das als Vermittler von Informationen und Diensten im Publikationsbereich dient.

Viele der im Bereich Electronic Information vorgestellten Referenzmodelle finden schon heute in der Electronic Information Verwendung. So entsteht eine Informations- und Dienstvielfalt, in der es den Benutzern an Übersicht mangelt.

Das PublicationPORTAL adressiert diese Problematik durch die Mediationsstärken des Geschäftsmodells E-Portal, bei dem andere Geschäftsmodelle als Web-Services eingebunden werden können, sowie durch Personalisierungsfunktionen und einem dreiphasigen Unterstützungsmodell: Orientierung und Vermittlung, integrierte Kooperation und Koordination und Zugangs- und Nutzungsunterstützung. Detaillierte Ausführungen zu dem im Rahmen eines GlobalInfo-Projektes des BMBF am Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS) der Universität Hamburg entstandenen Prototypen finden sich in (Bartelt, et al., 2001b; Offermann, 2001; Zirpins, et al., 2001). Abbildung 2.13 zeigt eine Architekturskizze der auf Basis des Open-Source-Frameworks Jetspeed der Apache Group entstandenen Implementation.

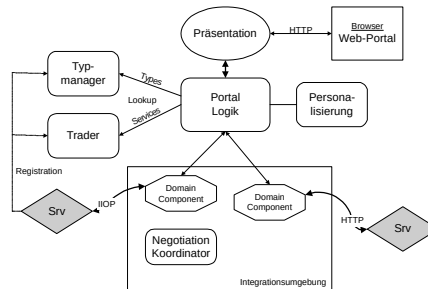


Abbildung 2.13: Architekturskizze PublicationPORTAL

2.1.3.5.4 Weitere Modelle des Electronic Commerce

Als interessante weitere Modelle des Electronic Commerce werden nun die E-Mall und Shopping-Agents vorgestellt. Erstere zeichnen sich als ein praktikables Modell für Mittler aus, während letztere einen innovativen Ausblick darstellen.

Eine typische *E-Mall* wird durch den Zusammenschluss mehrerer e-Shops gebildet. Die Shops können ein komplementäres Produktsortiment anbieten, beispielsweise für spezielle thematische Events. Es kann sich aber auch um horizontale oder vertikale Branchen-Malls handeln. Oft verfügt eine Mall über einheitliche Servicefunktionen wie Warenkorb-, Zahlungs- und Logistiksysteme. Die wesentlichen Vorteile einer Mall sind das ver-

größte Funktionsangebot, die einheitliche Benutzbarkeit des Systems sowie die gemeinsamen Servicefunktionen. Das Phasenmodell für E-Malls (Abbildung 2.14) verdeutlicht, dass insbesondere die Phase der Händlervermittlung, die Phase des Kaufs und der Logistik intensiv von einer E-Mall unterstützt werden.

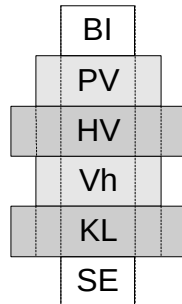


Abbildung 2.14: Phasenmodell für E-Malls

Shopping-Agents (Synonym ‚Kauf-Agenten‘) sind Software-Agenten (Weiss, 1999; Wooldridge, et al., 1995), die im Auftrag von Nachfragern nach Anbietern bestimmter Produkte und Dienstleistungen suchen und dabei Preise, Konditionen und ähnliches beachten und gegebenenfalls selbständig und autonom verhandeln und kaufen. Dadurch können für die Nachfrager Zeit- und Kostenersparnisse sowie ein besserer Grad der Bedürfnisbefriedigung erreicht werden. Die Anbieter müssen dafür mit kooperativen E-Shops oder anderen EC-Systemen im Internet vertreten sein. Sie erhalten so die Chance, mehr Kunden zu gewinnen und einen höheren Umsatz zu erzielen. Insbesondere aus diesem Geschäftsmodell resultiert jedoch auch eine erhöhte Markttransparenz, welche die Konkurrenzsituation der Anbieter untereinander verstärkt. Hier ist es nötig, dass möglichst viele wettbewerbsrelevante Konditionen und nicht nur der Einzelpreis eines Produktes in die Verhandlungen einbezogen werden. Beispielsweise können die Lieferbedingungen für einen Kauf entscheidend sein. Eine angemessene Vollständigkeit der berücksichtigten Konditionen ist zu gewährleisten (Bartelt, et al., 2000a). Ein Beispiel für einen Kauf-Agenten ist Jango (www.jango.excite.com), der aus dem Netbot-Projekt (Doorenbos, et al., 1997) hervorging. Abbildung 2.15 macht die umfassenden Ansprüche deutlich, die an Shopping-Agents im Kernbereich des Ablaufs einer Handelstransaktion gestellt werden.

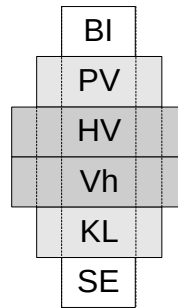


Abbildung 2.15: Phasenmodell für Shopping-Agents

Zum innovativen Forschungsbereich des Einsatzes von Agententechnologie im Electronic Commerce fanden schon früh Untersuchungen des Autors statt (Barbu, 1999). Die Forschung im Bereich des Einsatzes von Agententechnologie im E-Commerce ist durch die Gruppe AMEC (Agent Mediated E-Commerce) der EU-Initiative AgentLink (<http://www.-agentlink.org/>) und entsprechende wissenschaftliche Überblicke und Veröffentlichungen inzwischen gut dokumentiert.

Das oft als Schlagwort verwendete Geschäftsmodell des *E-Marketplace* kann auch als Vereinigungsmenge der Eigenschaften aller Geschäftsmodelle des Electronic Commerce gesehen werden. Ein E-Marketplace sollte alle Handelsphasen optimal unterstützen. Ein solches ‚Mega-Geschäftsmodell‘ ist jedoch in der praktischen Realisierung sehr aufwendig und anspruchsvoll. Es sollte nur eingesetzt werden, wenn auch tatsächlich entsprechende Anforderungen vorliegen.

2.1.3.6 Weitere Modelle und Bereiche des E-Business

Der Vielfalt und Kreativität auch im Detail sind kaum Grenzen gesetzt, so dass E-Business-Modelle auf unterschiedlichste Weise gestaltet werden können. Die Entwicklung neuer Modelle wird oft durch neue ökonomische Gegebenheiten und neuartige technologische Möglichkeiten angeregt.

Zweckbezogen sind ergänzend zu den oben genannten Bereichen auch weitere Bereichseinteilungen möglich. Auf den Bereich E-Finance wird im Folgenden kurz eingegangen.

2.1.3.6.1 Bereich E-Finance

Zwißler (Zwißler, 2002) grenzt den Bereich des E-Finance gesondert ab. Die Anwendungen beziehen sich darauf, einen Großteil des Routinegeschäfts online anzubieten. So führt die elektronische Verarbeitung zu hoher Aktualität und Interaktivität zu Gunsten des Kunden. Dabei stehen Kontoführung und Wertpapiergeschäft im Vordergrund. In jüngster Zeit

werden Finanzdienstleistungen auch von Nicht-Banken angeboten, beispielsweise von Einzelhandelskonzernen, die über virtuelle Konten eine höhere Kundenbindung anstreben, oder von Kapitalanlagegesellschaften, die über das elektronische Medium den Direktkontakt zum Kunden suchen.

Die Systeme für elektronische Finanzdienstleistungen werden weiterhin von der Bank als dem Dienstleister selbst betrieben. Besonderes Augenmerk finden in diesem Anwendungsbereich die Sicherheitsanforderungen sowohl im Kundenbereich als auch für die möglichst geschlossenen Systeme der Banken selbst.

E-Business-Modelle im E-Finance-Bereich sind beispielsweise Home-Banking, Internet-Banking oder Online-Brokerage.

2.1.4 Technologien für E-Business-Systeme

E-Business-Systeme stellen Anforderungen in verschiedenen Bereichen, die jeweils mit spezifischen Technologien abgedeckt werden müssen. Der technologische Entwicklungsstand ist dabei sehr unterschiedlich. Teilweise existieren einfache, gut entwickelte und breit akzeptierte Technologien wie http und HTML. Jedoch müssen einige Technologien, wie die Web-Services, noch weiterentwickelt werden, um die umfassenden Anforderungen besser abzudecken. Zum Teil sind ganz neue Technologien zu entwickeln, um unter anderem elektronische Verhandlungen gemäß den Ansprüchen abzudecken.

Zunächst soll ein kurzer Überblick über Anforderungen von E-Business-Systemen und mögliche Technologien gegeben werden.

Hier ist als erstes die *Anwendungslogik* zu nennen, die im Wesentlichen durch die Verwendung allgemeiner Programmiersprachen abgedeckt werden kann.

Die intensiv erforderliche *Datenhaltung* wird durch Datenbanken und Informationssysteme erleichtert.

Als *Kommunikationsplattform* wird im Hauptsächlichen heute das Internet mit seinen vielfältigen Möglichkeiten genutzt.

Eine *Integration von E-Business-Systemen* mit ERP- und Legacy-Systemen beispielsweise im Bereich der B2B-Integration erfordert entsprechende Adapter, die ebenfalls im Zuge von innerbetrieblicher EAI (Enterprise-Application-Integration) notwendig sind. Zur Herstellung von Interoperabilität sind auf den verschiedenen innewohnenden Schichten noch viele Lösungen zu schaffen. Wichtige unterstützende Technologien sind dabei XML, SOAP oder die Verwendung von Ontologien (Davies, et al., 2003).

Bezüglich der *Zahlungsabwicklung im Internet* sind schon vor einigen Jahren neue Technologien entwickelt worden, die sich jedoch nur zum Teil und kulturell abhängig durchgesetzt haben. In Deutschland sind derzeit konventionelle Verfahren wie das „Lastschriftverfahren“ beliebte Zahlungsmethoden. Selbst innovative und praktikable Verfahren wie beispielsweise die Zahlung nach Rückfrage über Handy, ‚Paybox‘ genannt, konnten sich bisher nicht durchsetzen.

Für den Vertragsabschluss sind ebenfalls Unterstützungssysteme zu entwickeln (Merz, et al., 1998) (Electronic Contracting).

Unter dem Aspekt der Verhandlungen sind Unterstützungs- und Automatisierungssysteme interessant.

Die Kunden- und die Verwaltungsschnittstelle erfordern den breiten Einsatz von Web-Technologien.

So nutzen E-Business-Systeme neben Technologien aus klassischen Bereichen, wie Datenbanken und spezielle Anwendungslogik mit integriertem ERP-Systeme, zu großen Teilen *Weborientierte Internettechnologien*. Hier geht es im Wesentlichen um die Entwicklung von Web-Anwendungen, die schon ausführlich in zahlreichen Werken (vgl. (Ceri, et al., 2003; Conallen, 2002; Merz, 1999; Merz, 2002; Zwißler, 2002), unter anderem auch zu Web-Entwurfsmustern (Fowler, 2003)) dargestellt wird. Beispiele für aktuelle Technologien sind:

- Programmiersprachen wie Java (Eberhart, et al., 2000) oder Skriptsprachen wie PHP
- HTML
- XML, XSLT (Langham, et al., 2002; Moczar, et al., 2003),
- Java Server Pages (JSP)
- Application Server, EJB, J2EE
- Sicherheits- und Vertrauensmechanismen
- Verteilte Programmierung (RPC, DCE, COM+, CORBA, RMI, JNDI, JMS, JDBC, ...)
- Web-Services (SOAP, UDDI, WSDL, WSFL...); (Ayala, et al., 2002; Schmelzer, et al., 2002; Snell, et al., 2002)

Im Bereich der Web-Anwendungen weisen E-Business-Systeme nur in geringem Maße Eigenschaften auf, die sie von anderen Systemen unterscheiden. Hier sind beispielsweise Module und Komponenten wie Warenkörbe oder ähnliches zu nennen, die spezifisch für die Anwendungsdomäne der E-Business-Systeme sind. Diese Aspekte werden auch vom Autor ausführlich in (Widhani, et al., 2002) diskutiert und ebenfalls in (Widhani, 2002) und (Böge, 2003) angesprochen.

Bei der Realisierung von kundenspezifischen E-Business-Systemen stehen verschiedene Ausgangspositionen und Möglichkeiten zur Verfügung.

Zunächst kann auf einem *Fundus von Basistechnologien*, wie Datenbanken, Web-Technologien, Programmiersprachen, ERP-Systemen usw., eine Applikation entwickelt werden. Es ist jedoch auch möglich *Komponenten, Frameworks und Klassenbibliotheken* zu verwenden. Oder man bedient sich so genannter *Standardsoftware*, wie sie beispielsweise von INTERSHOP, Broadvision oder SAP angeboten wird. Diese Standardsoftware muss jedoch oft noch *angepasst* werden, um wirklich ein kundenspezifisches System zu erhalten.

Die hier angesprochenen Technologien und ihre Anwendung sind verschieden weit entwickelt und etabliert. Die bestehenden Systemlösungen erfüllen jedoch oft noch nicht die Anforderungen der Anwender. Das EU-Projekt MEMO machte dies für den Bereich B2B EC deutlich (Quix, et al., 2002). Ein offensichtliches Defizit besteht in der Bereitstellung und Integration von Verhandlungssystemen im E-Business, für welches in dieser Arbeit Lösungsvorschläge entwickelt werden.

2.2 Modellierungsmethoden im Electronic Business

Modellierungsmethoden sind eine wichtige Grundlage für die Entwicklung und Analyse von E-Business-Lösungen und -Systemen.

Ausgangspunkt ist die Bildung von *Geschäftsmodellen im E-Business*. Es liegen derzeit nur wenige Arbeiten vor, die eine Formalisierung von Geschäftsmodellen im E-Business umsetzen. Timmers (Timmers, 1998) schuf hier 1998 erste Ansätze. Alt and Zimmermann (Alt, et al., 2001) heben in ihrer Einführung zur Special-Section „Business Models“ zum zehnjährigen Bestehen des Journals Electronic Markets, die Bedeutung eines grundlegenden Verständnisses von E-Business-Modellen hervor. An derselben Stelle werden spezialisierte Arbeiten von Essler et al., welche die Modellierung von Interaktivität in den Mittelpunkt rücken, und von O’Daniel, der ein Mehrwert-basiertes Modell anstrebt, vorgestellt.

Eine weitere wichtige und zum Teil weiter gefasste Arbeit zum Thema, im Folgenden intensiver dargestellt wird, kommt von Gordijn (Gordijn, 2002), der mit e³-Value ein auf Werte und den Austausch von Werten zentriertes Modell vorgeschlägt.

Außerdem wird auf *umfassende Modellierungsmethoden* für den Electronic Business, insbesondere auf RSD und E-BPMS, eingegangen.

Die *Geschäftsprozessmodellierung* nimmt für die Abläufe im E-Business eine besondere Rolle ein, da sowohl die internen Geschäftsprozesse der beteiligten Unternehmen berührt werden als auch die interorganisationel-

len Geschäftsprozesse zwischen den Unternehmen zentraler Bestandteil des E-Business sind.

2.2.1 Modellierung von Geschäftsmodellen und Softwarelösungen im E-Business

Im Laufe der letzten Jahre sind in Forschung und Praxis mehrere Ansätze zur E-Business-Modellierung vorgeschlagen worden (vgl. (Wriggers, 2003)). Hier sollen bedeutende und relevante Ansätze besprochen werden.

Zunächst wird der ‚Rational Unified Process‘ (*RUP*) vorgestellt, der trotz einer nur ansatzweisen Unterstützung einer E-Business-Modellierung eine gute Einordnung in den Softwareentwicklungsprozess gibt. Daran anschließend werden die spezifischeren Methoden *E-BPMS* und *RSD* näher beschrieben.

Die Methode *e³-Value* wird aufgrund ihrer besonderen Eignung zur Modellierung von Geschäftsmodellen in einem eigenen Abschnitt 2.2.2 dargestellt.

2.2.1.1 E-Business-Modellierung bei der Entwicklung von Softwarelösungen

Modellierungsmethoden finden bei der Entwicklung von Softwarelösungen weiterhin Anwendung. Dazu gehören auch Vorgehensmodelle wie der ‚Rational Unified Process‘ (*RUP*), der sich bei vielen objektorientierten Softwareprojekten etablieren konnte (<http://www.rational.com>) (Kruchten, 2000). Der *RUP* ist eng mit den Modellierungsmöglichkeiten der ‚Unified Modelling Language‘ (*UML*) verbunden.

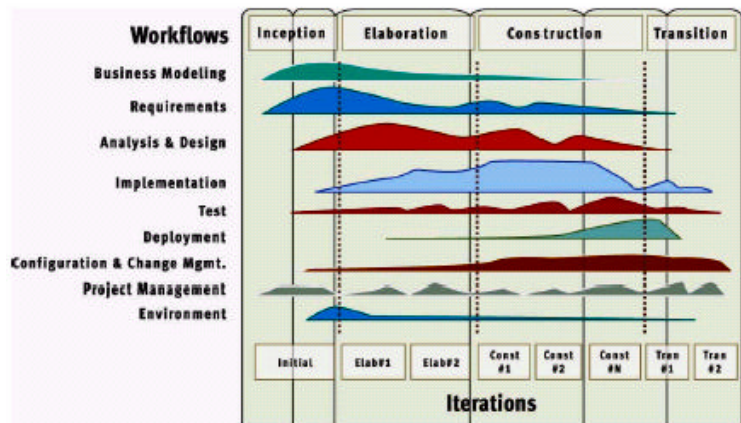


Abbildung 2.16: Phasen und Workflows beim RUP

Beim RUP werden vier Phasen (siehe auch Abbildung 2.16) unterschieden:

- *Inception*: Projektbeschreibung, Architektur, Planung, Anforderungsanalyse
- *Elaboration*: Prüfung von Architektur/Planung, detailliertes Design
- *Construction*: Implementierung, Einsatz der Ressourcen
- *Transition*: Auslieferung, Support

Durch die vier zeitlich geordneten Phasen ziehen sich die Workflows, die zur Umsetzung des jeweiligen Projekts erforderlich sind und die durch den RUP unterstützt werden. Sie werden parallel ausgeführt. Ihre iterative Wiederholung ist je nach Projektphase unterschiedlich intensiv. Abbildung 2.16 veranschaulicht dies deutlich. Der gesamte Ablauf wird auch Lebenszyklus genannt und hat als Ziel das fertige Softwaresystem.

Die Workflows werden jeweils durch ein UML-Aktivitätsdiagramm beschrieben, deren einzelne Aktivitäten in Dokumenten und UML-Diagrammen resultieren. Ein mögliches, wichtiges Dokument im RUP ist das Glossar, in dem die Fachbegriffe der betrachteten Domäne erfasst werden.

Der Workflow ‚Business Modelling‘ ist überwiegend in der einleitenden ‚Inception‘-Phase angesiedelt und dient nachfolgenden Workflows als Grundlage, wie Analysis & Design und Implementation.

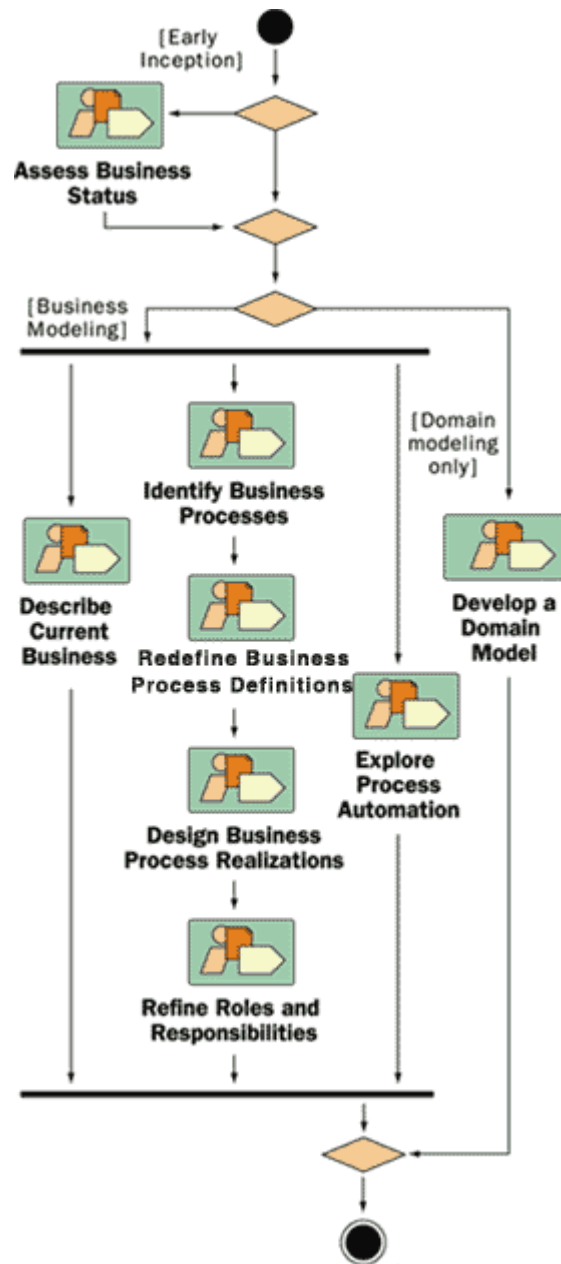


Abbildung 2.17: Business-Modelling-Workflow im RUP

Die folgende Auflistung zeigt wesentliche Aktivitäten des Workflows ‚Business-Modelling‘ (Abbildung 2.17):

- Beschreibung der Geschäftstätigkeit
- Identifizierung der Geschäftsprozesse (von dieser Aktivität werden unten beispielhaft die Details aufgeführt)

- Überarbeitung der Geschäftsprozesse
- Gestaltung der Umsetzung der Geschäftsprozesse
- Überarbeitung von Rollen und Verantwortlichkeiten
- Erkennen von Automatisierungsmöglichkeiten in den Geschäftsprozessen

Betrachtet man die Teilaktivität *‘Identify-Business-Processes’* näher, werden durch den RUP folgende Schritte festgelegt:

- ‘Maintain Business Rules’
- ‘Set and Adjust Goals’
- ‘Define Business Architecture’
- ‘Capture a Common Business Vocabulary’
- ‘Find Business Actors and Use Cases’

Die Anwendung der Aktivitätsschritte führt als Ergebnis zu den folgenden Dokumenten (‘Key Resulting Artifacts’):

- Business Rules
- Business Vision
- Business Architecture Document
- Business Glossary
- Business ‘Use-Case’ Model
- Business Use Case
- Supplementary Business Specification

Außer den durch UML gegebenen Möglichkeiten, hier speziell die ‘Use-Cases’, werden keine speziellen Diagramme zur Darstellung von Geschäftsarchitekturen verwendet, wie dies bei anderen Methoden der Fall ist.

UML-Diagramme haben sich als allgemeiner Modellierungsstandard etabliert haben. In ihrer Verwendung liegen besondere Vorteile:

- UML Tools sind bereits weit verbreitet.
- Modellierer können bekannte Notation nutzen und müssen sich nicht in spezielle Modellierungstechnik einarbeiten.
- UML wird von vielen Leuten verstanden, besonders ‘Use-Cases’ benutzen wenige Modellelemente; daher sind die Diagramme auch für Laien leicht verständlich.
- Attraktiv aus der Sicht von Softwareentwicklern, da das System-design nah ist.

Folgende Nachteile allerdings existieren bei der Business-Modellierung mit UML (vgl. (Xie, et al., 2001)):

- UML ist nicht für die Business-Modellierung entwickelt worden. Aufgrund der auf diesem Gebiet beschränkten Ausdrucksmöglichkeiten werden in ‘Use-Cases’ meist vergleichsweise einfache geschäftliche Beziehungen dargestellt. Es besteht sonst die Ge-

fahr, die Diagramme zu überladen. Es gibt Versuche UML mit eingebauten Mitteln (Stereotypes) bzw. durch andere ‚Extensions‘ zu erweitern

- Zwischen der Ebene der Analyse („Use-Cases“) und dem softwaretechnischen Design (Klassendiagramme) besteht eine Kluft. UML bietet kaum Unterstützung, um geschäftliche und softwaretechnische Aspekte miteinander in Verbindung zu bringen.
- Aus geschäftspolitischer Sicht ist die Darstellung der hierarchischen Struktur einer Organisation besonders wichtig. Unter UML können zwar Abhängigkeiten zwischen Objekten definiert werden, eine Darstellung von anderen Objekten ist nicht möglich.
- Verantwortlichkeiten für geschäftliche Aktivitäten können nicht ausgedrückt werden, denn oft wird die Ausführung bestimmter Teilaktivitäten anderen Akteuren überlassen, wobei der ursprüngliche Akteur jedoch verantwortlich bleibt.

Informationen, die nicht in den Use-Case-Diagrammen aufgenommen werden können, müssen folglich in den genannten anderen Dokumenten erfasst werden. So wächst in der RUP-Praxis dem Glossar eine wichtige Rolle zu. Es wird zu einem wichtigen Element im Workflow ‚Business-Modellierung‘ zur Beschreibung der Zusammenhänge der Anwendungsdomäne. Die textuelle Beschreibung über das Glossar bringt im Einzelnen jedoch folgende Probleme mit sich:

- Begriffe können mehrdeutig verwendet werden,
- für wichtige Dinge können konkrete Begriffe fehlen,
- Begriffe sind kontextabhängig,

Je nach Projektgröße wird sogar empfohlen, nur ein Glossar zu erstellen und damit den gesamten geschäftlichen Kontext zu beschreiben. In diesem Fall wird auf die Business-Modellierung verzichtet.

Die Business-Modellierung konzentriert sich im RUP auf die Geschäftsprozesse und das Glossar. Es scheint weniger geeignet zu sein, um die Komplexität von anspruchsvollen E-Business-Anwendungen mit vielen Akteuren zu reduzieren. Das RUP-Framework ist als Vertreter des Software-Engineerings anzusehen und zeigt anhand seiner Defizite das Potential für weitere E-Business-Engineering-Ansätze auf, die nachfolgend vorgestellt werden.

Weiterhin bietet der RUP jedoch eine gute Einordnung der Business-Modellierung in den gesamten Softwareentwicklungsprozess.

2.2.1.2 E-BPMS

E-BPMS (Electronic-Business-Process-Management-Systems) ist ein Framework, das auch für die Modellierung von E-Business-Anwendungen geeignet ist (vgl. (Bayer, et al., 2001; Wriggers, 2003)). Es

soll die nahtlose Modellierung von der strategischen Geschäftsmodell-ebene bis zur operativen IT-Infrastruktur-Ebene ermöglichen und ist auf das Geschäftsprozessmanagement zentriert.

Als Ziel der Anwendung von E-BPMS wird eine stetige Verbesserung von Geschäftsabläufen angesehen. Neben der Verbesserung von Prozessen soll die Methode das Einbinden weiterer Maßnahmen und Technologien erlauben (siehe auch Abbildung 2.18). Dazu zählen:

- Reorganisation
- Qualitätsmanagement
- Workflow-Management-Systeme
- Standardsoftware
- Wissensmanagement

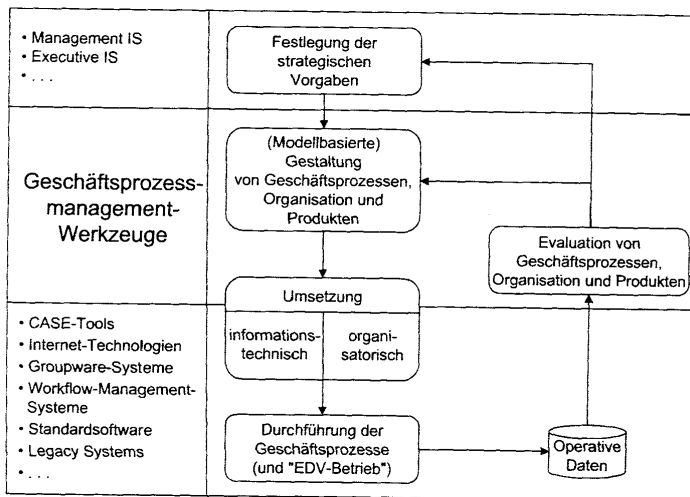


Abbildung 2.18: Regelkreis zum Geschäftsprozessmanagement nach dem E-BPMS-Paradigma

Die E-BPMS-Methode ist eine Entwicklung der Abteilung Knowledge Engineering des Informatikbereichs der Universität Wien. Sie findet im Spin-Off-Unternehmen BOC (Business Objectives Consulting) Verwendung. Das Unternehmen BOC entwickelte das Geschäftsprozessmanagement-Werkzeug *Adonis* (siehe Abbildung 2.19), mit dem das E-BPMS-Framework angewendet werden kann. Dieses Werkzeug wurde unter anderem wechselseitig mit Werkzeugen von IBM zur Prozessmodellierung integriert.

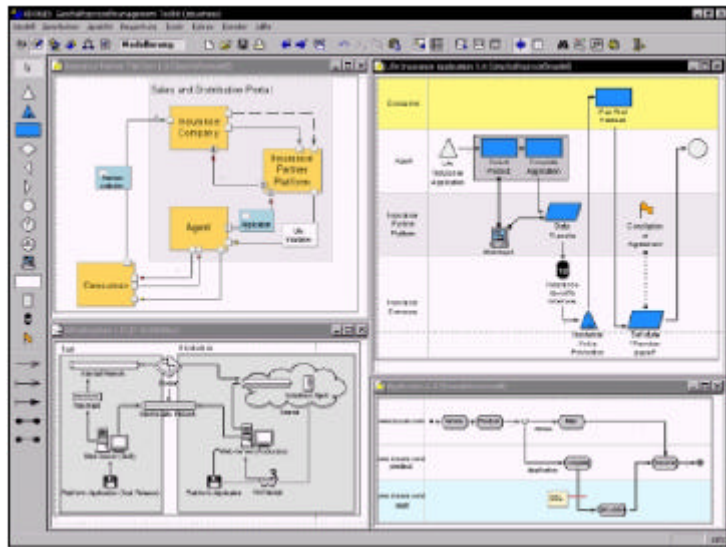


Abbildung 2.19: Das im (Meta-)Geschäftsprozessmanagement-Werkzeug ADONIS® umgesetzte E-BPMS-Framework

Eine Eigenschaft von E-BPMS ist das generische Vorgehensmodell, das an spezielle kunden- und projektspezifische Gegebenheiten angepasst werden kann.

Im Folgenden werden die Modellierungsebenen in E-BPMS und exemplarisch die Modellelemente für elektronische Geschäftsmodelle vorgestellt.

2.2.1.2.1 Modellierungsebenen in E-BPMS

E-BPMS unterscheidet vier Ebenen der Modellierung (siehe auch Abbildung 2.20), mit der die Komplexität von E-Business-Anwendungen beherrscht werden soll:

Strategische Ebene: Das Topmanagement entscheidet über das Geschäftsmodell eines Unternehmens, das bedeutet unter anderem die Marktstrategie, die Wahl der angebotenen Produkte und der einzubeziehenden Partner.

Geschäftsebene: Das operative Management verfeinert die strategischen Richtlinien geschäftsorientiert, d.h. sie definiert die Produkte und Geschäftsprozesse und setzt eine konkrete Organisationsstruktur um.

Implementierungsebene: Sie modelliert die Abhängigkeiten zwischen Produkten, Geschäftsprozessen, Organisationsstruktur und Informationssystemen (IS).

Ausführungsebene: Sie bildet die Laufzeitumgebung, Infrastruktur der Informationstechnologien (IT), wie z. B. Netzverbindungen, Server

und Zuweisung von E-Business-Anwendungen zu Hardwarekomponenten.

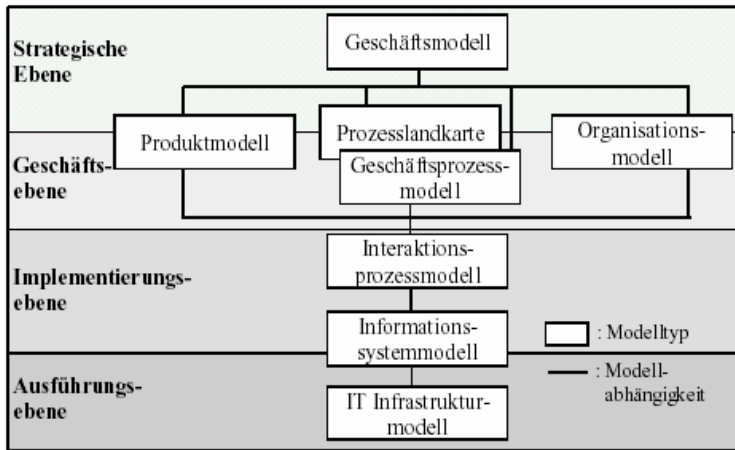


Abbildung 2.20: Modellierungsebenen im E-BPMS Framework

Abbildung 2.20 zeigt, durch welche Modelltypen die Zielsetzung der verschiedenen Ebenen umgesetzt wird. Ein Modelltyp besteht jeweils aus Modellierungsklassen und –beziehungen. Abhängigkeiten zwischen den einzelnen Ebenen werden durch modellübergreifende Referenzen festgelegt. Folgende Typen existieren auf den einzelnen Modellierungsebenen:

Strategische Ebene: Das *Geschäftsmodell* besteht aus einem dynamischen Teil (Produktangebot, Prozesse, Interaktionen der Akteure) und einem beschreibenden Teil mit geschäftspolitischen Entscheidungsgrundlagen (Business Plan, Finanzierungsplan, Beteiligungsverhältnisse, ...).

Geschäftsebene: Das *Produktmodell* spezifiziert ein Produkt oder eine Dienstleistung über deren Struktur, Geschäftsregeln, kalkulatorische Kosten etc. Die *Prozesslandkarte* gibt einen Überblick über identifizierte Geschäftsprozesse und deren Klassifikation (Kerngeschäftsprozesse, Serviceprozesse usw.). Das *Geschäftsprozessmodell* beschreibt die dynamischen Aspekte eines Geschäftsmodells. Im *Organisationsmodell* wird die Aufbauorganisation mit Elementen wie Organisationseinheiten, Rollen, Zuständigkeiten, Mitarbeitern, usw. beschrieben.

Implementierungsebene: Das *Interaktionsprozessmodell* beschreibt Interaktion und Navigation des Benutzers mit der Benutzeroberfläche (GUI). Im *Informationsmodell* wird die E-Business-Anwendung softwaretechnisch modelliert. Für diese Modellierung legt E-BPMS keine konkrete Modellierungssprache fest. Vielmehr kann je nach Zielplattform eine geeignete Sprache wie beispielsweise UML für objektorientierte Systeme

oder Workflow-Modellierung für ein Workflow-Management-System benutzt werden.

Ausführungsebene: Das *IT-Infrastrukturmodell* beschreibt Netz- und Hardwareressourcen eines Unternehmens und seiner Geschäftspartner.

2.2.1.2.2 Modellelemente für elektronische Geschäftsmodelle

In E-BPMS können die nachfolgenden Modellelemente für elektronische Geschäftsmodelle verwendet werden:

Geschäftsakteure sind Teilnehmer des Geschäftsmodells, deren rechtliche Stellung im E-Geschäftsmodell bedeutend ist. Zu ihnen zählen Personen, Unternehmen, Abteilungen oder Unternehmensnetzwerke. Sie verhalten sich ähnlich und ist ihre konkrete Ausprägung nicht von Bedeutung, werden sie zu abstrakten Geschäftsakteuren. Sie werden als Vierecke modelliert und gestapelt gezeichnet.

Kategorie von Geschäftsakteuren: Geschäftsakteure, die eine gleichartige Rolle spielen, werden zu einer Klasse gruppiert.

Produkte werden von den Geschäftsakteuren erstellt und ausgetauscht

Prozessfragmente: Bündel von Aktivitäten, das jeweils von einem Geschäftsakteur ausführt wird.

Darüber hinaus gibt es noch folgende Beziehungsklassen:

- *Produktfluss:* Er zeigt an, welche Produkte welchen Geschäftsakteuren angeboten werden und die Art des Transports.
- *Kapitalfluss:* Er definiert Zahlungsströme zwischen den Geschäftsakteuren.
- *Informationsfluss:* Er zeigt welche Informationen zwischen den Geschäftsakteuren ausgetauscht werden.
- *Juristische Beziehung:* Sie grenzt die Art der Kooperation zwischen den Geschäftsakteuren ab.
- *Aufbauorganisatorische Beziehung:* Sie zeigt bei komplexen Geschäftsakteuren die Aufbauorganisation.

2.2.1.3 RSD

Die ‚Rapid-Service-Development‘ (RSD) Methode definiert einen integrierten, modellbasierten Ansatz, um schnell und effizient E-Business-Lösungen zu entwerfen. Sie wurde explizit mit dem Ziel geschaffen, den Bedürfnissen über das Internet vernetzter Unternehmen gerecht zu werden. Das Networked-Enterprise (NE) steht bei RSD im Mittelpunkt der Betrachtung.

Unter *Networked-Enterprise (NE)* wird jedes koordinierte Vorgehen von mindestens zwei autonomen Parteien verstanden, die internetbasierte Technologien für Information und Kommunikation benutzen (Steen, et al., 2002). Dies umfasst E-Cooperation, Virtuelle Unternehmen, Lieferketten, elektronische Märkte, Auktionen und virtuelle Gemeinschaften. NE können nach Charakter und Größe sehr unterschiedliche Formen annehmen, von kooperierenden Ein-Mann-Firmen bis zu Global-Player-Konzernen, die Lieferanten, Handel, Logistik und Unternehmen in ihre Lieferkette integrieren. Sie lassen sich nach Grad von Informationsaustausch, Zusammenarbeit, gegenseitiger Abhängigkeit und Stabilität sowie der Topologie (Stern, Kette oder Netzwerk) unterscheiden.

Da die NE-Definition eine große Bandbreite von Kooperationen zwischen Organisationen umfasst, sind die RSD-Konzepte relativ umfassend angelegt.

RSD entstand aus der Initiative ‚Giga-Port‘ mehrerer niederländischer Ministerien mit dem Ziel, in Holland eine moderne breitbandige Kommunikationsinfrastruktur zu schaffen.

Betrachtet wird in dieser Arbeit der Bereich, in dem RSD es den Nutzern ermöglicht, Software zur elektronischen Ausführung von Geschäften über Unternehmensgrenzen hinweg selbst zu generieren (vgl. (GigaTS, 2002)).

2.2.1.3.1 Die RSD-Cornerstones

Die RSD-Methode (Steen, et al., 2000) grenzt sieben verschiedene Betrachtungsebenen ab, die so genannten ‚Cornerstones‘ (siehe Abbildung 2.21):

Ambition & Scope beschreibt die Ziele der zu entwickelnden Transaktionsdienste jeweils unter Berücksichtigung der geschäftlichen Ziele und der einzusetzenden Technologien. Auch die Startpunkte und Abhängigkeiten werden festgelegt.

Networked-Enterprise modelliert auf hoher Ebene Kooperationen der beteiligten Akteure jeweils unter Berücksichtigung ihrer geschäftlichen Funktionen und der Austauschflüsse zwischen ihnen.

Transaction-Scenarios stellen die Prozesse und Transaktionen zwischen den Organisationen dar, die nötig sind, damit eine Kooperation stattfindet.

Procedures sind detaillierte Beschreibungen der Interaktionen und Nachrichtenformate, die zwischen den Organisationen zur Ausführung von Transaktionen eingesetzt werden.

Systems definieren die Architektur und die Komposition von Transaktionssystemen.

System-Components spezifizieren die Funktionalität von klar abgegrenzten, wieder verwendbaren Softwaremodulen.

Protocols beschreiben im Detail die Protokolle zur Kommunikation zwischen den Komponenten und teilweise auch innere Abläufe der Komponenten.

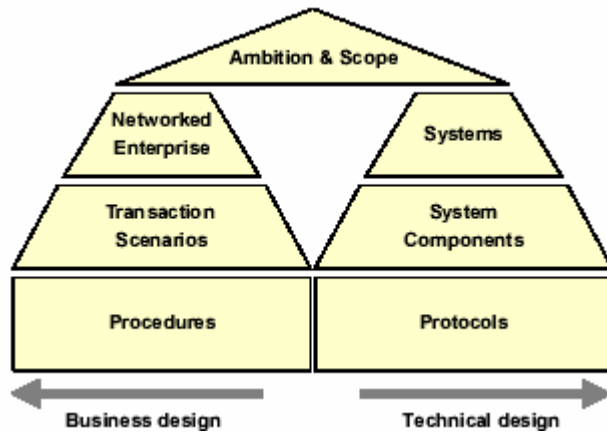


Abbildung 2.21: Die sieben ‚Cornerstones‘ des RSD-Frameworks

Der Detailgrad der von den ‚Cornerstones‘ beschriebenen Modelle und Designentscheidungen nimmt von oben nach unten zu. Die rechte Seite einer Ebene kann jeweils als technische Umsetzung des Business-Designs der linken Seite verstanden werden. Neben den zwei Dimensionen der Genauigkeit und der Business- bzw. Geschäftsorientierung existiert als implizite dritte Dimension die Entwicklungsebene, die jeweils von Analyse über Design bis zur Umsetzung reicht. Diese Entwicklung der dritten Dimension findet in jedem ‚Cornerstone‘ statt. Denn Ziel ist nicht nur eine technische Gesamtlösung, sondern auch eine Integration der Geschäftsmodelle der Beteiligten.

Je nachdem welche Teilaufgabe des gesamten E-Business-Systems umgesetzt werden soll (z. B. Prozessstandardisierung, Legacy-Integration, Neuentwicklung einer Anwendung), gibt es verschiedene Wege, um das RSD-Framework zu nutzen. Dabei müssen nicht alle ‚Cornerstones‘ berücksichtigt werden.

Die *Networked-Enterprise-Models* (NEMs) sollen bei RSD alle spezifischen, relevanten Aspekte erfassen. Mit einer Vielfalt an Diagrammen, die jeweils eine spezielle Sicht auf das Modell liefern, wird die Kooperation der am NE teilnehmenden Organisationen beschrieben. NEs sind in der Regel sehr komplex, da meist viele Kooperationspartner mit ihren Aktivitäten modelliert werden müssen. Um die Komplexität dieser speziellen Modelle zu reduzieren, wird eine Aufteilung in einfachere Unter-NEMs vorgeschlagen, die sich auf bestimmte Teile der Kooperation beschränken.

Die NEMs werden mittels einer eigenen Sprache definiert, die in den drei geschäftlich orientierten ‚Cornerstones‘ (Networked-Enterprise, Transaction-Scenarios und Procedures) benutzt werden. Für die technisch orientierte Seite (Systems, System-Components, Protocols) wird UML zur Modellierung empfohlen. UML wird ebenfalls zur Beschreibung der abstrakten Syntax der NEM-Sprache benutzt.

Folgende fünf Bereiche beziehungsweise Domänen werden als entscheidende Aspekte eines NE angesehen:

- *Akteur (Actor)*: Konzepte zur Modellierung der physikalischen oder logischen Struktur eines NE,
- *Rolle (Role)*: Konzepte zur Modellierung der Verantwortlichkeiten der Beteiligten einander gegenüber,
- *Funktion (Function)*: Konzepte zur Modellierung der Aktivitäten oder Funktionen und der Flüsse von Informationen, Produkten und Geld in einem NE,
- *Prozess (Process)*: Konzepte zur Modellierung der interorganisatorischen Prozesse in einem NE,
- *Daten (Data)*: Konzepte für Modellierung der Daten und Nachrichten.

Zu einer umfassenden Beschreibung eines NE tragen alle fünf Sichten gemeinsam bei. Die ersten vier (also alle außer der Daten-Domäne) liefern jeweils einen eigenen Diagrammtyp. Die Verbindung zwischen den Sichten sollen Konsistenzregeln herstellen (z. B. Zuordnungstabellen), welche die Diagramme in Beziehung setzen.

In Abbildung 2.22 werden die grundlegenden Modellierungskonzepte und die Modellbeziehungen der NEM-Sprache gezeigt. Die NEMs bestehen also unter anderem aus den wesentlichen Elementen Akteur, Kanäle (Channel), Rolle, Verlauf (Flow) und Funktion. Akteure übernehmen Rollen und führen Funktionen aus. Eine Rolle wird durch keine oder mehrere Akteuren wahrgenommen und umfasst eine oder mehrere Funktionen. Ein Kanal verbindet mindestens zwei Akteuren. Ein Verlauf kann Kanäle benutzen und Rollen oder Funktionen passieren.

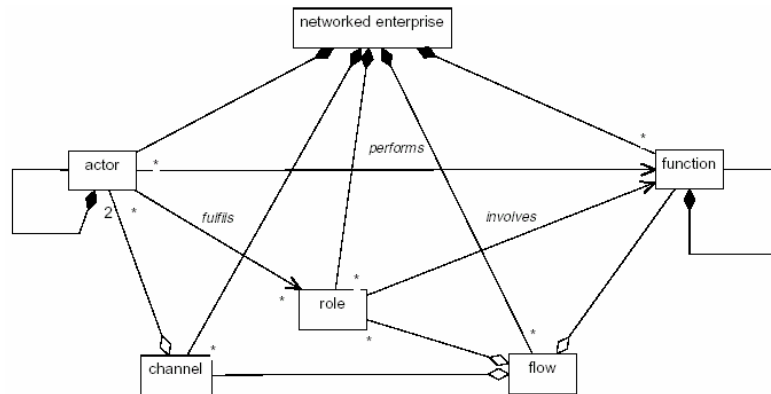


Abbildung 2.22: RSD-Metamodell zur Erfassung von Networked-Enterprises

Besonders ausführlich wird in den nächsten Abschnitten der Networked-Enterprise-Cornerstone mit den Domänen Akteur, Rolle und Funktion beschrieben, da dieser Teil von RSD den größten Bezug zum E-Business-Modellierung aufweist.

2.2.1.3.2 Die RSD-Modelle

Die RSD-Modelle der fünf Bereiche Akteur, Rolle, Funktion, Prozess und Daten werden im Folgenden näher beschrieben. Die letzten beiden Bereiche jedoch nur in knapper Form, da sie im Bereich der hier im Vordergrund stehenden Networked-Enterprises kaum Verwendung finden. Weitergehende Details zur graphischen Repräsentation der Modelle finden sich bei Steen et al. (Steen, et al., 2000) oder Wriggers (Wriggers, 2003).

In der *Akteur-Domäne* wird ein Diagramm beschrieben, wer die Partner in einem NE sind und mit wem sie durch welche Kanäle („Channels“) interagieren. Je nach gewählter Sichtweise werden einzelne Personen, organisatorische Einheiten, wie Abteilungen oder wie im Normalfall Unternehmen, als Akteure erfasst.

Ein Akteur zeichnet sich durch Folgendes aus:

- Ein Akteur ist eine aktive organisatorische Einheit. Andere Elemente eines NE, wie Produkte und Ressourcen, besitzen im Gegensatz dazu nur ein passives Verhalten,
- Anonyme Akteure wie Kunden oder Logistikdienstleister werden als Role-Player bezeichnet.
- Akteure können weitere Unterakteure enthalten. Dies ist nützlich bei hierarchischen Unternehmensstrukturen, z. B. ein Unternehmen enthält Abteilungen, die wiederum aus Personen bestehen.

Damit Akteure miteinander interagieren können, werden sie durch Kanäle verbunden. Es wird dabei zwischen logischen Kanälen für den Austausch von Informationen (z. B. per Telefon, E-Mail, Internet oder Virtual-Private-Network) oder physikalischen Kanälen für den Austausch von Gütern bzw. Dokumenten (z. B. LKW oder Seeschiff) unterschieden. Zwar verbinden Kanäle meist zwei Akteure, jedoch sind für Multi-Cast-Operationen auch mehrere Teilnehmer denkbar. Sie werden sowohl bidirektional als auch unidirektional genutzt. Mitunter können sie benannt werden, z. B. „Internet“ wenn zwei Akteure entsprechend Informationen austauschen. Bei logischen Kanälen wird der Name meist weggelassen.

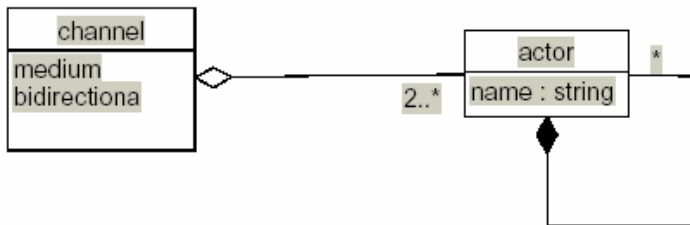


Abbildung 2.23: Konzepte in der Akteur-Domäne

Unter Berücksichtigung der Rollen wird von konkreten Akteuren abstrahiert. Es entsteht eine *Rollen-Domäne*. Die Abbildung 124 verdeutlicht die Stellung der Akteure hinsichtlich ihrer Rollen und Funktionen in einem NE. Dabei ist von besonderer Bedeutung, inwiefern Informationen, Geld und Güter zwischen diesen Rollen fließen.

Eine Rolle repräsentiert eine Ansammlung von Aufgaben und Verhaltensweisen, die von einem oder mehreren Akteuren erfüllt werden.

Jeder Akteur eines NE nimmt eine Rolle ein, sei es nun als Kunde, Lieferant, ‚Broker‘, Versteigerer, Intermediär oder Käufer, mit der diese abstrakt beschrieben werden.

Durch die Abstrahierung erhält man ein generisches und wieder verwendbares Modell, da ein konkreter Akteur durch seine Rollenfestlegung austauschbar ist. Ebenso kann ein Akteur bedingt durch eine Veränderung im Verlaufe der Zeit mehr als eine Rolle einnehmen. Mitunter kann eine Rolle von mehr als einem Actor ausgefüllt werden.

Eine Schachtelung einer Rolle ist nicht möglich, d.h. eine Rolle kann nicht Teil einer anderen sein, da auch die jeweils zugeschriebenen Aufgaben nicht geschachtelt werden können.

Wird nun eine zeitlich geordnete Sequenz von Informations- oder Gütertransfers zwischen Rollen bzw. Funktionen betrachtet, spricht man von einem Verlauf. Ein Verlauf ist von einer oder mehreren Quellen auf eine oder mehrere Senken gerichtet. Er besteht aus einem oder mehreren Verlaufsssegmenten. Rollen enthalten wiederum keine oder mehrere Ver-

laufsverbindungen (Connector), die für die Segmente Verbindungspunkte darstellen. Ein Verlauf wird abwechselnd aus Verbindungen und Segmenten zusammengesetzt. Ein Verlauf muss mit einer Verbindung starten und enden (siehe auch Abbildung 2.24).

Rollen sind Quellen und Senken für den Verlauf. Sie produzieren oder verbrauchen Verläufe. Diese können an den Verbindungen geteilt oder zusammengeführt werden, wenn sie entsprechend als Verlaufs-Join und Verlaufs-Split ausgeführt sind. Verlaufssegmente können Kanälen zugeordnet werden. Ein Verlauf darf keine Zyklen enthalten. Ihre Richtung wird mit Pfeilen angezeigt. Jeder Verlauf hat einen Namen und einen Typ. Der Typ bestimmt, welche Art von Handelsgütern transferiert wird:

- Information, z. B. Produktbeschreibungen, Preislisten
- Gegenstände, z. B. Fertigprodukte, Teile, Geld

Sie können in einer Typhierarchie organisiert werden. Eine Aufnahme von Objekten in einem Verlauf kann nur stattfinden, wenn ein entsprechender Subtyp vorhanden ist. Zum Beispiel kann ein Produktverlauf Rohmaterial und Fertigprodukte transferieren, aber zu einem Rohmaterialverlauf passen keine Ersatzteile.

Verläufe können einem oder mehreren Kanälen zugeordnet werden.

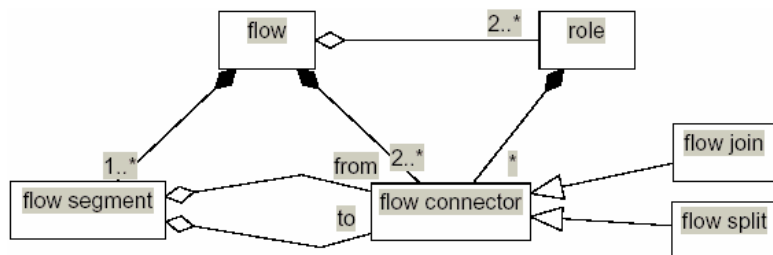


Abbildung 2.24: Konzepte in der Rollen-Domäne

Beispiele für Verläufe:

- In einer Lieferkette bestehen ein Downstream-Verlauf von Produkten vom Produzenten zum Endverbraucher und ein entgegengesetzter Upstream-Verlauf mit Bestellungen und Geld.
- Bei einer Auktion gibt es einen Verlauf von Produktinformationen vom Verkäufer, über den Auktionator, zum Käufer. Ein Verlauf aus Geld und Produkten läuft in entgegengesetzte Richtung.

In der *Funktionen-Domäne* in Abbildung 2.26 werden die Verläufe zwischen den Funktionen dargestellt. Unter Funktionen wird eine logische Menge von Aktivitäten verstanden. Sie besitzen Ports durch die Verläufe ein- und austreten. Daher kann man Funktionen als Relation zwischen ein- und austretenden Verläufen beschreiben. Die Ports stellen das Interface der Funktion nach außen dar. Funktionen können ebenso wie Rollen

Verlaufsverbindungen zugeordnet werden. Die Ports sind dann die Punkte, an denen ein Verlaufssegment eine Funktion betritt oder verlässt.

Funktionen steuern Vorgänge, die von Verläufen produziert, verbraucht, geteilt („Split“), zusammengeführt („Join“) oder transformiert werden. Sie verfeinern daher das Verhalten der Verlaufsverbindungen und werden funktional definiert, d.h. man kann für sie keinen Startzeitpunkt, Dauer oder Ende bestimmen. Da Funktionen sich auf eine einzelne Rolle beziehen, haben sie eine festgelegte Aufgabe, für dessen Ausführung letztendlich die Rolle verantwortlich ist. Ansonsten kann eine Funktion von jeder Rolle ausgeführt werden. Es besteht auch die Möglichkeit, dass Funktionen aus mehreren Unterfunktionen bestehen, deren Türen mit denen der übergeordneten Funktion zusammenfallen.

In Abbildung 2.25 wird das Metamodell aus Abbildung 2.24 mit Verläufen und Rollen um die Funktionen verfeinert. Abbildung 2.26 zeigt die Verbindung der wesentlichen Elemente der Funktionen-Domäne: Funktion, Verlauf, Verlaufssegment und „Port“.

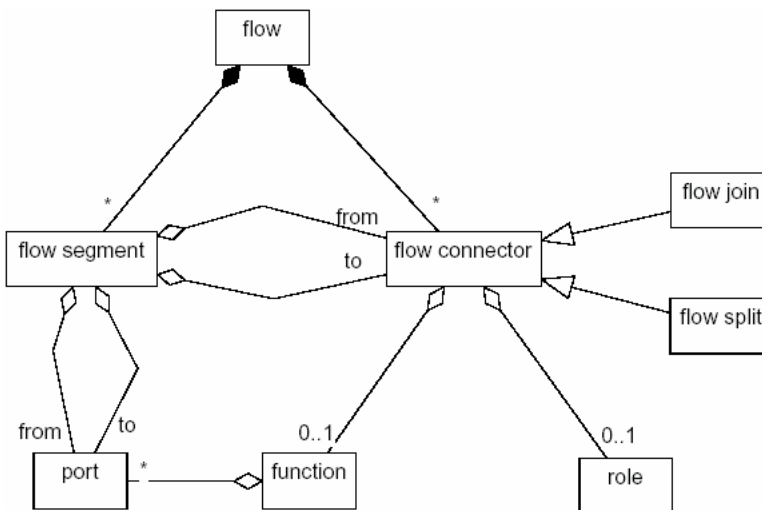


Abbildung 2.25: Metamodell für Verläufe, Rollen und Funktionen

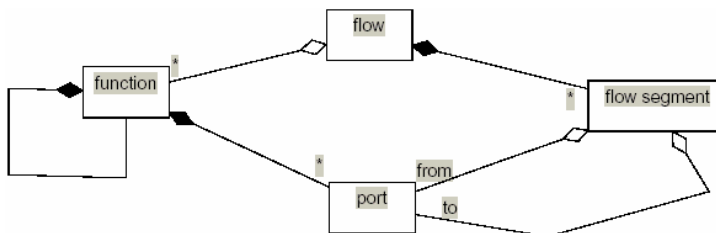


Abbildung 2.26: Konzepte in der Funktionen-Domäne

Als ‚Store‘ wird eine Funktion bezeichnet, die Verläufe verbindet, zusammenführt („Join“) oder teilt („Split“), ohne dabei Inhalt oder Typ zu verändern.

Es kommt häufig vor, dass die über Verläufe zwischen zwei Funktionen transferierten Güter oder Informationen (Artefakte) nicht direkt weiterverarbeitet werden, sondern zunächst in einem Lagerhaus oder einer Datenbank zwischengespeichert werden. Daher hat man bei RSD mit den ‚Stores‘ spezielle Punkte geschaffen, an denen Geschäftsprozesse entkoppelt werden können.

Wird ein bestimmtes Verhalten in einem speziellen Fall einmalig angewendet, so spricht man von einer Aktion („Aktion“). Sie können Funktionen, Rollen und Akteuren zugeordnet werden. Eine Funktion hingegen beschreibt die generelle Anwendung desselben Verhaltens in allen Fällen. Die Aktion stellt eine Abstraktion für eine Aktivität in der realen Welt dar. Sie tritt zu einem bestimmten Zeitpunkt auf, ist atomar, kann aber auch eine gewisse Zeit andauern.

In der *Prozess-Domäne* in Abbildung 2.27 werden die Aktivitäten im Bereich des NE modelliert. Man unterscheidet verschiedene Prozesse:

Ein Business-Prozess beschreibt Verhaltensschritte, die zur Erreichung eines bestimmten Ziels führen. Interorganisationelle Prozesse sind solche, die Grenzen von Organisationen überschreiten. Ein Prozess startet mit einem Trigger (Anstoß), umfasst ‚Actions‘, ‚Interactions‘, ‚Transactions‘, ‚Transfers‘ oder ‚Causality-Relations‘ und endet mit einer Abschlussaktion.

Die Konzepte der Prozess-Domäne stammen überwiegend aus der Business-Prozess-Modellierungssprache Amber, die im Testbed Projekt (Lankhorst, et al., 1999) entwickelt wurde. Für die NEM-Sprache wurden als Verfeinerungen der ‚Interaction‘ die Elemente ‚Transaction‘ und ‚Transfer‘ hinzugefügt (siehe Abbildung 2.27). Ebenso wurden als Spezialisierung der Interaction-Relation die ‚Transaction‘ und die ‚Transfer-Relation‘ eingeführt.

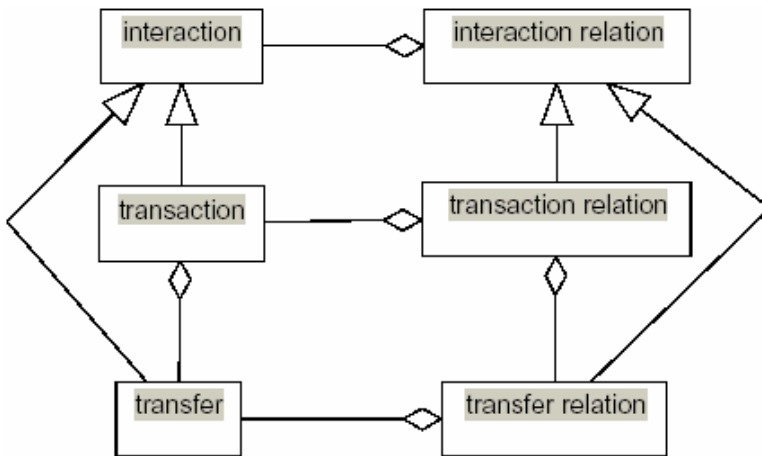


Abbildung 2.27: Konzepte der Prozess-Domäne

Das wichtigste Element der *Daten-Domäne* ist das Handelsgut, Artefakt genannt. Ein Artefakt repräsentiert ein logisch oder physikalisch passives Objekt. Das sind beispielsweise Produkte, Nachrichten, Verträge, Formulare und andere Güter oder Informationen. Ein Artefakt hat einen Typ, z. B. Information, Geld, Dokument oder Nachricht. Der Typ des Artefakts kann auch aus einfachen Typen zusammengesetzt sein.

Artefakte kommen bei ‚Actions‘, ‚Interactions‘, ‚Transactions‘ und ‚Transfers‘ vor. Verläufe können auch als Strom von Artefakten angesehen werden, wodurch der Verlaufstyp dem Artefakttyp entspricht. ‚Stores‘ sind Sammelager für Artefakte eines spezifischen Typs.

Wie Artefakttypen modelliert werden sollen, ist bei RSD noch nicht endgültig festgelegt. Vorgeschlagen werden zunächst UML-Klassendiagramme.

2.2.2 E3-Value-Modellierung

Die e³-Value-Modellierung ist ein von Gordijn (Gordijn, 2002) entwickelter wertbasierter Ansatz zur E-Business-Modellierung, der zwar umfassend angelegt ist, bei dem jedoch E-Business-Geschäftsmodelle im Vordergrund stehen. Er bedient sich graphischer, konzeptueller Modellierung und trägt so über eine gute Verständlichkeit der Modelle auch zu einer hohen praktischen Nutzbarkeit des Ansatzes bei. Es ist eine Möglichkeit zur konkreten Szenario-Modellierung enthalten.

Bei der Entwicklung von E-Business-Lösungen identifiziert Gordijn drei verschiedene Sichten bei der Modellierung (siehe Abbildung 2.28):

Das *E-Business-Modell* zeigt die grundlegenden Abläufe eines Unternehmens zur Umsetzung seiner Geschäftsidee. Nur Aktivitäten, die von außen betrachtet wertvoll sind (also Value enthalten), werden untersucht. Interne Prozesse sind ausgeblendet. Das E-Business-Modell beantwortet die Fragen: *wer bietet wem was* und erwartet dafür *welche Gegenleistung*. Die Gestaltung der Abläufe dieser Ebene obliegt normalerweise dem Top-Management. Modelliert wird auf dieser Stufe durch die E³-Value-Diagramme.

Durch die *E-Business-Prozesse* werden die Aktivitäten und Abläufe im Unternehmen detaillierter beschrieben. Leitfrage ist, wie und von wem werden welche Aktivitäten ausgeführt. Konkrete Tätigkeiten von Mitarbeitern werden deutlich. Dies kann mit High-Level-Petri-Netzen (daraus sind die Ereignisgesteuerten Prozessketten EPK entwickelt worden) oder UML-Diagrammen (‘Activity’ / ‘Statecharts’ / ‘Use-Cases’) modelliert werden. Verantwortlich auf dieser Ebene sind Manager der zweiten Führungsebene.

Die Sicht der *E-Software-Architektur* liefert der Software ein Modell, welche die Aktivitäten im Unternehmen unterstützt. Hier kann beispielsweise der RUP-Ansatz eingesetzt werden. Zuständig auf dieser Ebene ist in der Regel die IT-Abteilung eines Unternehmens.

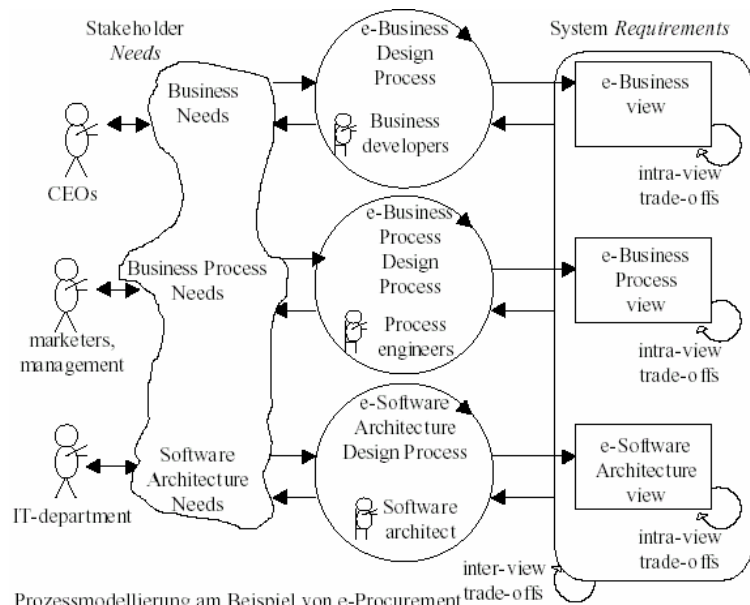


Abbildung 2.28: Sichten bei der Entwicklung von E-Business-Lösungen

Bei der Entwicklung der e³-Value-Modellierung gab es mehrere Anforderungen:

- Bisher gab es keine Möglichkeit, das Business-Modell losgelöst vom Prozessmodell zu betrachten.
- Es sollte eine formale, graphische Notation verwendet werden. Die bisher vorherrschende Praxis textueller Beschreibungen bietet viele Möglichkeiten für Missverständnisse. Eine graphische Notation dagegen bietet weiterhin den Vorteil eines schnellen Überblicks. Insbesondere Personen unterschiedlicher Fachrichtungen (Wirtschaft & IT) haben so die Möglichkeit auf der Basis eines verständlichen Modells zu kommunizieren
- Das Modell soll die Möglichkeit zur Analyse bieten.
 - So sollen Verhältnisse und Konstellationen, die sich aus den Geschäftsbeziehungen ergeben, identifiziert werden können.
 - Das Modell soll eine Grundlage für Profitabschätzungen bieten.
 - Das Modell soll ‚Was-wäre-wenn-Analysen‘ unterstützen.
- Das Modell soll die Anforderung der *Einfachheit* erfüllen. Dazu gehört es, eine rasche Übersicht zu ermöglichen. Dies hat zur Folge, dass nur eine begrenzte Anzahl an Elementen, Konstruktionen und Funktionalitäten zur Verfügung gestellt werden. Das erleichtert sowohl die Modellierung als auch das Lesen eines vorliegenden Modells.

2.2.2.1 Basisstrukturen der e³-Value-Modellierung

Bei der e³-Value-Modellierung geht es grundsätzlich darum, die Betrachtung der Geschäftsbeziehungen auf den reinen Nutzen beziehungsweise Profit auszurichten, den so genannten *Value*. Dieser kann in zahlreichen Formen wie Produkten, Dienstleistungen, Wissen oder Erfahrungen auftreten. Alle Geschäftsbeziehungen werden lediglich nach dem *Value*, den sie erbringen, beurteilt.

Zur Beschreibung von Geschäftsszenarien werden in der e³-Value-Modellierung folgende Konzepte verwendet:

Akteur:

Ein Akteur („Actor“) wird von seiner Umgebung als eine unabhängige wirtschaftliche und handelnde Einheit wahrgenommen. Als Akteure werden in erster Linie Firmen oder Personen auftreten. Ein Akteur kann die so genannten Value-Aktivitäten ausführen, die weiter unten beschrieben werden. Durch die Ausführung dieser Aktivitäten verschafft sich jeder Akteur wirtschaftliche Vorteile.

Akteure werden durch ein Rechteck mit spitzen Ecken dargestellt. Im Inneren des Rechteckes kann zweckmäßigerweise eine Benennung stehen.

Wertobjekte:

Akteure tauschen Wertobjekte (,Value-Object') miteinander aus. Ein Wertobjekt kann eine Dienstleistung, ein Produkt oder sogar eine Erfahrung sein. Sehr wichtig für dessen Charakterisierung ist die Tatsache, dass es für einen oder für mehrere der beteiligten Akteure von Wert ist.

Die Wertobjekte werden als Beschriftungen der Pfeile repräsentiert, die ihrerseits den weiter unten beschriebenen Wertaustausch bezeichnen.

,Value-Port':

Ein Akteur benutzt einen ,Value-Port', um seiner Umgebung zu zeigen, dass er bestimmte Wertobjekte anbietet oder empfangen kann. Durch dieses Konzept ist eine abstrakte Darstellung von Beziehungen zwischen Akteuren möglich, deren Anfragen und Angebote sich entsprechen.

,Value-Ports' werden als schwarze Punkte in den Akteuren dargestellt.

Wertaustausch:

Ein Wertaustausch (,Value-Exchange') findet zwischen zwei ,Value-Ports' statt und verbindet diese so miteinander. Er repräsentiert einen potenziellen Weg, auf dem Wertobjekte zwischen den Akteuren ausgetauscht werden können.

Ein Wertaustausch wird mit einem gerichteten Pfeil zwischen den beteiligten ,Value-Ports' gekennzeichnet.

,Value-Interface':

Das ,Value-Interface' baut auf dem Konzept des ,Value-Ports' auf. Durch ein ,Value-Interface' werden diese gruppiert, um einen Zusammenhang in einer Transaktion darzustellen. Es gibt an, welche Wertobjekte unter welchen Bedingungen gegeneinander getauscht werden. Sie können nicht weiter geteilt werden, d.h. entweder finden an allen in diesem ,Value-Interface' gruppierten Ports entsprechende Transaktionen statt oder an keinem. Akteure haben ein oder mehrere ,Value-Interfaces'.

,Value-Interfaces' werden als abgerundete Rechtecke um eine Gruppe von ,Value-Ports' dargestellt.

Wertangebot:

Ein Wertangebot (,Value-Offering') ist eine Zusammenfassung oder Gruppierung des Wertaustauschs. Es zeigt, welche Wertobjekte als Gegenleistung für andere ausgetauscht werden können. Ein Wertangebot darf entweder vollständig oder gar nicht ausgeführt werden.

Es wird implizit durch ein ,Value-Interface' festgelegt.

Marktsegment:

Als Marktsegment ('Market-Segment') werden mehrere gleichartige Akteure bezeichnet. Die Akteure haben die gleichen Charakteristika und Entscheidungsfunktionen an den 'Value-Interfaces' und können daher aus Sicht des Business-Modells analog behandelt werden. Das Konzept des Marktsegments teilt somit den aus einer Gesamtmenge von Akteuren bestehenden Markt in gleichartige Untermengen auf.

Marktsegmente werden durch dreimalig hintereinander gelegte, überlappende Wiederholung der Rechtecke der Akteure gekennzeichnet.

„Composite-Actor“:

Ein „Composite-Actor“ (dt. zusammengesetzter Akteur) bezeichnet eine Gruppe von Akteuren, die Teile ihrer Aktivitäten zusammengeschlossen haben und ihrer Umwelt gegenüber als ein einziger Akteur auftreten. Dazu ist es überdies nötig, dass die beteiligten Akteure gemeinsame „Value-Interfaces“ vereinbaren, über die sie mit ihrer Umwelt interagieren.

In der e³-Value-Modellierung werden zusammengesetzte Akteure durch ein Rechteck gekennzeichnet, das die zusammenarbeitenden Akteure umschließt.

Mehrwertaktivität:

Eine Mehrwertaktivität („Value-Activity“) ist eine Handlung innerhalb des Modells, die von einem Akteur ausgeführt wird und dessen Nutzen erhöht. Üblicherweise können derartige Aktivitäten in Unteraktivitäten unterteilt werden, wobei es für die Zulässigkeit einer solchen Unterteilung nötig ist, dass jede dieser Unteraktivitäten ebenfalls den Nutzen erhöht.

Eine Mehrwertaktivität wird repräsentiert durch ein Rechteck mit abgerundeten Ecken, welches im Rahmen des Akteurs eingezeichnet wird, der die Mehrwertaktivität ausführt.

2.2.2.2 „Use-Cases“

In späteren Arbeiten schlägt Gordijn so genannte „Use-Cases“ vor, die den Ablauf einer Geschäftsbeziehung verdeutlichen sollen. Dieser Ansatz geht sehr stark in den Bereich der Prozessmodellierung. Darüber hinaus widerspricht diese Funktion den ursprünglichen Anforderungen der e³-Value-Modellierung, da sie das Modell überlädt und die Übersichtlichkeit und die Verständlichkeit verringert.

Daher werden „Use-Cases“ im Folgenden nicht weiter berücksichtigt.

2.2.2.3 Analyse von e³-Value-Modellen

Bei der Analyse von e³-Value-Modellen kann zwischen zwei Möglichkeiten der Analyse unterschieden werden. Einerseits ergibt sich durch die grafische Darstellung die Möglichkeit in übersichtlicher Form wesentli-

che Elemente und Eigenschaften des Geschäftsmodells zu erfassen (‚Direkte Analyse‘). Andererseits kann analytisch auf Basis des Modells ermittelt werden, wie die Unternehmung auf variable Umgebungsbedingungen reagiert (‚Was wäre wenn‘ Analysen / ‚E-Valuation‘).

2.2.2.3.1 Direkte Analyse

Bisher stand die Modellierung einer Geschäftsidee anhand des e³-Value-Konzepts im Vordergrund. Dadurch wurde die Geschäftsidee genauer ausgearbeitet und es sollte ein allgemein kommunizierbares Verständnis geschaffen werden.

Dieses Verständnis ist insbesondere für die Bewertung eines bereits modellierten Szenarios notwendig. Im Folgenden wird hervorgehoben, welche Informationen durch ‚einfaches Ansehen‘ dem Modell entnommen werden können:

Wertströme: Wichtig ist die Möglichkeit, zu erkennen, in welche Richtung welche Ströme von Werten, beispielsweise von Geld, fließen und wodurch dieser Fluss ausgelöst wird. Diese Möglichkeit ist durch die grafische Darstellung, zusammen mit den ‚Use-Case-Maps‘ gegeben. Es kann eindeutig jeder kausale Zusammenhang zwischen verschiedenen Wertaustauschen identifiziert werden.

‚Bundling‘: Das Gruppieren von ‚Value-Ports‘ zum Zweck der Erschaffung unteilbarer Kombinationen, in denen Objekte ausgetauscht werden, kann aus verschiedenen Gründen erfolgen (z.B. Wertsteigerung gegenüber den Einzelprodukten). Solche Kombinationen und die genauen Bedingungen können in einem Modell festgelegt und wieder erkannt werden.

Kundenbindung: Es ist interessant zu identifizieren, an welche(n) Anbieter ein Kunde gebunden ist, d.h. von wem er seine Produkte bezieht. Zum Beispiel ermöglicht dieses Wissen die Abschätzung der Wirksamkeit von gezielten Maßnahmen im CRM. Die Möglichkeit, Kundenbindung zu erkennen, ist eine weitere Information, die das Modell zur Verfügung stellen kann.

Preisbestimmung: Auch über diejenigen Akteure, die in der Lage sind Preise für bestimmte Produkte festzulegen, gibt das Modell Auskunft. Dieses Wissen ermöglicht zum einen die Abschätzung der Macht gegenüber seinen Kunden, die der Anbieter im Szenario hat (ggf. ein Monopol), und zum anderen die Abhängigkeit eines Anbieters von anderen Anbietern (Zulieferern).

Wettbewerb: Auch über die Anbieterverhältnisse aus Sicht des Kunden kann Aufschluss gewonnen werden. So kann die Stärke des Kunden, die mit zunehmendem Wettbewerb wächst und sich durch mehrere Anbieter und Wertangebote äußert, abgeschätzt werden.

2.2.2.3.2 E-Valuation

Auf Grundlage des Modells können ‚Was wäre wenn‘-Analysen durchgeführt werden, die Aufschluss darüber geben können, wie sich eine Unternehmung bei Veränderung der Umgebungsbedingungen verhält.

Die Analyse erfolgt im Wesentlichen in drei Schritten:

1. Erstellung einer Profit-Nutzen-Schätzung:

Im ersten Schritt muss für jeden Akteur eine Nutzenaufstellung angefertigt werden, damit der für ihn entstehende Nutzen abgeschätzt werden kann. Der Weg zu dieser Aufstellung ist folgender:

Zunächst wird eine vorläufige Aufstellung angefertigt, indem beim Startstimulus beginnend der Szenariopfad durchlaufen wird. Jedes Passieren eines ‚Value-Interfaces‘ führt dabei zu einem Hinzufügen der ein- und ausgehenden Wertobjekte zu einer Aufstellung. Ist diese Aufstellung vollständig, so wird sie für jeden Akteur reduziert, indem alle irrelevanten Objekte wieder entfernt werden. Irrelevant ist ein Object, wenn es keinen Wertfluss darstellt und an den Akteur in dem gleichen Szenariopfad wieder abgegeben wird, in dem er es erhalten hat.

2. Bewertungsschema:

Nachdem nun die Aufstellung vorliegt, muss eine Bewertung der Wertobjekte erfolgen. Häufig wird diese Bewertung in Form von Funktionen erfolgen, die auch die Grundlage für die spätere Veränderung der Variablen bilden.

3. E-Valuation:

Da jetzt sowohl die Aufstellung der Wertobjekte, als auch das Bewertungsschema vorliegt, kann das zugrunde liegende Geschäftsmodell auf Rentabilität überprüft werden, so dass eine Aussage über die Profitstabilität eines Modells gemacht werden kann. Dabei ist es möglich, Veränderungen bei Grundannahmen (z.B. Kunden, Preise, Steuern etc.) vorzunehmen. Darüber hinaus können auch zwei Modelle mit demselben Gegenstand aber unterschiedlicher Struktur auf diese Weise bei gleichen Annahmen verglichen werden.

Bei der Profitanalyse ist zu beachten, dass ein Modell nur Sinn macht, wenn *jeder* der beteiligten Akteure Profit bzw. Nutzen erzielt, da

ansonsten eine Mitwirkung dieses Akteurs aufgrund der anfangs getroffenen Annahme der Profit- / Nutzenorientierung auszuschließen ist.

2.2.2.4 Das E3-Value-Metamodell

Den e³-Value-Modellen liegt ein Metamodell zugrunde, das nach Gordijn in Abbildung 2.29 wiedergegeben wird.

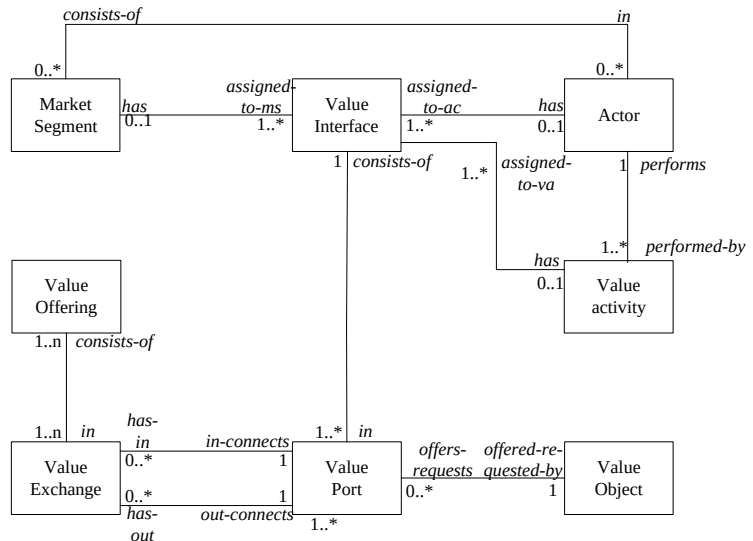


Abbildung 2.29: E³-Value-Metamodell / Ontologie

Ein konkretes Beispiel für die Anwendung der e³-Value-Modellierung folgt später (siehe auch Abbildung 2.35 auf Seite 101).

2.2.3 Analyse der E-Business-Modellierungsmethoden

E-Business-Modellierung stellt eine Möglichkeit der Integration betriebswirtschaftlicher Aspekte des E-Business und des Softwareentwurfs dar. Denn die hohe Komplexität von modernen E-Business-Anwendungen verlangt ganzheitliche Ansätze. Durch die Aufteilung der Gesamtaufgabe auf mehrere Sichten, Ebenen und Diagramme als Kennzeichen aller betrachteten Frameworks, wird der Entwurf von E-Business-Anwendungen, das E-Business-Engineering, erheblich überschaubarer. Gleichzeitig wird man den unterschiedlichen Bedürfnissen verschiedener Benutzergruppen gerecht.

Alle betrachteten Frameworks eignen sich zur Modellierung und Umsetzung großer Systeme. Trotzdem können sie auch für kleinere E-Business-Systeme sinnvoll genutzt werden, indem auf bestimmte Schritte

bewusst verzichtet wird. Meist wird von einem Top-Down-Vorgehen ausgegangen. Dies kann auf die gemeinsame Philosophie bei allen Techniken zurückgeführt werden, nach der zunächst das Top-Management die grobe Geschäftsstrategie festlegt, die in den nachfolgenden Stufen verfeinert und weiter ausgeführt wird.

Beim *RUP* wird die E-Business-Modellierung kaum unterstützt, da die Entwicklung eines Softwareprodukts im Mittelpunkt steht. Wiederverwendbarkeit von Komponenten und eine Beschränkung auf die UML-Standarddiagramme stehen im Vordergrund. Eine Erweiterung mit stärkerer Unterstützung der E-Business-Modellierung, unter Einbeziehung der betriebswirtschaftlichen Aspekte, ist aber vorstellbar. Insgesamt ist RUP sehr ausgereift und hat dies durch eine Vielzahl erfolgreich durchgeführter Projekte bewiesen.

Durch den RUP kann die E-Business-Modellierung gut in den praktischen Softwareentwurfsprozess eingeordnet werden.

E³-Value unterstützt die Modellierung auf eine praktische Art und Weise. Die *e³-Value*-Modelle sind einfach und schnell entworfen und übersichtlich. Sie sind daher besonders nützlich, um einen groben Überblick über E-Business-Szenarien und deren Lösungen zu bekommen. Die Methode benötigt die wenigsten Stufen aller untersuchten Methoden. Sämtliche geschäftlichen Aspekte werden in einem Diagramm erfasst und bilden die E-Business-Modellierungsstufe. Darüber hinaus werden nur noch die Stufen E-Business-Prozess und E-Software-Architektur unterschieden. Möchte man aber mit Hilfe der ‚Use-Case-Maps‘ den Übergang zur nächsten Stufe, der Prozesssicht (und nachfolgend zum Softwareentwurf), bewältigen, wirken die Modelle dadurch leider überladen. Hier erweist sich, dass eine weitreichende Integration von betriebswirtschaftlicher Sicht und Softwaretechnik eben doch mehrere Stufen und Diagramme benötigt, wie etwa bei RSD verwirklicht. *e³-Value* wurde bisher von Gordijn und seinen Mitarbeitern erfolgreich in praktischen Projekten angewendet. Es wird bisher lediglich durch eine Vorlage für Microsoft-Visio zum Zeichnen von Diagrammen unterstützt. Spezielle Tools zur Unterstützung der Methode sind nicht bekannt.

Die *E-BPMS*-Methode wird von der BOC-Unternehmensberatung in praktischen Projekten genutzt. E-BPMS ist methodisch nicht weit von *e³-Value* entfernt. Der Übergang vom E-Business-Modell zum Softwareprodukt wird aber methodisch besser durch entsprechende Zusatzdiagramme unterstützt.

Die Struktur der Modellierungsebenen in E-BPMS lässt sich gut mit den drei Sichten von *e³-Value* vergleichen und folgendermaßen zuordnen:

- E-Business-Modell-Sicht: Strategische Ebene

- E-Business-Prozess: Geschäftsebene
- E-Software-Architektur: Implementierungs- und Ausführungsebene

Das Softwaretool Adonis zeichnet sich durch seine hohe Flexibilität bei der Einbeziehung weiterer Tools zur Prozess- und Workflow-Modellierung aus. Außerdem bietet es vorbildliche Möglichkeiten zum Entwurf von Diagrammen und auch zu deren weiteren Verarbeitung (z. B. Simulation). Adonis kann derzeit als einziges Tool angesehen werden, das stabil und mächtig genug ist, um bei größeren Projekten eingesetzt zu werden.

Am weitestgehenden RSD versucht betriebswirtschaftliche und softwaretechnische Aspekte zusammenzubringen. Leider erscheint diese Integration für die praktische Einsetzbarkeit noch nicht ausgereift genug umgesetzt zu sein. Vermutlich wird deswegen bisher nicht über relevante RSD-Projekte aus der Praxis berichtet. Die Vielzahl der Diagramme und Stufen macht den Ansatz etwas unübersichtlich. Bei RSD wird im Unterschied zu e³-Value nicht nur die Übertragung von werthaltigen Objekten betrachtet, sondern alle Objekte, die zwischen Beteiligten einer E-Business Kooperation fließen können. Dies macht RSD mächtiger, aber auch etwas unübersichtlicher.

Die hierarchische Einteilung der ‚Cornerstones‘ bei RSD (siehe Abbildung 2.21) ist vergleichbar mit den drei Sichten, die von Gordijn bei e³-Value definiert werden, wobei die ‚Cornerstones‘ wie folgt den e³-Value-Sichten zugeordnet werden können:

- E-Business-Modell-Sicht: ‚Ambition&Scope‘, ‚Networked-Enterprise‘ und ‚Systems‘
- E-Business-Prozess: ‚Transaction-Scenarios‘ und ‚System-Components‘
- E-Software-Architektur: ‚Procedures‘ und ‚Protocols‘

Das Tool RSD Studio steht zum Entwurf der geschäftstechnischen Modelle zur Verfügung. Hervorzuheben ist vor allem das klar strukturierte Vorgehen, das alle Phasen des E-Business-Engineering gut unterstützt. Umfangreiche Informationen zur Methode sind frei verfügbar.

Die Elemente der graphischen Notation von e³-Value, E-BPMS und RSD lassen sich in folgender Weise vergleichen:

Tabelle 2.5: Vergleich der Modellelemente von e³-Value, E-BPMS und RSD

e ³ -Value	E-BPMS	RSD
Akteur	Geschäftsakteur	Akteur
Mehrwertaktivität	Prozessfragment	Funktion / Prozess / Action
‚Value-Interface‘	-	-

„Value-Port“	-	Verlauf / Connector / Port
Wertaustausch	Produkt-/Kapital-/Informationsfluss	Kanal / Verlauf
Wertobjekt	Produkt	Artefact
Wertangebot	-	-
Marktsegment	Abstrakter Geschäftsakteur	Mehrfach-Akteur
„Composite-Actor“	Kategorie von Geschäftsakteur	Geschachtelter Actor
-	Juristische Beziehung/ Aufbauorganisatorische Beziehung	-

Zusammenfassend werden e³-Value, E-BPMS und RSD anhand von charakterisierenden Kriterien der Methoden und ihrer unterstützenden Tools in Tabelle 2.6 auf Seite 80 verglichen.

Tabelle 2.6: Vergleich wichtiger Eigenschaften von e³-Value, E-BPMS und RSD

Kriterium	e ³ -Value	E-BPMS	RSD
Anschaulichkeit	Nur eine Diagrammform, sehr schnell zu erfassen	Mehrere Diagramme, überschaubar	Viele Diagramme, erfordert etwas Einarbeitung
Mächtigkeit	Unterstützung beschränkt sich auf die wesentlichen Aspekte der E-Business-Modellierung	Deckt gesamtes E-Business-Engineering ab, Fokus auf Prozessmodellierung	Sehr umfassende Lösung für gesamtes E-Business-Engineering
Integration	Kaum vorhanden, da Fokus auf Business-Modellierung	Gut, Hyperlinks zwischen verschiedenen Modellen	Fließender Übergang, Modelle werden mehrfach auf verschiedenen Modellierungsebenen verwendet
Toolunterstützung	nur Visio Schablonen	Adonis ist ein relativ mächtiges Tool	einfaches Tool
Simulationsmöglichkeiten	Gar nicht	Relativ umfangreich	Einfach formal
Speicherformat	nur Visio Diagramme	XML-Format	eigene Skriptsprache adl
Dokumentation	Gut, >20 verfügbare Paper	Nur wenige Paper verfügbar, da Methode von BOC vermarktet wird	Gut, >20 verfügbare Paper

E³-Value erweist sich als besonders geeignet für die spezialisierte Aufgabe der Modellierung von E-Business Geschäftsmodellen auf abstrakter konzeptioneller Ebene. RSD ist dagegen zur umfangreichen Modellierung bei E-Business Projekten bis hin zur Softwareerstellung zu favorisieren.

2.2.4 E-Business-Geschäftsprozessmodellierung

Die in einem Unternehmen und zwischen verschiedenen Unternehmen organisatorisch ablaufenden Prozesse können, wie schon oben erwähnt, mit Hilfe der Geschäftsprozessmodellierung erfasst werden. Die Methoden dieser Modellierung sollen nun näher vorgestellt werden.

Als weit verbreitete Methode finden die *ereignisgesteuerten Prozessketten* (EPKs) Verwendung. Sie sind ein anerkannter Standard für die Modellierung von Prozessen und finden beispielsweise im SAP R/3 Business Engineer Anwendung.

EPKs bauen im Wesentlichen auf Petri-Netzen auf und besitzen eine ähnliche Knoten-Kanten-Struktur. Darüber hinaus gibt es Versuche, EPK mit UML zu verknüpfen, wodurch eine implementationsnähere Modellierung möglich wird.

2.2.4.1 Elemente von EPK

EPK werden aus mehreren Elementen zusammengestellt. Dies sind Knoten, Kanten, Verknüpfungsoperatoren und Modellierungsregeln.

2.2.4.1.1 Knoten

Durch Knoten werden in einer EPK die relevanten Objekte und Tätigkeiten modelliert. Die wesentlichen Arten von Knoten sind:

- *Ereignisse* sind eines von zwei Grundelementen einer EPK. Ereignisse stellen Zustände dar, die eine bestimmte Konstellation repräsentieren. Üblicherweise hat ein solcher Zustand die Ausführung einer Aktivität zur Folge.
Ereignisse werden üblicherweise durch ein lang gezogenes Sechseck dargestellt.
- *Funktionen* stellen die für das Modell relevanten Aktivitäten dar. Solche Aktivitäten überführen grundsätzlich einen Eingangszustand (Ereignis) in einen Ausgangszustand (Ereignis), wobei zu berücksichtigen ist, dass Funktionen die einzigen explizit zeitverbrauchenden Elemente einer EPK sind.
Funktionen werden üblicherweise durch ein abgerundetes Rechteck dargestellt.

- *Organisationseinheiten* sind Personen, Stellen oder Abteilungen in einem Unternehmen. Durch die Repräsentation von Organisationseinheiten in einem Modell kann die Unternehmenshierarchie dargestellt werden. Organisationseinheiten werden einer oder mehreren Funktionen zugeordnet, mit deren Ausführung sie betraut sind. Organisationen werden üblicherweise durch Ellipsen dargestellt.
- *Funktionseinheiten* sind hingegen die Ressourcen, die bei der Ausführung der einzelnen Funktionen eine Rolle spielen. So werden sowohl die benötigten (verbrauchbaren) Ressourcen, beispielsweise Rohstoffe, als auch der Output einer Funktion, beispielsweise das erstellte Produkt, als Funktionseinheiten dargestellt. Funktionseinheiten werden üblicherweise als Rechtecke repräsentiert.

2.2.4.1.2 Kanten

Wie bereits erwähnt ermöglichen die Knoten keine Darstellung von Beziehungen. Daher dienen Kanten in erster Linie dazu, die verschiedenen Knoten zueinander in Beziehung zu setzen. In EPK werden im Wesentlichen drei Arten von Kanten anhand ihrer Funktionen unterschieden:

- *Kontrollfluss*: Diese Kanten beschreiben den zeitlich-logischen Ablauf und die Abhängigkeiten von Funktionen und Ereignissen. Kontrollflusskanten werden als durchgezogene, gerichtete Kanten (Pfeil) dargestellt. In einigen Quellen wird der Kontrollfluss durch gestrichelte gerichtete Kanten dargestellt, um eine Unterscheidung vom Informations- / Materialfluss zu erleichtern. Da der Kontrollfluss aber grundsätzlich zwischen Funktionen und Ereignissen verläuft, ist die Möglichkeit zur Unterscheidung auch mit der hier gewählten Darstellung als durchgehende, gerichtete Linie (Pfeil) gegeben.
- *Informations- / Materialfluss*: Diese Kanten sind üblicherweise eine Zuordnung von Funktionseinheiten zu Funktionen. Dargestellt als durchgezogene, gerichtete Kanten (Pfeil) stellen sie zusätzlich dar, ob eine Funktionseinheit als In- oder Output einer Funktion dient.
- *Organisationseinheitenzuordnung*: Diese Kanten ordnen Funktionen jene Organisationseinheiten zu, die für ihre Ausführung verantwortlich sind. Diese Zuordnung

wird als ungerichtete Kante (Linie) dargestellt.

2.2.4.1.3 Verknüpfungsoperatoren

Verknüpfungsoperatoren sollen die Möglichkeiten, die sich für die Modellierung des Kontrollflusses ergeben, erweitern. Sie bieten die Gelegenheit, parallele Abläufe, Alternativen oder Ausschlüsse zu formen. Grundsätzlich gibt es drei Arten von Verknüpfungsoperatoren:

- *AND-Operator*: der AND-Operator steht für die logische UND-Verknüpfung und wird durch einen Kreis mit dem logischen ‚und‘-Zeichen darin dargestellt.
- *OR-Operator*: der OR-Operator steht für die logische ODER-Verknüpfung und wird durch einen Kreis mit dem logischen ‚oder‘-Zeichen darin dargestellt.
- *XOR-Operator*: der XOR-Operator steht für die logische EXKLUSIV-ODER-Verknüpfung und wird durch einen Kreis mit einem diagonalen Kreuz darin dargestellt.

Je nachdem, welche Elemente durch die Operatoren verbunden sind, unterscheidet man zwischen Ereignisverknüpfung und Funktionsverknüpfung.

2.2.4.1.4 Modellierungsregeln

Die zentralen Regeln für die Modellierung mittels EPK sind:

- Der Beginn ist immer ein Ereignis.
- Das Ende ist immer ein Ereignis.
- Ereignisse und Funktionen tauchen im Wechsel auf.
- Die Bildung von Schleifen ist möglich.

2.2.4.2 EPK und UML

Die Möglichkeiten, die EPK mit den bisher vorgestellten Elementen bieten, sind völlig ausreichend um Geschäftsprozesse zu modellieren.

Im Kontext der Entwicklung von Softwaresystemen können EPK jedoch als Stufe zwischen Geschäftsmodell und Implementation/Architektur aufgefasst werden. Mit diesem Blickwinkel ist es nützlich, einen Schritt weiter zu gehen und die Nutzung von UML einzuführen, die es erlaubt das Prozessmodell etwas näher an die Architektur heranzurücken. Dies geschieht, indem eine objektorientierte Sichtweise eingenommen wird, ohne dadurch die Verständlichkeit zu beeinträchtigen. Im Gegenteil wird die Verständlichkeit und insbesondere die Kommunizier-

barkeit eines solchen Modells, in dem fachliche Objekte erstellt werden, durch den Einsatz von UML noch erhöht.

Es gibt mehrere Punkte an denen EPK mit UML verknüpft werden können. Im Folgenden sollen nur die wichtigsten aufgezeigt werden.

Einheiten → Klassendiagramme

Die erste Idee, die den objektorientierten Gedanken in die EPK einführen soll, ist die Darstellung von Objekten durch UML-Klassen. Dabei werden sowohl Operationseinheiten als auch Funktionseinheiten nicht mehr in der oben erwähnten Art dargestellt, sondern durch UML-Klassen, die es erlauben, den Einheiten auch Operationen und Attribute zuzuordnen.

Funktionen → Operationen

Die Nutzung von Operationen durch Funktionen baut auf dem im vorigen Abschnitt beschriebenen Gedanken auf. Durch die erfolgte Definition der verfügbaren Operationen eines Objektes ist es jetzt möglich, genau zu modellieren, welche der Operationen bei einer Funktion des Geschäftsprozesses benötigt werden

Funktionen → Attribute

In derselben Art und Weise wie die Operationen, können auch die Attribute eines Objektes jetzt den Funktionen zugewiesen werden. Dabei kann auch genau definiert werden, ob die Funktion lesend und / oder schreibend auf das Attribut zugreift. Obwohl direkt Bezug auf das Attribut genommen wird, wird implizit eine Zugriffsfunktion angenommen.

Operationen → EPKs

Die Hinterlegung von Operationen durch EPKs ist eine Möglichkeit, einen Prozessschritt auf übersichtliche Art und Weise weiter zu detaillieren. Dabei wird für die Operationen, die für ein Objekt definiert wurden, eine EPK erstellt, die den objektinternen Ablauf der Operation darstellt.

Die Modellierung von Geschäftsprozessen im E-Business gibt die Möglichkeit sowohl die unternehmensinternen als auch die interorganisationalen Abläufe die im Zusammenhang mit den Aufgaben im elektronischen Handel stehen, detailliert zu untersuchen, zu gestalten und entsprechende Lösungen und Systeme zu entwickeln.

Der notwendige Überblick und die grundlegenden Konzepte können mit Hilfe der weiter oben vorgestellten konzeptionellen E-Business-Modellierung von Geschäftsmodellen erreicht werden. Dies wird anschaulich im nächsten Abschnitt am Beispiel des E-Procurement gezeigt.

Die Modellierungsmethoden sind auch Grundlage für die Betrachtung speziellerer Funktionalitäten im E-Business, wie E-Business Verhandlung-

gen und Verhandlungskommunikation. Darauf wird weiter in Kapitel 4 eingegangen.

2.3 E-Procurement als typischer Bereich des E-Business

E-Procurement, auch als E-Purchasing oder Online-Beschaffung bekannt, befasst sich mit der elektronisch unterstützten Beschaffung in Unternehmen.

Der Bereich des E-Procurement dient als Anwendungsdomäne in dieser Arbeit, da sich in ihm vielfältige Elemente des E-Business wieder finden und er ebenso reichhaltige Potenziale für die Anwendung von Verhandlungen bietet. Dem E-Procurement kommt auch aus betriebswirtschaftlicher Sicht im Hinblick auf die dadurch mögliche Optimierung der Prozesskosten in der Beschaffungslogistik besondere Aufmerksamkeit zu (Losbichler, et al., 2002; Schneider, et al., 2000).

Zunächst wird allgemein die Beschaffung in Unternehmen betrachtet, danach die elektronische Beschaffung charakterisiert und dann eine grundlegende Struktur für E-Procurement-Systeme erarbeitet. Unter Anleitung des Autors wurde ein prototypisches E-Procurement-System entwickelt (Böge, 2003), auf das später noch im Detail eingegangen wird (siehe auch Kapitel 6). Es folgt die Betrachtung von Modellierungsmethoden für E-Procurement und die Diskussion von Verhandlungsaspekten im E-Procurement.

2.3.1 Beschaffung in Unternehmen

Unternehmen müssen laufend Produkte für den eigenen Konsum beschaffen. Sie konsumieren diese Produkte (Verbrauchsgüter) oder nutzen sie im weitesten Sinne zum Aufbau der Unternehmensinfrastruktur (Gebrauchsgüter). Die Beschaffung der Güter wird durch ihre besonderen Charakteristika hinsichtlich der Art ihres Einsatzes beeinflusst.

Gemäß ihrem Einsatz wird zwischen direkten und indirekten Produkten, Materialien bzw. Gütern unterschieden (vgl. (Dolmetsch, 2000)).

Direkte Produkte werden im Produktionsprozess zu einem Endprodukt verarbeitet oder sind Teil eines Endproduktes. Beispiele wären dafür Rohöl in einer Raffinerie oder Reifen oder Karosserieteile in einer Fahrzeugfabrik. Diese Güter besitzen im Unternehmen eine hohe strategische Bedeutung. Durch ihr Fehlen kann die gesamte Produktion zum Erliegen kommen. Sie werden meist regelmäßig von denselben Lieferanten bezo-

gen. Häufig handelt das beschaffende Unternehmen auch Rahmenverträge aus oder geht sogar Planungspartnerschaften mit den Lieferanten ein. Aufgrund der hohen strategischen Bedeutung ist die Beschaffung und Disposition von direkten Materialien meist in ein ‚Enterprise-Resource-Planing-System‘ integriert.

Zu den *indirekten Produkte* zählen hingegen alle Waren, die nicht in die Produktion einfließen. Dazu gehören Produkte für den administrativen Bereich, wie beispielsweise Büromaterialien, Büroausstattung oder andere nicht produktionsrelevante Dienstleistungen. Eine weitere Gruppe der indirekten Materialien sind die so genannten MRO-Produkte (Maintenance, Repair, Operations). Sie werden von Unternehmen für die Instandhaltung, die Wartung, die Reparatur oder den Betrieb von Maschinen benötigt. Ebenfalls zu dieser Gruppe gehören Produkte für Forschung und Entwicklung, wie z.B. Mess- und Zusatzgeräte oder Laborbedarf. Auch die Produktionsanlagen selbst werden dazugezählt. Bei MRO-Produkten handelt es sich meist um einen branchenspezifischen, produktionsnahen Bedarf. Indirekte Materialien haben meist einen relativ geringen Warenwert und werden in großer Menge eingekauft. Ein Bedarf an diesen Waren kann bei jedem Mitarbeiter des Unternehmens, der damit zum Bedarfsträger wird, anfallen.

Die folgende Tabelle gibt eine Übersicht über die wichtigsten Unterschiede bei der Beschaffung von direkten und indirekten Materialien:

Tabelle 2.7: Beschaffung indirekter und direkter Produkte (vgl. (Dolmetsch, 2000))

Beschaffung indirekter Produkte	Beschaffung direkter Produkte
Verschiedenste Produkte	Fest definiertes Produktionsmaterial
Standardprodukte	Häufig speziell entwickeltes Material
Bedarfsträger ist jeder Mitarbeiter	Beschaffungsanforderung von dediziertem Planer oder Beschaffer
Teilweise Genehmigung notwendig	Keine Genehmigung notwendig
Katalogeinkauf	Stücklistenauflösung

Aufgrund der unterschiedlichen Anforderungen bei der Beschaffung von indirekten und direkten Produkten sind auch die Beschaffungsvorgänge unterschiedlich. Somit ist auch eine jeweils adäquate elektronische Unterstützung notwendig.

2.3.2 Elektronische Beschaffung

Die elektronische Beschaffung mit Hilfe von Web- und Informationssystemen weist Charakteristika auf, die im Weiteren diskutiert werden. Diese Systeme müssen auf die möglichen Ausprägungen des Geschäftsmodells E-Procurement angepasst sein.

2.3.2.1 Beschaffung mit ERP-Systemen

Direkte Produkte werden üblicherweise mit Unterstützung durch ‚Enterprise-Resource-Planing-Systemen‘ (ERP-Systemen) beschafft. Durch ihre hohe strategische Bedeutung für das Unternehmen, werden diese Beschaffungsanforderungen schon seit langer Zeit in ERP-Systemen berücksichtigt.

Unter ‚Enterprise-Resource-Planing‘ versteht man die umfassende Steuerung, Kontrolle und Integration wichtiger Unternehmensabläufe. Bei ERP-Anwendungssystemen handelt es sich meist um modulare Softwarelösungen, die über entsprechende Schnittstellen zu einem Gesamtsystem integriert werden. Auf diese Weise soll eine Unterstützung des Unternehmens in wichtigen Bereichen, wie beispielsweise in der Finanz-, Personal- oder Produktionsplanung, ermöglichen werden. ERP-Systeme lassen sich auch als bereichsübergreifende Standardsoftware-Lösungen zur Steuerung, Optimierung und Ausführung von Geschäftsprozessen charakterisieren.

Das zurzeit wohl bekannteste ERP-System stammt von der SAP AG aus Walldorf. R/3 ist der Nachfolger des noch ausschließlich großrechnerbasierten R/2 und wurde 1992 eingeführt. Die nachfolgende Auflistung der Teilkomponenten soll einen Eindruck vom Umfang dieser Software vermitteln. SAP R/3 umfasst derzeit folgende Komponenten, die auch als Module bezeichnet werden:

- *Projektsystem (PS)*: Unterstützung von Planung, Kontrolle und Überwachung langfristiger Projekte,
- *Personalwirtschaft (HR)*: vollständig integriertes System zur Unterstützung der Planung und Kontrolle von Personalaktivitäten,
- *Instandhaltung (PM)*: unterstützt die Planung, Verarbeitung und Ausführung von Instandhaltungsaufgaben,
- *Produktionsplanung (PP)*: Planung und Kontrolle der Herstellungsaktivitäten eines Unternehmens innerhalb der täglichen Geschäftsabläufe,
- *Vertrieb (SD)*: hilft, alle Aufgaben und Aktivitäten zu optimieren, die beim Verkauf, bei der Lieferung und bei der Fakturierung anfallen,

- *Finanzwesen* (FI): beinhaltet u.a. die automatische Hauptbuchführung, Reporting sowie Debitoren- und Kreditorenbuchhaltung,
- *Controlling* (CO): internes Rechnungswesen (Kostenarten-, Kostenstellen-, Profit-Center-, Prozesskostenrechnung sowie Produktkalkulation und Ergebnis- & Marktsegmentrechnung),
- *Materialwirtschaft* (MM): Beschaffungs- und Lagerhaltungsunterstützung für Verkauf, Lieferung und Fakturierung.

Ein wichtiges Element in einem ERP-System ist die Verwaltung von Materialstammsätzen. Diese Datensätze beschreiben Produkte aus Sicht der verschiedenen Abteilungen (z.B. Einkauf, Disposition, Lagerhaltung). Produkte, die über einen solchen Materialstamm verfügen, werden als *kodierte Produkte* (oder als *Material*) bezeichnet. Direkte Produkte sind aufgrund ihrer strategischen Bedeutung für das Unternehmen und ihrem regelmäßigen Verbrauch kodiert.

Der Datenerfassungs- und Datenpflegeaufwand für kodierte Produkte ist hoch. Die Menge der indirekten Produkte ist sehr umfangreich und schnelllebig, so dass auf das Einpflegen dieser Produkte in ein ERP-System häufig oder zum Großteil verzichtet wird. So stehen bei der Beschaffung von indirekten Produkten oft keine ausreichenden elektronischen Unterstützungsfunktionen zur Verfügung. Entsprechend hoch sind die Prozesskosten in diesem Bereich.

2.3.2.2 Elektronische Beschaffung indirekter Produkte

Um eine elektronische Unterstützung auch bei der Beschaffung indirekter Produkte zu gewährleisten, wäre der Einsatz bestehender ERP-Systeme nahe liegend. Jedoch weisen ERP-Systeme zwei wichtige Nachteile beim Einsatz zur Beschaffung indirekter Güter auf:

- *Hohe Komplexität*: Durch ihre Mächtigkeit sind ERP-Systeme sehr komplex und erfordern ein hohes Maß an Systemkenntnis. Sie können im Allgemeinen nur von Mitarbeitern produktiv eingesetzt werden, die sich regelmäßig damit befassen. Beispielsweise von Mitarbeitern aus der Einkaufsabteilung eines Unternehmens.
- *Aufwendige Stammdatenpflege*: Jedem Produkt ist in einem ERP-System ein Materialstammsatz zugeordnet. Da die Produktpalette der indirekten Güter sehr umfangreich sein kann und sich auch oft ändert, wäre der Datenpflegeaufwand sehr groß.

So ist die Bestellung von indirekten Materialien häufig die letzte Bastion des *papiergestützten Einkaufs*. Da sich aber in diesem Bereich Produkte, Preise und Anbieter häufig ändern, treten viele Probleme und hohe Kosten bei dieser konventionellen Art der Beschaffung indirekter Güter auf:

- Die hohe Anzahl von Lieferanten erschwert die Suche nach der optimalen Einkaufsquelle,
- Vorhandene Kataloge sind oftmals veraltet und verzögern so die Artikelauswahl,
- Zu viele Produktnummern und Varianten erschweren die Auswahl eines geeigneten Produktes,
- Die Suche in verschiedenen Katalogen sowie die Ermittlung des aktuellen Preises ist mühsam und zeitintensiv,
- Produkt- und Preisvergleiche sind aufwendig,
- Notwendige, interne Genehmigungsprozesse belasten den Einkauf und binden dessen Ressourcen,
- Bestellprozesse sind in ihrer Gesamtheit für den Einkauf schwer zu überblicken.

Somit liegen die Prozesskosten eines Beschaffungsvorganges meist weit über dem Materialwert der beschafften, indirekten Produkte.

Die Entwicklung einer elektronischen Unterstützung für die Beschaffung indirekter Produkte, also E-Procurement, ist aus betriebswirtschaftlicher Sicht sehr wichtig.

Weitere Chancen bietet E-Procurement durch die Vermeidung von Medienbrüchen während des gesamten Beschaffungsprozesses. Da umständliche Mehrfacheingaben von Daten in das System entfallen, wird eine Beschleunigung dieses Prozesses und eine Kostenreduktion erreicht.

Durch E-Procurement-Systeme können die Beschaffungsprozesse im Unternehmen dezentralisiert werden und somit den Einkauf entlasten. Dabei ist durch die automatischen Auswertungen trotzdem ein guter Überblick für die Einkaufsabteilung garantiert.

2.3.2.3 Ausprägungen des E-Procurement

E-Procurement ist ein Geschäftsmodell, das Prozesse und Beziehungen zwischen Geschäftspartnern, Kunden und Mitarbeitern eines Unternehmens beschreibt, die durch elektronische Medien, insbesondere durch das Internet, unterstützt werden (Abbildung 2.30).

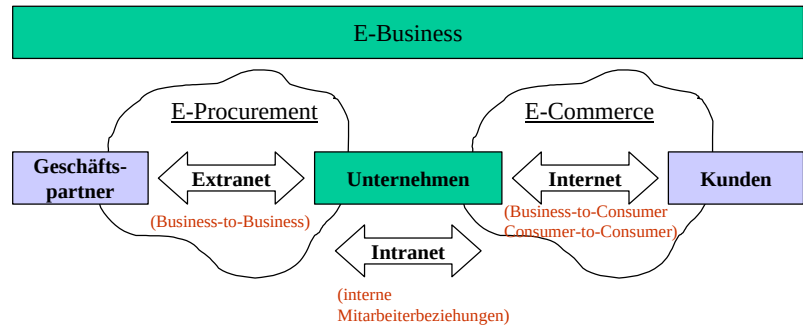


Abbildung 2.30: Einordnung E-Procurement

Je nachdem in welcher Konstellation Käufer, Verkäufer und gegebenenfalls Mittlern stehen, sind verschiedene Ausprägungen und Szenarien des E-Procurement möglich. Eine große Auswahl ergibt sich aus der Art des Umgangs mit den Produktkatalogen. Diese Kataloge fassen das Angebot der Beschaffung zusammen. Abhängig davon, wer die Speicherung und Pflege dieser Produktkataloge übernimmt, kann folgende Kategorisierung der elektronischen Beschaffungssysteme vorgenommen werden:

- Anbieterzentriert,
- Käuferzentriert,
- ‚Broker‘/Marktplatz.

2.3.2.3.1 Anbieterzentrierte Beschaffungssysteme

E-Procurement-Systeme, die den Anbieter in das Zentrum des Beschaffungsszenarios stellen und die Käufer um diesen ‚Hub‘ gruppieren, werden auch als ‚Seller-Centric‘ bezeichnet (Abbildung 2.31). Das Procurement-System und der Produktkatalog befinden sich hier beim Anbieter. Über das Internet können die Kunden durch das Produktsortiment navigieren und Käufe tätigen.

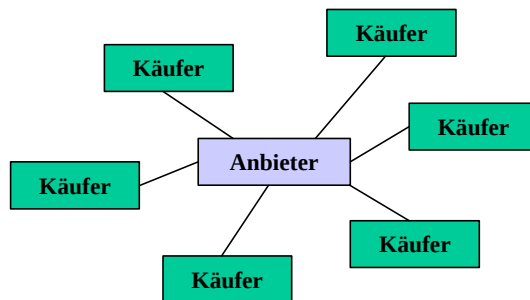


Abbildung 2.31: Anbieterzentriertes Beschaffungsmodell

Im Bereich Business-to-Consumer-E-Commerce ist dieses Modell der Standard. Diverse Webshops wie Amazon.com oder Quelle.de sind nach diesem Modell aufgebaut.

Für den Käufer bietet diese sehr flexible Variante den Vorteil der geringsten Investitionskosten. Er behält die volle Kontrolle über die Funktionalität und damit verbundenen Kosten des Systems.

Nachteilig sind für den Käufer die eingeschränkten Möglichkeiten. Meist sind nur Produktsuche und Bestellungen möglich. Den Bestellprozess des Käufers unterstützt das Anbietersystem in der Regel nicht.

Außerdem führt jeder Anbieter nur einen, nämlich seinen eigenen Produktkatalog. Will der Käufer zwischen mehreren Anbietern auswählen, so ist er gezwungen, sich mit den unterschiedlichen Systemen der verschiedenen Anbieter vertraut zu machen. Dies kann viel Zeit in Anspruch nehmen. Ein Preisvergleich zwischen Anbietern gestaltet sich dementsprechend sehr schwierig.

Beispiele für diese Art der Beschaffung findet man z.B. bei Dell (www.dell.com) und Cisco. (www.cisco.com)

2.3.2.3.2 Käuferzentrierte Beschaffungssysteme

In einem Käufer zentrierten („Seller-Centric“) E-Procurement-System werden die Produktkataloge verschiedener Anbieter in einem aggregierten Universalkatalog zusammengefasst (vgl. Abbildung 2.32). Unternehmensmitarbeiter können von ihrem Computer aus über ein Benutzer-Interface (z.B. dem Webbrowser) Einsicht in diesen Gesamtkatalog nehmen und Bestellungen tätigen.

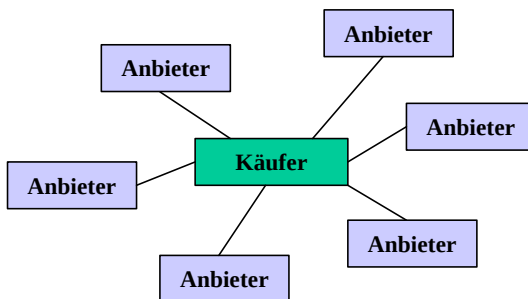


Abbildung 2.32: Käuferzentriertes Beschaffungsmodell

Bei dieser Herangehensweise hat das Unternehmen zum einen Kontrolle über den Produktkatalog und zum anderen Daten und Prozesse über den Beschaffungsvorgang. Dies zieht nach sich, dass das E-Procurement-System leichter mit anderen Komponenten der Unternehmensorganisation (z.B. Finanzmanagement) kommunizieren kann. Es unterstützt die Zielsetzung der Prozesskostenreduzierung des Systems. Das System kann

leichter erweitert werden, da die Kontrolle beim Käufer liegt. Eine weitere Vereinfachung durch einen aggregierten Katalog besteht weiterhin bei der Produktsuche und –auswahl durch die Mitarbeiter des Unternehmens, die nun in der Lage sind, einen Preisvergleich einzelner Produkte zwischen verschiedenen Anbietern vorzunehmen. Obwohl der Katalog auf Käuferseite gepflegt wird, kann dem Kataloganbieter über ein Extranet generell ein partieller Zugriff zur Aktualisierung der Produktdaten, -preise und ihrer Verfügbarkeit angeboten werden. Ein Nachteil dieser Variante ist der erhöhte Kostenaufwand für den Käufer, der durch die Systembereitstellung und -wartung entsteht. Ariba, Commerce-One und Netscape bieten kommerzielle Systeme an, die auf diesem Modellansatz aufbauen.

2.3.2.3.3 Marktplatzorientierte Beschaffungssysteme

Im ‚Broker‘-/Marktplatzmodell wird die Beschaffungsfunktionalität in Form von Produktkatalogen und Bestellmöglichkeit von einem Drittanbieter bereitgestellt (Abbildung 2.33). Dieser Anbieter wird ‚Marketplace-Provider‘ genannt und verwaltet die Produktkataloge selbst. Er stellt Transaktionssysteme zur Verfügung, die es Anbietern und Käufern ermöglichen miteinander in Kontakt zu treten und Geschäfte abzuwickeln. Als Ziel wird die Schaffung einer möglichst großen, internetbasierten Handelsumgebung mit standardisierten Regeln zur Handelsabwicklung angestrebt.

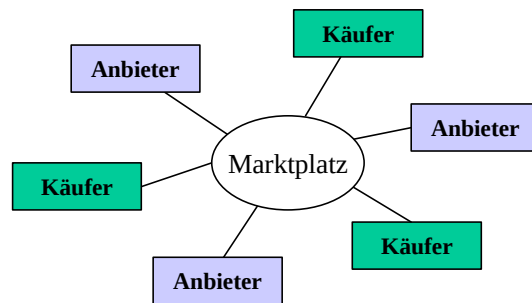


Abbildung 2.33: Marktplatz/,Broker‘ als Mittler der Akteure

Diese Variante des E-Procurement eignet sich besonders für Unternehmen, die gleichzeitig als große Anbieter und Käufer in ihrem Marktsegment auftreten. Ihr Vorteil besteht nämlich darin, dass über den Marktplatz Verbindung zu vielen Abnehmer und Anbietern bestehen. Außerdem erleichtert die Spezialisierung des 'marketplace provider' erweiterte Funktionen wie etwa Preisverhandlungen und Auktionen.

Die Etablierung und der Betrieb dieses Geschäftsmodells stellt jedoch eine besondere Herausforderung hinsichtlich des Erfolges, der darin besteht, die Interessen der Verkäufer und Käufer ausgewogen zu beachten.

2.3.3 Systeme für die elektronische Beschaffung

Insbesondere bei mittleren und großen Unternehmen geht der Trend dahin, elektronische Beschaffungssysteme, also E-Procurement-Systeme, die Desktop-Purchasing-Systeme (DPS, DP-Systeme) genannt werden, einzusetzen. Es handelt sich dabei um die Umsetzung eines käuferzentrierten Modells der elektronischen Beschaffung.

Aufgrund einer hohen Anzahl an Beschaffungsvorgängen für indirekte Güter ist die Einführung solcher Systeme lohnend. Sie führen zu erheblichen Prozesskosteneinsparungen. Durch die DPS können die Produktkataloge unternehmensintern gepflegt werden, so dass eine individuelle Anpassung mittels des Beschaffungssystems vorgenommen werden kann. Zusätzlich können die Prozesse des Beschaffungsvorgangs auch mit anderen Teilsystemen des Unternehmens integriert werden. DP-Systeme sind intranetbasierte Systeme, die an die bestehende, betriebswirtschaftliche Standardsoftware des beschaffenden Unternehmens angekoppelt werden. Sie ermöglichen eine dezentrale Bestellabwicklung für die Mitarbeiter des Unternehmens. Ferner umfassen sie das Content-Management, das aus dem Import und der Aktualisierung von Produktkatalogen in unterschiedlichen Formaten besteht, und halten Funktionen für den Datenaustausch mit den Lieferantensystemen bereit.

2.3.3.1 Architekturen

In der Praxis sind verschiedene Ansätze für DP-Systeme vorhanden, die sich hauptsächlich im Umfang und dem Ausmaß der Integrationsfähigkeit in bestehende ERP-Systeme unterscheiden. Nenninger stellt in einer Marktübersicht der KPMG Unternehmensberatung (Nenninger, 1999) wichtige kommerzielle Systeme vor.

Im Folgenden werden die häufigsten Typen von DP-Systemen vorgestellt.

Procurement-Based: Diese Art der Procurement-Systeme ist als Vollprodukt konzipiert, d. h. sie decken eine große Anzahl von Funktionen ab, die zur Beschaffungsunterstützung notwendig sind. Dazu zählen beispielsweise Komponenten zur Abbildung des Workflows eines Bestellvorganges.

Dieser Typ bietet nur eine partielle Integration in bestehende ERP-Systeme, z.B. bei der Rechnungslegung. Die Katalogintegration ist nicht

der Hauptaspekt dieser Systeme und muss deshalb manuell erfolgen oder durch Drittanbieter übernommen werden. (siehe unten: Content Provider)

ERP-Driven: Diese Art von Systemen heben sich durch ihre Integrationsfähigkeit in ein bestehendes ERP-System hervor. Dabei werden wichtige Komponenten wie z.B. Workflow, Administration und Rechnungslegung über die entsprechenden Module des ERP-Systems abgewickelt. Diese intensive Bindung hat auch Nachteile, da sie eine aufwendige Stammdatenpflege erfordert.

Catalog-Provider: Als Catalog-Provider bezeichnet man Drittanbieter, die sich auf das Produktkatalogmanagement spezialisiert haben und diese Dienstleistung E-Procurement-Anwendern anbieten. Ein Catalog-Provider erhält Kataloge in verschiedenen Formaten von Lieferanten, übernimmt die Katalogintegration dieser heterogenen Daten und sorgt für die regelmäßige Aktualisierung. Der Procurement-Betreiber selbst ist von dieser Aufgabe befreit und erhält den fertigen Gesamtkatalog. Diese Variante kann auch als Ergänzung für Systeme, die auf E-Procurement basiert sind, angesehen werden.

Buy & Sell-Side: Einige Unternehmen (z.B. Netscape) bieten zweigeteilte Procurementlösungen an. Diese bestehen aus einer Lieferantenkomponente und einer dazugehörigen Komponente für die Beschaffungsseite. Da beide Teile von demselben Anbieter stammen, ist ein sehr einfacher, fest vorgegebener Datenabgleich zwischen Beschaffer und Lieferant möglich. Es entstehen jedoch höhere Lizenz- und Einführungskosten für die teilnehmenden Seiten.

2.3.3.2 Komponenten

Ein Desktop-Purchasing-System muss zwei Hauptaufgaben erfüllen können:

Zum einen muss es eine *umfassende Funktionalität zur Unterstützung von Beschaffungsvorgängen* bereitstellen, die an die unterschiedlichen Bestellprozessabläufe individuell angepasst werden können.

Zum anderen müssen diese Funktionen den *einzelnen Bedarfsträgern* (dies sind die Mitarbeiter des Unternehmens) *in einer einheitlichen, ergonomischen Form dezentral zugänglich gemacht* werden.

Um diese Anforderungen zu erfüllen bietet sich die Abstraktion des DP-Systems in die folgenden Komponenten an, die auch in Abbildung 2.34 im Kontext dargestellt werden:

- Multi-Lieferanten-Katalog
- Benutzer-Front-End

- Workflow-Komponente
- Administration, ERP-Integration & Reporting



Abbildung 2.34: Komponentenübersicht eines Desktop-Purchasing-Systems (Nenninger, 1999)

2.3.3.2.1 Multi-Lieferantenkatalog

Der Multi-Lieferantenkatalog bildet die Basis des gesamten Beschaffungssystems. Bei diesem Katalog handelt es sich um eine tatsächliche oder logische Verbindung von Katalogen verschiedener Lieferanten, deren Artikel in elektronischer Form beschrieben und gespeichert werden (siehe unten: Katalogorganisation). Die zentrale Datenhaltung des Katalogs ermöglicht dem Unternehmen, die Kontrolle über das Einkaufsverhalten zu behalten. Des Weiteren kann durch die Vorauswahl von Kataloganbietern die Gruppe der Lieferanten im Vorfeld selektiert werden. Dabei gibt es zwei wichtige Unterscheidungskriterien für den Umgang mit Lieferantenkatalogen:

- Funktionelle Katalogorganisation
- Katalogwartung

Unter *funktioneller Katalogorganisation*, die statisch ist, wird die logische Organisation der Produktdaten aus den einzelnen Lieferantenkatalogen innerhalb des Multi-Lieferantenkataloges verstanden.

Eine erste Variante dieser Organisation basiert auf dem Vorgehen, die Produktdaten der einzelnen Kataloge in eine gemeinsame, hierarchisch gegliederte Produkthierarchie einzubetten. Dadurch erscheinen gleiche oder ähnliche Artikel in den gleichen Produktgruppen des Gesamtkataloges und sind so besser vergleichbar. Der Bedarfsträger erhält dadurch eine

sehr gute Übersicht über das vorhandene Angebot und kann leicht Preise und Leistungen vergleichen. Die Markttransparenz ist in diesem Fall sehr hoch. Damit eine übergreifende Gliederung der verschiedenen Artikel möglich ist, müssen diese anhand eines gemeinsamen Schemas klassifiziert werden. Dies geschieht, indem die von den Lieferanten angegebenen Artikelbezeichnungen vom System um einen unabhängigen Materialklassifikationsschlüssel erweitert werden. Beispiele für solche Klassifizierungsstandards sind UN/SPSC (,United Nations' / ,Standard Products and Services Classification'), EAN (Europäische Artikel Nummerierung) und eCl@ss (,Electronic Classification'). Bis auf die EAN sind diese Standards hierarchisch aufgebaut. Dies ist sinnvoll, da bei der Produktsuche oftmals nicht die exakte Artikelbeschreibung bekannt ist, und sich Suchende auf diese Weise an einer Produkthierarchie orientieren können. Hinsichtlich Integration und Wartung der Daten ist diese Variante der Katalogorganisation aufwändiger als der im Folgenden beschriebene zweite Ansatz.

In dieser Variante werden die Katalogdaten der verschiedenen Anbieter logisch getrennt gehalten. Jedes Katalogschema wird dem jeweiligen Anbieter zugänglich gemacht. Dadurch kann dieser selbst seinen Katalogabschnitt gestalten und warten. Dadurch verschlechtert sich zwar die Transparenz der Produkte für den einzelnen Unternehmensmitarbeiter, gleichzeitig sinkt aber der Wartungsaufwand, den das beschaffende Unternehmen zu leisten hat.

Bei der *Katalogwartung* stehen die *dynamischen* Attribute im Mittelpunkt der Betrachtung. Zu den Katalogdaten zählen neben Produktbeschreibung und Klassifizierungsnummern unter anderem auch Preis, Verfügbarkeit, Bezugsquellen und Artikelnummern. Einige dieser Attribute ändern sich ständig und müssen daher regelmäßig angepasst werden. Dies kann auf unterschiedliche Weise geschehen. Ausschlaggebend ist dabei jeweils, wer die Aktualisierungen vornimmt.

Der *Lieferant* kann die Aktualisierungen durchführen, wenn das beschaffende Unternehmen ihm Zugriff auf seinen Katalog gewährt. Dadurch wird Arbeitsaufwand auf den Lieferanten verlagert. Das beschaffende Unternehmen muss lediglich Kontrollaufgaben übernehmen, beispielsweise das Überprüfen der Einhaltung von eventuell bestehenden Rahmenverträgen.

Nimmt das *beschaffende Unternehmen* die Aktualisierungen selbst vor, dann ist dies besonders dann mit viel Aufwand verbunden, wenn die Anzahl der Lieferanten groß ist. Es besteht dabei die Gefahr, dass Ressourcen, die durch den Einsatz eines elektronischen Beschaffungssystems im Einkauf frei werden, nun für die Katalogpflege eingesetzt werden müssen. Als Gegenleistung erhält das Beschaffungsunternehmen dafür die Kontrolle über den Katalog.

Bei einem ‚Content-Broker‘ handelt es sich um ein Dienstleistungsunternehmen, das die Integration der Katalogdaten übernimmt. Dafür erhält es von den einzelnen Lieferanten die nicht standardisierten Produktdaten und stellt den aggregierten Katalog dem beschaffenden Unternehmen in dessen Intranet oder über ein Extranet zur Verfügung. Es entstehen dadurch zwar Kosten, im Gegenzug kann jedoch auf die Kompetenz eines spezialisierten Anbieters zurückgegriffen werden.

2.3.3.2.2 Benutzer-Front-End

Einen wichtigen Teil eines Electronic-Procurement-Systems stellt das Benutzer-Front-End dar. Hierbei handelt es sich um eine Web-Oberfläche, die von den einzelnen Mitarbeitern des Unternehmens dezentral über einen normalen Webbrowser abrufbar ist und die alle wichtigen Funktionen zur einfachen Durchführung einer Bestellung ermöglicht. Neben einer komplexen Funktionalität muss das Benutzer-Front-End aber auch einfach zu bedienen sein, da dies eine wichtige Voraussetzung für die Akzeptanz des gesamten Systems ist. Folgende grundlegende Funktionen sind dabei zu beachten:

Suchmaschine: Die Suchmaschine ist neben der hierarchischen Gliederung der Produktkategorien eine wichtige Hilfe, um das Finden von gesuchten Artikeln zu unterstützen. In der einfachsten Suchvariante gibt der Bedarfsträger einen Artikelnamen in eine Suchmaske ein und erhält als Antwort ein passendes Angebot. Da oftmals eine genaue Produktbezeichnung nicht bekannt ist oder variieren kann, müssen für eine effiziente Suche weitere Kriterien angegeben werden können. Dazu gehören Produktgruppen, ähnliche Produkte, Standardnummern oder Produktfunktionen.

Unstrukturierte Anfragen: Trotz Artikelbeschreibungen kann es sein, dass sich ein Beschaffer nicht sicher ist, ob ein ausgesuchter Artikel für die zu erledigende Aufgabe geeignet beziehungsweise passend ist. Dann sollte es möglich sein über das Front-End eine unstrukturierte Anfrage an den Lieferanten des Kataloges, aus dem der Artikel stammt, zu richten. Dazu eignet sich besonders das Ausfüllen eines Formulars, aus dem dann vom System automatisch eine Anfrage, beispielsweise eine Email, generiert und verschickt wird. Denkbar sind auch andere Formen der elektronischen Kooperation wie auch ein Produktforum, in dem Nachfragen beantwortet werden und so später auch von anderen Mitarbeitern eingesehen werden können.

Speicherbare Einkaufskörbe: Oftmals wiederholen sich Bestellungen bestimmter Artikel, beispielsweise wenn es darum geht, Kopiererpapier oder Druckerpatronen zu ersetzen. Ein wiederholter Bestellvorgang wird

dann beschleunigt, wenn ein bei einer vorhergehenden Bestellung zusammengestellter Warenkorb abgespeichert wurde und bei Folgebestellungen einfach wieder verwendet werden kann.

Verfügbarkeitsprüfung: Bei zeitkritischen Bestellungen ist es wichtig, dass der Beschaffer bereits vor Bestellung in Erfahrung bringen kann, ob ein bestimmter Artikel beim Lieferanten vorrätig ist und er mit einer rechtzeitigen Lieferung rechnen kann. Eine solche Verfügbarkeitsprüfung kann auf drei Weisen, die sich nach der Aktualität der Daten unterscheiden, erfolgen:

- Die *echtzeit-basierte Nachfrage* liefert aktuelle Daten. Hier wird eine Verfügbarkeitsanfrage vom Lieferanten direkt anhand einer Datenbankanfrage an den eigenen Lagerbestand beantwortet. Es entsteht jedoch ein hoher Datenfluss.
- Bei der *batch-basierten Anfrage* übermittelt der Lieferant die Verfügbarkeitsdaten in regelmäßigen Abständen. Der Datenfluss ist hier gering. Allerdings ist diese Variante der Echtzeitanfrage in der Aktualität unterlegen.
- Eine dritte Möglichkeit ist die *message-basierte Verfügbarkeitsnachfrage*. Dabei werden Anfragen nachrichtenbasiert, etwa per Email, an den Lieferanten geschickt. Nachteilig ist hier, dass der anfragende Mitarbeiter nicht in jedem Fall mit einer schnellen Antwort beziehungsweise mit konstanten Antwortzeiten rechnen kann. Ein Vorteil ist, dass der Lieferant die Möglichkeit hat auf die Verfügbarkeitsnachfrage des Beschaffers auch intern zu reagieren und beispielsweise für einen wichtigen Auftrag die Verfügbarkeit eines Produktes bedarfsorientiert erhöhen kann.

2.3.3.2.3 Workflow-Komponente

Das wichtigste Ziel beim Einsatz eines Electronic-Procurement-Systems ist die Kostenreduzierung durch die Optimierung des Bestellprozesses. Dazu ist es notwendig, die Bestellvorgänge schneller abzuwickeln.

Zum einen erreicht man dies durch die Verringerung von Medienbrüchen, das heißt der gesamte Vorgang erfolgt durchgehend elektronisch, so dass beispielsweise niemand Daten in Papierform vorhalten muss. Zum anderen bewirkt man dies aber auch durch die Vereinfachung von Bestellvorgängen. Der Bestellwert vieler Bestellungen indirekter Güter ist sehr gering, so dass der Aufwand einer manuellen Überprüfung, z.B. durch einen Vorgesetzten oder den Einkauf, nicht sinnvoll ist. Eine Workflowkomponente dient dazu, diese einfachen Vorgänge vorzusortieren, so dass der automatische Ablauf nicht behindert wird. Dazu werden von der Administration so genannte Businessregeln („Business-Rules“) als Abbildung von Genehmigungsverfahren definiert. Diese Regeln legen

fest, welche Bestellvorgänge keine explizite Genehmigung benötigen, etwa weil der Bestellwert nur gering ist oder Bestellungen eines bestimmten Lieferanten präferiert werden sollen. Die Regeln legen auch fest, wie der Genehmigungsvorgang bei anderen Bestellungen ablaufen soll, d.h. welcher Vorgesetzte die Bestellung autorisieren muss. Businessregeln können z.B. auch Bestellbudgets für einzelne Abteilungen, Produktgruppen oder Lieferanten festlegen. Dadurch kann der Einkauf das Bestellverhalten steuern, um beispielsweise mit bestimmten Lieferanten ausgehandelte Rahmenverträge effektiv zu nutzen.

2.3.3.2.4 Administration, ERP-Integration & Reporting

Die Administrationskomponente ermöglicht entscheidungsbefugten Personen, wie dem Einkauf, das Einrichten und Steuern des Beschaffungssystems. Dazu gehören unter anderem das Anlegen von Benutzern des Systems und das Definieren von Businessregeln für den Ablauf von Bestellvorgängen.

Über die Administration wird des Weiteren die Kopplung an bestehende ERP-Systeme durchgeführt.

Das Reporting wird parallel zu den anderen Komponenten durchgeführt. Es besteht unter anderem aus einer Logging-Komponente, die über Lieferantenaktivitäten Buch führt und spätere Bewertung einzelner Lieferanten zulässt sowie einem Statistikmodul, das eine Auswertung des Beschaffungsverhaltens von Benutzungsgruppen ermöglicht, beinhaltet.

2.3.4 Modellierungsmethoden im E-Procurement

Zur tiefer gehenden detaillierten Beschreibung von E-Procurement-Lösungen werden die in Abschnitt 2.2 erarbeiteten Modellierungsmethoden auf die Anwendungsdomäne des E-Procurement beispielhaft angewandt. Dabei bewährte sich die Modellierung von Geschäftsmodellen mit der e³-Value Methode und die Modellierung von Geschäftsprozessen mit EPK (Crantz, 2002).

Als konkretes Szenario wird eine käuferzentrierte E-Procurement-Variante als exemplarisches Beispiel gewählt, bei der die Kataloge über einen Content Provider / ‚Broker‘ bezogen werden. In dieser komplexeren Variante mit mehreren Akteuren kommt die Modellierungsmethode deutlicher zum Tragen.

Unabhängig von konkreten Anforderungen aus realen Szenarien, gibt es viele Fälle zur Modellierung solcher Systeme, so hier eine Möglichkeit beispielhaft ausgewählt wurde. Sobald eine Modellierungsvariante vor-

liegt, können oft auch mit geringem Aufwand weitere Varianten entworfen und analysiert werden.

Der Detaillierungsgrad der Modelle ist im Folgenden auf einem mittleren Niveau gehalten, um eine übersichtliche und sinnvolle Diskussion zu ermöglichen.

Zunächst wird mit Hilfe der e³-Value-Modellierung ein mögliches E-Business-Modell spezifiziert und darauf folgend die internen Geschäftsprozesse beispielhaft dargestellt.

2.3.4.1 E-Business-Modellierung für E-Procurement

Die gewählte Modellvariante umfasst auf der oberen Ebene drei Akteure in der e³-Value-Modellierung: die Anbieter, den ‚Broker‘ / ‚Content-Provider‘ und das beschaffende Unternehmen.

Bei den *Anbietern* handelt es sich um ein Market Segment. Die Aufgaben der Anbieter umfassen im Wesentlichen die Bereitstellung eines Kataloges für den ‚Broker‘ und die Lieferung der Waren.

Der ‚Broker‘ ist der Vermittler zwischen Unternehmen und Anbieter. Er ist dafür zuständig, die Kataloge der Anbieter zusammenzufassen und zur Verfügung zu stellen. Ferner leitet er die Bestellungen an die entsprechenden Anbieter weiter und erhält dafür eine Bezahlung. Alternative Varianten wären direkte Kontakte zwischen Anbieter und Unternehmen zur Bestellung oder Bezahlung.

Das *Unternehmen* ist als ‚Composite-Actor‘ modelliert. Es fasst die relevanten Unternehmensbereiche, von denen jeder als Akteur modelliert ist, zusammen. Diese einzelnen Akteure sind:

- *Bedarfsträger*: Bestellt die Ware und ist Endempfänger
- *Controlling*: Prüft eine Bestellung anhand bestimmter Kriterien auf ihre Zulässigkeit
- *Rechnungsabteilung*: Prüft die Rechnung und bezahlt sie
- *Wareneingang*: Nimmt Lieferungen (inkl. Lieferscheine) in Empfang, prüft sie und leitet sie weiter
- *E-Procurement(EP)-System*: Verwaltet Bestellungsdaten und stellt sie bei Bedarf zur Verfügung. Außerdem ist das EP-System für die Oberfläche und die Bereitstellung des Sammelkataloges zuständig.

Abbildung 2.35 zeigt das e³-Value-Modell der gewählten E-Procurement-Variante.

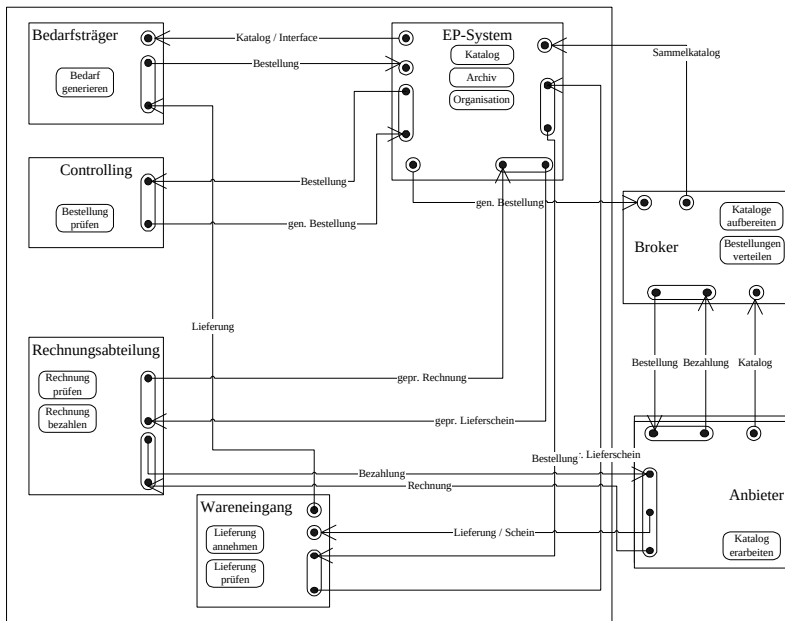


Abbildung 2.35: e³-Value-Modell E-Procurement

Ausgehend von den einzelnen Akteuren werden nun auch deren ‚Value-Activities‘, ‚Value-Ports‘ und Wertaustausche betrachtet.

Der *Bedarfsträger* ist eine initiale Instanz des Modells. Er ‚generiert‘ den Bedarf, d.h. beim Auftreten eines Bedarfs benutzt er den Katalog, der ihm vom E-Procurement-System zusammen mit der Benutzeroberfläche zur Verfügung gestellt wird. Mit Hilfe des Katalogs wählt er die gewünschten Waren aus. Als Ergebnis dieser Bestellung erhält er später die Lieferung mit den Waren. Bestellung und Lieferung sind eine nicht trennbare Einheit, weswegen sie in einem ‚Value-Interface‘ zusammengefasst werden.

Das *Controlling* ist ein Akteur, der vom E-Procurement-System eine aufgegebenen Bestellung vorgelegt bekommt. Nach welchen Kriterien entschieden wird, ob eine Bestellung genehmigungspflichtig ist, spielt an dieser Stelle keine Rolle. Nach Abschluss des Genehmigungsverfahrens wird die genehmigte Bestellung an das E-Procurement-System zurückgegeben.

Das *E-Procurement-System* ist zentraler Bestandteil des Modells. Es verwaltet den internen Katalog und wird darüber hinaus als Archiv für Bestellungen und Belege, wie Lieferscheine und Rechnungen, sowie für Fragen der Organisation, wie beispielsweise der Steuerung des Genehmigungsablaufs, genutzt. Aufgrund dieser vielfältigen Aufgabe ist es auch an einer großen Anzahl von Wertaustausche beteiligt. So erhält es den Sammelkatalog vom ‚Broker‘, den es gleichzeitig zusammen mit der

Benutzeroberfläche dem Benutzer als potentiellern Bedarfsträger zur Verfügung stellt. Bei einem Bestellvorgang nimmt es die Bestellung vom Bedarfsträger entgegen, leitet sie gegebenenfalls an die Controllingabteilung zur Genehmigung weiter und gibt die endgültige Bestellung schließlich an den ‚Broker‘ weiter. Die Archivfunktion besteht darin, dass Dokumente entgegengenommen und zur Verfügung gestellt werden. Beispiele für die Entgegennahme von Dokumenten wären ein geprüfter Lieferschein von der Warenannahme oder eine geprüfte Rechnung von der Rechnungsabteilung. Werden der Rechnungsabteilung Lieferscheine oder dem Wareneingang im Sinne einer Prüfungsunterstützung Bestellungen übermittelt, so stellt das E-Procurement-System Dokumente zur Verfügung.

Die *Rechnungsabteilung* in einem Unternehmen erhält die Rechnung für erfolgte Lieferungen und bezahlt diese nach einer Prüfung. Um erfolgte Lieferungsvorgänge nachvollziehen zu können, bezieht die Rechnungsabteilung den Lieferschein vom E-Procurement-System, an welches auch die geprüfte Rechnung zurückgegeben wird.

Der *Wareneingang* nimmt die gelieferte Ware samt Lieferschein vom Anbieter entgegen und prüft die Lieferung, indem auf die Bestellung im E-Procurement-System zurückgegriffen wird. Der geprüfte Lieferschein wird für die Rechnungsabteilung ins E-Procurement-System eingegeben und die Lieferung wird an den Endempfänger, den Bedarfsträger, weitergeleitet.

Der ‚Broker‘ nimmt von verschiedenen Anbietern Kataloge entgegen und bereitet diese in einem einheitlichen Format auf. Den Sammelkatalog stellt er durch das E-Procurement-System dem Unternehmen zur Verfügung. Vom E-Procurement-System seinerseits kann er Bestellungen erhalten, wobei es Aufgabe des ‚Broker‘ ist, die Bestellungen den entsprechenden Anbietern zuzuordnen. Für die Weiterleitung der Bestellungen erhält der ‚Broker‘ vom Anbieter eine Bezahlung. Weitere Erlösmodelle sind in Kombination mit alternativen Modellierungsmöglichkeiten denkbar.

Der *Anbieter* erarbeitet aus seinem Angebot einen Katalog, den er an den ‚Broker‘ weitergibt. Vom ‚Broker‘ nimmt der Anbieter dann die Bestellungen entgegen. Für die Abwicklung der Lieferung ist der Anbieter ebenfalls zuständig, d.h. er liefert die Ware inklusive Lieferschein an den Wareneingang, schickt die Rechnung an die Rechnungsabteilung und erhält dafür im Gegenzug seine Bezahlung.

Durch das Modellierungsmittel ‚Composite-Actor‘, werden alle zum Unternehmen gehörenden Akteure zusammengefasst. Die ‚Value-Ports‘, die mit externen Akteuren einen Wertaustausch durchführen, sind somit als Schnittstellen des Unternehmens zu betrachten. Für den Anbieter als

externen Akteur ist es beispielsweise nicht entscheidend, an welche Abteilung im Unternehmen er die Rechnung schickt.

Die E-Business-Modellierung erleichtert es, alternative Varianten eines Modells zu erkennen. Exemplarisch sollen im Folgenden Änderungsmöglichkeiten am Modell aufgezeigt werden.

Die *Sammlung und die Aufbereitung der Kataloge* kann vom Unternehmen selbst durchgeführt werden, wodurch der ‚Broker‘ überflüssig wird. Die Darstellung würde durch eine Erweiterung des Unternehmens erfolgen. Dies bedeutet, dass der ‚Broker‘ als Teil des Composite-Actors-Unternehmen betrachtet wird.

Die *Bezahlung eines externen ‚Broker‘* kann auf verschiedene Arten erfolgen. Beispielsweise durch das Unternehmen als Gegenleistung für die erhaltene Katalogsammlung. In diesem Fall würde ein Wertaustausch ‚Bezahlung‘ vom Unternehmen zum ‚Broker‘ existieren, welches in einem ‚Value-Interface‘ mit der Katalogbereitstellung gruppiert ist. Die Bezahlung des ‚Broker‘ kann auch durch den Anbieter erfolgen, sobald der Katalog des Anbieters aufgenommen wird. In diesem Falle wird eine Gruppierung mit dem Katalogaustausch erfolgen, der zwischen Anbieter und ‚Broker‘ stattfindet. Durch diese verschiedenen Modellierungen werden alternative Bezahlungsvarianten verdeutlicht und können auch für eine konkrete Situation bewertet werden.

2.3.4.2 Prozessmodellierung für E-Procurement

Die Geschäftsprozesse in einem E-Business-Modell können als weitergehende Modellierungsmöglichkeit betrachtet werden. Dies soll beispielhaft für das gewählte E-Procurement-Szenario gezeigt werden.

Als Grundlage und zur besseren Übersicht wird zunächst ein grobes Übersichtsmodell erstellt (Abbildung 2.36). Es bildet die Grundlage für die weitere detailreichere Modellierung.

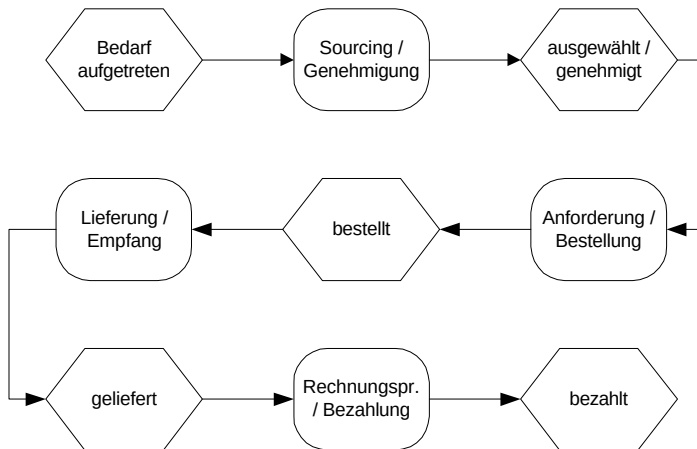


Abbildung 2.36: EP-Prozessmodell (Übersicht)

Es sind vier Funktionen enthalten, die genauer zu modellieren sind:

- Sourcing / Genehmigung
- Anforderung / Bestellung
- Lieferung / Empfang
- Rechnungsprüfung / Bezahlung

Jede Funktion wird nun wiederum durch EPKs dargestellt. Zusätzlich erfolgt jeweils eine kurze textuelle Beschreibung des Ablaufs, um die grafischen Darstellungen zu ergänzen.

Aus den einzelnen, aneinander anschließenden EPK's ergibt sich das Gesamtbild.

Ergänzend zu den Standards für EPK werden Funktionen, die völlig autonom ohne Benutzerbeteiligung vom System ausgeführt werden können, gestrichelt dargestellt. Dies erlaubt auf den ersten Blick eine Übersicht über die Aufgaben, die das System den menschlichen Benutzern komplett abnehmen kann. Die Übersicht kann als Grundlage für eine Abschätzung dienen, an welchen Punkten und in welchem Umfang Zeit und Kosten eingespart werden.

Diese Konvention kann als Charakteristik einer speziellen E-Business-Modellierung angesehen werden.

2.3.4.2.1 Sourcing / Genehmigung

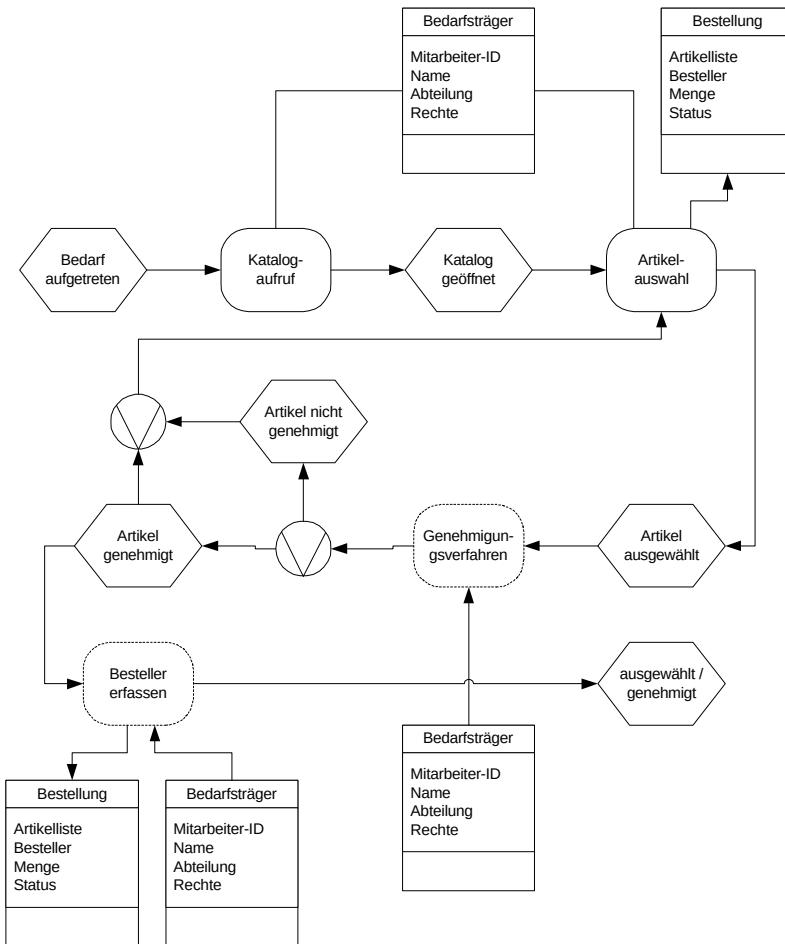


Abbildung 2.37: EP-Prozessmodell-Sourcing / Genehmigung

Ausgelöst wird ein Bestellvorgang durch das Auftreten eines Bedarfs. Unter Zuhilfenahme eines Katalogs können ein oder mehrere Artikel ausgewählt werden, um den Bedarf zu befriedigen. Für jeden dieser Artikel wird zunächst festgestellt, ob er grundsätzlich bestellt werden darf. Ist dies der Fall und ist die Artikelauswahl abgeschlossen, so werden zum Warenkorb noch die Daten des Bestellers erfasst, bevor der Sourcing / Genehmigungs – Vorgang abgeschlossen wird.

2.3.4.2.2 Anforderung / Bestellung

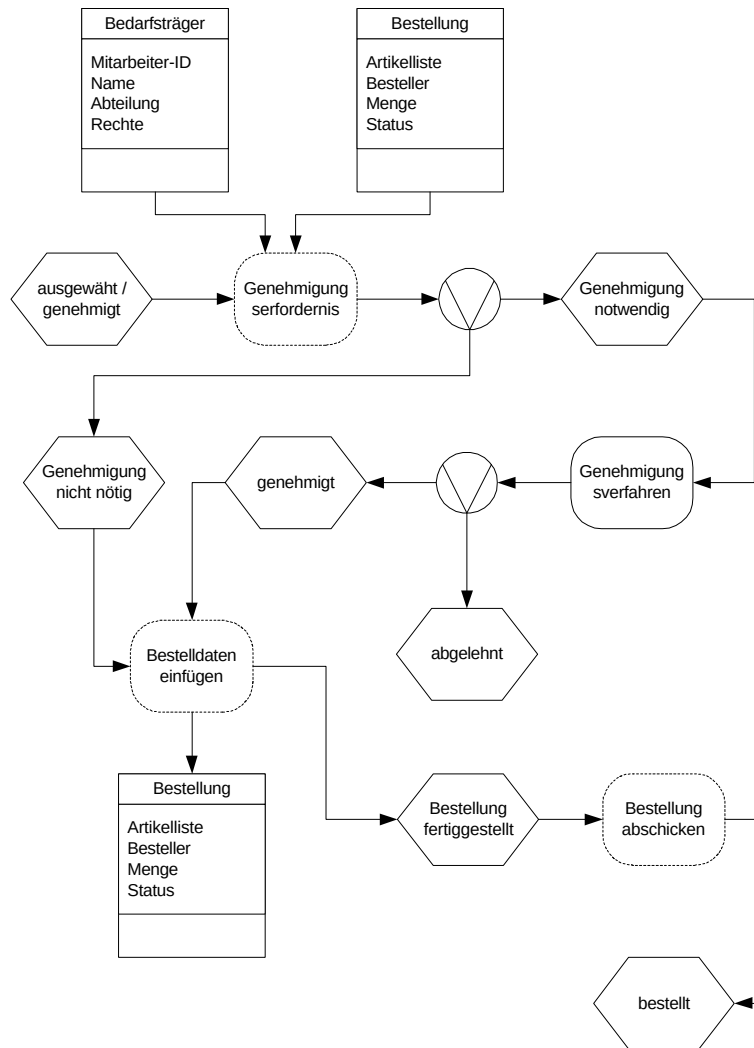


Abbildung 2.38: EP-Prozessmodell-Anforderung / Bestellung

Nachdem ein kompletter Warenkorb vorliegt, wird noch einmal geprüft, ob der Warenkorb in seiner Gesamtheit genehmigt werden muss. Ist das der Fall, so wird ein Genehmigungsverfahren eingeleitet, an dessen Ende entweder eine Genehmigung oder eine Ablehnung steht. Letztere hat einen Abbruch des Bestellvorgangs zur Folge. Im Falle einer Genehmigung wird genauso vorgegangen wie bei Genehmigungsfreiheit. Es werden die Bestelldaten eingefügt, beispielsweise Firmendaten, Lieferantendaten oder Materialstammdaten, und die Bestellung wird an die Lieferanten verschickt.

Das Modell könnte durch Einbeziehung einer Lagerbestandsprüfung oder durch weitere Detaillierung des Genehmigungsverfahrens weiter erweitert und verfeinert werden.

2.3.4.2.3 Lieferung / Empfang

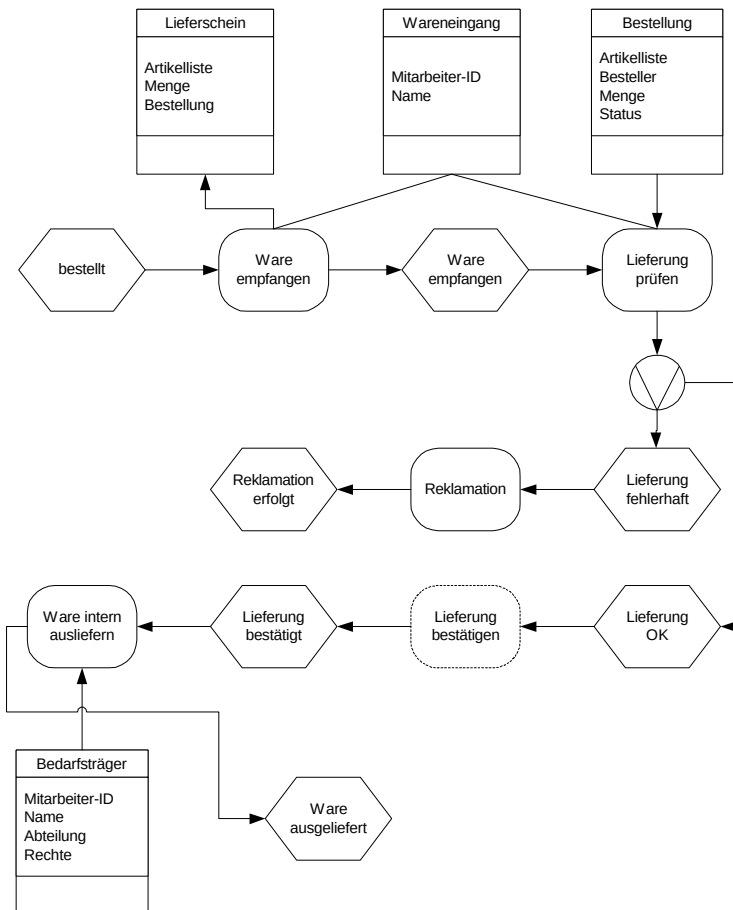


Abbildung 2.39: EP-Prozessmodell-Lieferung / Empfang

Nachdem im vorigen Schritt die Ware bestellt wurde, wird der Prozess erst fortgesetzt, sobald die bestellte Ware angeliefert wird. Ist das erfolgt, so wird die Ware gegen den Lieferschein geprüft. Zusätzlich kann auch noch eine Prüfung gegen die Bestellung erfolgen. Schlägt diese Überprüfung fehl, so wird die Ware beim Lieferanten reklamiert. Ist die Prüfung erfolgreich abgeschlossen worden, so kann die Lieferung im System bestätigt werden. Da von einer zentralen Warenannahme

ausgegangen wird, muss abschließend die Ware noch intern ausgeliefert werden.

Auch an dieser Stelle besteht noch weiteres Potential für Erweiterungen, beispielsweise die Behandlungen von Teillieferungen.

2.3.4.2.4 Rechnungsprüfung / Bezahlung

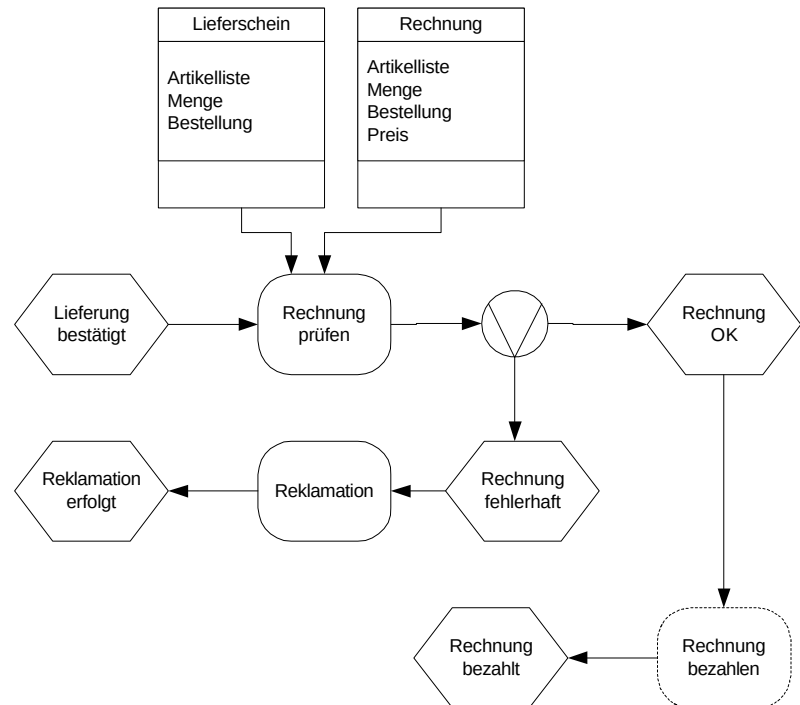


Abbildung 2.40: EP-Prozessmodell-Rechnungsprüfung / Bezahlung

Das Prozessmodell geht davon aus, dass die Rechnung eingeht, sobald die Lieferung bestätigt wurde. Die Rechnung wird anhand des Lieferscheines geprüft, reklamiert, wenn sie fehlerhaft ist, und bezahlt, wenn sie korrekt ist.

2.3.5 Verhandlungsaspekte im E-Procurement

Gerade im Bereich des E-Procurement besteht Bedarf einer dynamischen und flexiblen Aushandlung von Preisen und Konditionen (Bichler, et al., 2002).

Gegenstand dieser Arbeit sind jene Prozesse, die im laufenden Geschäft des E-Procurement vorkommen. Die vorherige Aushandlung von Rah-

menverträgen oder ähnlichen Vereinbarungen ist damit nicht Teil der Betrachtung.

Es geht also um das Zustandekommen und die dynamische Festlegung von einzelnen Konditionen eines Handels.

Als mögliche verhandelbare Konditionen kommen unter anderem in Frage:

- der Preis eines Artikels
- der Liefertermin eines Artikels
- die Qualität eines Artikels
- Umfang von Garantieleistungen
- Umfang von Serviceleistungen zu den Artikeln / zum Kauf
- einfache oder wechselseitige Abhängigkeiten zwischen den Konditionen
- Konditionen, die Teilgruppen von Artikeln eines Angebotspaketes betreffen; beispielsweise Rabatte bei Erreichen eines Gesamtbestellvolumens

Die in Abschnitt 4.1 entwickelten Fallbeispiele geben einen Einblick in mögliche konkrete Szenarien.

2.3.5.1 Modellierung von Verhandlungsaspekten im E-Procurement

Die im E-Procurement auftretenden Verhandlungsaspekte lassen sich generell und beispielhaft an der in Abschnitt 2.2 entwickelten E-Business-Modellierung aufzeigen.

In einer Verhandlung entsprechen die *Verhandlungsteilnehmer* dem Konzept des *Akteurs* in der e³-Value-Modellierung.

Die *Verhandlungsgegenstände* entsprechen den *Wertobjekten*, da sie für wenigstens einen Akteur von Wert sind. Eine detaillierte Modellierung der Wertobjekte ist für die Abbildung von Verhandlungen wichtig, um auch die verhandelbaren Attribute eines Verhandlungsgegenstandes deutlich zu machen.

Das Konzept des *Wertaustausches* in der e³-Value-Modellierung entspricht dem *Austausch von Verhandlungsgegenständen* und Gegenleistungen nach dem Vertragsabschluss.

Bei zusammengesetzten Verhandlungen, die aus Teilverhandlungen mit gleichzeitig mehreren anderen Verhandlungsteilnehmern bestehen, kann in der e³-Value-Modellierung das Konzept *Value-Angebot* benutzt werden. In diesem Konzept findet ein Austausch aller Wertobjekte statt, wenn alle Teilverträge abgeschlossen sind, oder es findet kein Austausch statt, wenn mindestens eine Teilverhandlung nicht erfolgreich war. Im letzten Fall müssen die abgeschlossenen Teilverträge zurückgenommen werden und das Protokoll muss diese Rücknahme erlauben.

Die Konzepte ‚*Value-Interface*‘ und ‚*Value-Ports*‘ aus der e³-Value-Modellierung informieren in einer Verhandlung darüber, welche Verhandlungsgegenstände ein Verhandlungsteilnehmer zum Tausch, zum Kauf oder zum Verkauf anbieten möchte und was er als Gegenleistung dafür erwartet.

In der e³-Value-Modellierung sind jeweils alternative Szenarien bildbar. Das entspricht auch einer Verhandlung über verschiedene Konstellationen von Wertaustauschen.

Als ein konkretes Beispiel kann das E-Procurement-Szenario in Abbildung 2.35 betrachtet werden. Das e³-Value-Modell zeigt einen möglichen Austausch des Wertobjekts ‚Bestellung‘ gegen das Wertobjekt ‚Bezahlung‘ zwischen ‚Broker‘ und Anbieter. Die Höhe des jeweiligen Austauschs ist verhandelbar.

Mit Hilfe von Elementen der e³-Value-Modellierung ist es möglich, die grundlegenden Informationen über Verhandlungsteilnehmer und Verhandlungsgegenstände darzustellen, die für eine Verhandlung wichtig sind. Für eine noch genauere Beschreibung der Verhandlungsszenarien müssen weitere Elemente in die e³-Value-Modellierung aufgenommen werden. So können zum Beispiel Informationen über den Ablauf einer Verhandlung, wie Richtung oder Vertragsakzeptanz, an den ‚*Value-Interfaces*‘ angegeben werden. Eine andere Möglichkeit ist die Angabe des Verhandlungsprotokolls an den ‚*Value-Interfaces*‘.

Zwar werden mit der e³-Value-Modellierung nicht alle für eine Verhandlung relevanten Eigenschaften dargestellt, aber die wesentlichen Elemente einer Verhandlung können mit Hilfe dieser Modellierungsmethode grafisch dargestellt werden.

3 Elektronische Verhandlungen: Grundlegende Strukturen und Unterstützungssysteme

Eine Verhandlung kann definiert werden als eine Diskussion, bei der die beteiligten Parteien Informationen austauschen und zu einer gemeinsamen Einigung kommen (vgl. (Davis, et al., 1983)).

Verhandlungen treten in zahlreichen Kontexten wie in der Politik, im Recht, im privaten Bereich und insbesondere auch in geschäftlichen Beziehungen auf.

Für diese Arbeit stehen die Verhandlungen in Geschäftsbeziehungen und hier die elektronisch unterstützten Verhandlungen und die dafür notwendigen Systeme und Infrastrukturen im Mittelpunkt.

3.1 Charakterisierung von Verhandlungen

Zunächst soll eine Ausgangsdefinition für Verhandlungen vorgestellt werden, die dann in der Folge durch weitere Sichten ergänzt und detailliert wird.

Davis und Smith (Davis, et al., 1983; Smith, 1980), die sich mit ‚Distributed-Problem-Solving‘ beschäftigten und dafür das bekannte ‚Contract-Net‘ entwickelten, charakterisieren eine Verhandlung als

- “a discussion
- in which interested parties
- exchange information and
- come to an agreement”

Diese allgemeine, jedoch auch für den Electronic Business zutreffende und auch heute noch gültige Definition führt zu den folgenden Überle-

gungen im Kontext dieser Arbeit, die als ergänzende und konkretisierende Definition zu sehen sind:

An einer Verhandlung sind zwei oder mehr autonome Akteure (Parteien) beteiligt. Eine Voraussetzung ist, dass die Akteure an einer gemeinsamen Verhandlung interessiert sind. Dieses Interesse beruht auf einem Interessenskonflikt bezüglich bestimmter Verhandlungsgegenstände bzw. -objekte, dessen Lösung jedem der beteiligten Akteure Vorteile bringen kann. Um eine Lösung zu erreichen, sind die Akteure bereit in einer wechselseitigen Diskussion zu kommunizieren. Diese Diskussion kann dazu führen, dass die Akteure mit ihren ursprünglich unterschiedlichen Zielvorstellungen zu einer für alle Seiten akzeptablen Lösung und damit Einigung finden. Das jederzeit mögliche Ende einer Verhandlung ist der, gegebenenfalls einseitige, Abbruch der Verhandlung. Bei einem Abbruch ist das Interesse eines oder mehrerer Akteure als Voraussetzung einer Verhandlung nicht mehr gegeben.

Während einer Verhandlung können sich im Wesentlichen folgende Punkte verändern:

- die subjektiven Einschätzungen der beteiligten Akteure zu den Verhandlungsobjekten,
- die Verhandlungsobjekte selbst können konkretisiert, ergänzt oder sogar ganz verändert werden.

Diese Veränderungen sind bedingt durch:

- den *Informationsaustausch*, die Diskussion, die zwischen den beteiligten Akteuren stattfindet,
- *externe Informationen*, welche die beteiligten Akteure während der Verhandlung aufnehmen oder aktiv beschaffen,
- den *Fluss der Zeit*, der den aktuellen Zeitpunkt ändert und so neue Verhältnisse schaffen kann; z.B. kann ‚Zeitdruck‘ ausgelöst werden.

Zur weiteren Einordnung von Verhandlungen ist deren Charakteristik als Prozess einer Entscheidungsfindung von besonderer Bedeutung. Rebstock definiert diesen Prozess als einen „auf gegenseitigem Informationsaustausch ... [beruhenden] Prozess der Entscheidungsfindung zwischen zwei oder mehr Parteien, der in der Regel eine Einigung zum Ziel hat“ (Rebstock, 2001). Eine Verhandlung kann somit auch als ein Entscheidungsprozess gesehen werden und zwar als ein Informationsaustausch, bei dem zwei oder mehrere Akteure (Parteien) eine Reihe von Entscheidungen treffen, die einen Kompromiss zum Ergebnis haben (Kersten, et al., 2000).

Zur allgemeinen Unterstützung von Entscheidungsprozessen gibt es eigene Forschungen (vgl. (North-Holland, 1985)) und Systeme, die als ‚Decision-Support-Systeme‘ (DSS) bezeichnet werden.

Verhandlungsunterstützungssysteme (,Negotiation-Support-Systems') werden auch als NSS sowie als ,Group-Decision-Support-Systeme' (GDSS) bezeichnet. Die Begriffe ,Group-Decision-Making' und ,Negotiation' werden des Öfteren als Synonyme verwendet, wie es auch in den einschlägigen Journals und Konferenzen, beispielsweise der ,Group Decision and Negotiation Journal" sowie der gleichnamige Konferenzreihe zu sehen ist.

Ein Unterscheidungskriterium ist jedoch, dass ein Entscheidungsprozess eher als eine Verhandlung gesehen werden kann, wenn die Verhandlungspartner klar von einander abgegrenzte, individuelle Ziele verfolgen und die Parteien die Informationen über ihre Ziele untereinander nicht preisgeben wollen (Kettunen, 1999).

3.2 Klassifikation von Verhandlungen

Die Klassifikation von Verhandlungen ist eine ausgesprochen anspruchsvolle Aufgabe, da es in vielfältigen Situationen zu Verhandlungen kommen kann, die auf äußerst vielfältige Art und Weise durchgeführt werden können.

Selbst bei Einschränkung auf die Anwendungsdomäne des Electronic Business ergeben sich noch viele Klassifikationsmöglichkeiten.

3.2.1 Klassifikationskriterien

Aus den Elementen des oben vorgestellten Basismodells für Verhandlungen ergeben sich direkt Klassifikationskriterien. Im Folgenden wird nur auf die komplexeren Kriterien eingegangen, die für die später vorgestellten Klassifikationssichten und -schemata wichtig sind.

3.2.1.1 Distributive und integrative Verhandlungen

Wenn in einer Verhandlung eine Partei nur auf Kosten einer anderen Partei ihre Position verbessern kann, so spricht man von einer distributiven Verhandlung. Dies ist beispielsweise bei Verhandlungen der Fall, bei denen nur über ein Attribut verhandelt wird. In 3.2.1.3 wird auf diese einattributiven Verhandlungen näher eingegangen.

Bei integrativen Verhandlungen kann eine gemeinsame Lösung gefunden werden kann, die gleichzeitig den Nutzen aller beteiligten Parteien erhöht. Es entsteht eine so genannte Win-Win-Situation.

Multiattributive Verhandlungen sind meist integrativ, weil es für jeden Verhandlungsteilnehmer möglich ist, durch das Nachgeben in Hin-

sicht auf ein spezielles Attribut bei einem anderen Attribut, das für diese Partei wichtiger ist, einen Vorteil zu erzielen. Diese Abwägungen zwischen den Attributen werden auch als ‚Trade-Off‘ bezeichnet. Der Verhandlungsprozess wird dadurch jedoch auch komplexer als der einer einattributiven Verhandlung. Die Parteien müssen Schwerpunkte beziehungsweise Präferenzen setzen, welche Attribute ihnen in der Verhandlung besonders wichtig sind, bei denen sie also nicht Nachgeben wollen, und bei welchen Attributen Zugeständnisse an den Verhandlungspartner denkbar sind. Multiattributive Verhandlungen geben außerdem die Möglichkeit, während der Verhandlung weitere Attribute der Ware in die Diskussion mit aufzunehmen und so „den zu verteilenden Kuchen“ zu vergrößern. Obwohl multiattributive Verhandlungen meist integrativ sind, können sie als Sonderfall jedoch auch distributiv sein.

3.2.1.2 Anzahl der Verhandelnden

Die Anzahl der an einer Verhandlung beteiligten Parteien bestimmt im Wesentlichen die so genannte *Protokollkategorie*. Gibt es nur zwei Verhandlungsparteien, spricht man von einer *bilateralen Verhandlung*. Nehmen mehrere Parteien an der Verhandlung teil, entspricht dies einer *multilateralen Verhandlung*. Nach Rebstock (Rebstock, 2001) werden weiterhin *einseitig multilaterale Verhandlungen*, beispielsweise viele Auktionen, und *beidseitig multilaterale Verhandlungen* wie Börsen oder doppelte Auktionen unterschieden.

Die einseitig multilaterale Verhandlung, auch One-To-Many-Verhandlung genannt, ist ein häufiges Szenario im Electronic Business bei dem beispielsweise ein Käufer mit mehreren potentiellen Verkäufern in Verhandlung tritt.

Eine einfache Lösung für diesen Fall ist die Abwicklung von mehreren bilateralen, integrativen Verhandlungen des Käufers mit den einzelnen Verkäufern. Aus den Ergebnissen dieser Verhandlungen muss der Käufer dann anschließend das für ihn beste Ergebnis ermitteln. Es entsteht dabei die Schwierigkeit, dass die einzelnen Verhandlungen sehr unterschiedlich verlaufen können, da die verschiedenen Verkäufer möglicherweise unterschiedliche Aspekte ihrer Produkte hervorheben möchten. Hieraus können sich große Unterschiede in der Wahl der zu verhandelnden Attribute ergeben, so dass sich die Ergebnisse nicht mehr einfach miteinander vergleichen lassen. Davon abgesehen wäre ein solcher Prozess sehr zeitaufwendig (vgl. (Bichler, et al., 1999)).

Eine One-To-Many-Verhandlung stellt oft ein typisches Auktions-szenario dar. Auktionen sind größtenteils distributiv. Auch in diesem Fall handelt es um eine distributive Verhandlung. „The negotiation is implicitly competitive in nature, as it is resolving a conflict of several suppliers over a single mutually exclusive goal, the deal with the buyer“ (Bichler,

et al., 1999). Man könnte den beschriebenen Verhandlungsprozess als eine Art hierarchisch strukturierte Verhandlung auffassen, eine distributive Verhandlung über mehrere integrative Verhandlungen.

Bisherige Forschungen über Verhandlungen in der Theorie wie auch in der Praxis haben gezeigt, dass integrative den distributiven Verhandlungen im Allgemeinen vorzuziehen sind (Kersten, et al., 2000). Von einer integrativen Verhandlungsform können alle verhandelnden Parteien profitieren und somit werden im Gegensatz zu distributiven Verhandlungen die besseren Ergebnisse erzielt. Bei der hier zu Grunde liegenden One-To-Many-Verhandlung ergeben sich jedoch im integrativen Fall die oben erwähnten Schwierigkeiten.

Nach Kersten et al. (Kersten, et al., 2000) bietet sich für ein solches Szenario eine Kombination aus Auktion und Verhandlung an. Hierbei sind drei Varianten denkbar. Zunächst besteht die Möglichkeit, einer vorangehenden Auktion, welche die Anzahl der Anbieter einschränkt. Mit denen tritt der Käufer dann anschließend in bilaterale Verhandlungen. Des Weiteren können Verhandlungen zuerst stattfinden und anschließend eine Auktion. Die vorausgehenden Verhandlungen helfen festzulegen, welche Attribute verhandelt werden und wie hoch die Anfangsgebote für die folgende Auktion liegen. Als dritte Möglichkeit ist eine hybride Form von Auktion und Verhandlung denkbar, die bereits vor drei Jahren von Teich vorgeschlagen wurde (Teich, 2000). Hierbei gibt es einen Käufer, der mehrere Anbieter auswählt und dann nach bestimmten Kriterien deren Anzahl, basierend auf den von ihm gewählten Attributen, reduziert. Dies kann wahlweise automatisch oder manuell geschehen, wobei das automatisierte Vorgehen eher einer Auktion entspricht, während das manuelle Verfahren einzelne integrative Verhandlungen mit den Anbietern erlaubt. Außerdem gibt es noch die Möglichkeit, durch einen Pausierungsmodus einzelne Anbieter zeitweise von der Auktion auszuschließen. Durch das Wechseln zwischen den einzelnen Modi entsteht eine hybride Form von Auktion und Verhandlung.

3.2.1.3 Anzahl der Attribute – Multiattributive Verhandlungen

Das Merkmal der *Anzahl der Attribute* bestimmt, über wie viele Attribute ein Verhandlungsobjekt verfügt. Beinhaltet es nur eines, so spricht man von einer einattributiven Verhandlungsform. Diese ist beispielsweise oft bei Auktionen anzutreffen, bei denen nur über den Preis einer Ware verhandelt wird. Verfügt das Verhandlungsobjekt über mehrere Attribute, so spricht man von einer multiattributiven Verhandlung. Bei diesen Verhandlungen muss eine Einigung bezüglich aller Attribute erfolgen.

3.2.1.4 Anzahl der Positionen

Das Merkmal *Anzahl der Positionen* gibt die Möglichkeit an, über mehrere verschiedene Produkte auf einmal zu verhandeln. Dies ist hauptsächlich bei Auktionen relevant.

3.2.1.5 Anzahl der Verhandlungen

Das Merkmal der *Anzahl der Verhandlungen* gibt an, ob parallel mehrere Verhandlungen geführt werden.

3.2.1.6 Automatisierungsgrad

Der *Automatisierungsgrad* untergliedert sich in vollautomatisierte Verhandlungen, hybride bzw. teilautomatisierte Verhandlungen und lediglich elektronisch unterstützte Verhandlungen.

Vollautomatisierte Verhandlungen werden von Rebstock (Rebstock, 2001) auch als *ergebnisorientiert* betrachtet, wohingegen er elektronisch unterstützende Ansätze als *prozessorientiert* einordnet. Dies entspricht der Klassifizierung von Verhandlungsunterstützungssystemen (NSS) von Yuan et al. (Yuan, et al., 1998) in ‚Solution-Driven-NSS‘ und ‚Process-Support-NSS‘.

Nach dem bisherigen Stand der Forschung scheint diese Einteilung jedoch fraglich, da sowohl automatisierte, prozessorientierte Systeme, wie auch wenig bis gar nicht automatisierte, lösungsorientierte Systeme zumindest denkbar sind. Insofern ist der Automatisierungsgrad eines Systems kein Hinweis darauf, ob es ein lösungsorientiertes oder ein prozessorientiertes System ist.

3.2.1.7 Mediationstyp

Der *Mediationstyp* gibt an, ob die Verhandlung durch einen Vermittler (Mediator) unterstützt wird oder ob die Parteien direkt miteinander verhandeln. Ein Vermittler ist meist erforderlich um multilaterale Verhandlungen zu führen. Bilaterale Verhandlungen kommen dagegen oft auch ohne einen Vermittler aus.

Eine weitere verhandlungsbezogene, inhaltliche Sicht sieht den Vermittler als Akteur, dem die anderen Verhandlungsparteien gegenüber gegebenenfalls ihrer eigenen Interessen mehr offenbaren. Mit diesem kombinierten Zusatzwissen ist der Vermittler in der Lage, für alle beteiligten bessere Lösungen vorzuschlagen.

3.2.2 Klassifikationssichten

Auf der Grundlage des Bereichs Electronic Business kann in der Folge auf bestimmte Ausprägungen von Verhandlungen und Klassifikationssichten fokussiert werden, die in dieser Domäne besondere Relevanz haben und jeweils verschiedene Präferenzen für einzelne Kriterien aufweisen.

3.2.2.1 Auktionen

Auktionen nehmen eine besondere Stellung im Bereich der elektronischen Verhandlungen ein, da sie seit jeher ein klassisches Mittel für Verhandlungen sind und durch ihre exakten Protokolle besonders für den Einsatz im Electronic Business geeignet sind. Auktionen sind eine Verhandlungsform und enthalten explizite Ablaufprotokolle, die auch in andere Verhandlungsformen eingebettet werden können. Auktionen sind in den Wirtschaftswissenschaften schon lange Gegenstand der Forschung (Klemperer, 1999; McAfee, et al., 1987; Milgrom, 1989; Milgrom, 1998; Milgrom, et al., 1982). Aber auch aus Sicht der Informatik wurden sie eingehend betrachtet (Noriega, 1997; Reck, 1997) und gute Klassifikationen entwickelt (Reck, 1997).

Auktionen werden derzeit auf vielen elektronischen Marktplätzen im B2B oder B2C-Bereich eingesetzt. Siehe dazu auch das Geschäftsmodell der E-Auktionen in Kapitel 2. Jedoch steht dabei zumeist nur die Einigung über ein einzelnes Attribut der Ware, beispielsweise den Preis, im Vordergrund.

Um eine weitere Abgrenzung zwischen Auktionen und Verhandlungen zu erreichen, lässt sich sagen, dass bei einer Auktion der Wert bzw. der Preis des Produktes durch einen Prozess festgelegt wird, der nur von einer Seite gesteuert wird. Bei einer Verhandlung sollte dieser Prozess jedoch von allen Teilnehmern gesteuert werden (Kersten, et al., 2001). Die Vorteile von Auktionssystemen sind nach Kersten und Lo die Prozesseffizienz, die einfache Benutzbarkeit, die Erreichbarkeit und die simultane Verarbeitung einer sehr großen Zahl von Benutzern. Jedoch funktionieren Auktionen nur mit bekannten und wohl definierten Objekten, während Verhandlungen es erlauben, diese Objekte erst durch die Verhandlung zu definieren und gleichzeitig auch die Vorstellungen und Präferenzen der Verhandelnden zu prägen. Hierdurch ist es bei Verhandlungen auch möglich Objekte zu verhandeln, die sehr komplex oder nicht ausreichend definiert sind. Eine Verhandlung ist typischerweise kostenintensiver in Bezug auf die Zeit und den Aufwand, die benötigt werden, um zu einer Einigung zu gelangen (Kersten, et al., 2001). Kersten und Lo halten fest, dass trotz der Vorteile, die eine Auktion bieten kann, Verhandlungen auf

keinen Fall durch Auktionen ersetzt werden können, da es sich um zwei komplementäre Mechanismen handelt.

Über eine Verhandlung werden meist auch länger andauernde Geschäftsbeziehungen geknüpft. Diese Option ist jedoch bei rein auktionsbasierten Marktplätzen oft stark reduziert.

3.2.2.2 Klassifizierung nach Rebstock

Rebstock (Rebstock, 2001) nutzt sechs der oben vorgestellten Kriterien zur Klassifizierung von elektronischen Verhandlungen:

- Anzahl der Verhandellnden (Protokollkategorie / Bi- oder Multilateral)
- Anzahl der Attribute
- Anzahl der Positionen
- Anzahl der Verhandlungen
- Automatisierungsgrad
- Mediationstyp

Dieses Klassifikationsschema ist inzwischen weithin bekannt.

Eine weitere beachtenswerte Klassifikation ist die zunächst als ‚London Classification‘ und später als ‚Montreal Classification‘ weitergeführte und von Ströbel (Ströbel, 2003) dokumentierte Klassifikation.

3.2.3 Verhandlungen im Electronic Business

Elektronische Verhandlungen sind nach ihrer Einordnung in der Vereinbarungsphase von elektronisch abgewickelten Handelstransaktionen ein integraler Bestandteil des Electronic Business.

Beam und Segev (Beam, et al., 1997) definieren eine Verhandlung im Electronic Commerce als: „the process by which two or more parties multilaterally bargain resources for mutual intended gain, using the tools and techniques of electronic commerce“.

Zum aktuellen Stand von Verhandlungen im Electronic Business merkt Rebstock (Rebstock, 2001) an, dass „der derzeitige Funktionsumfang elektronischer Märkte – in der Regel katalogbasierte Beschaffung, einattributive Auktionen und Ausschreibungen – ... aus Sicht der Unternehmenspraxis oft unbefriedigend [ist], da mit ihm nur ein kleinerer Teil des Gesamtbeschaffungsvolumens abgedeckt werden kann. Weitergehende Verhandlungsfunktionen werden vermisst.“

Es besteht also weitgehender Forschungs- und Handlungsbedarf in diesem Feld.

Verhandlungen im Electronic Business sind zunächst allgemein zu betrachten. Es existieren jedoch einige Charakteristika.

In Verhandlungen im Electronic Business kann über vielfältige Konditionen verhandelt werden. Diese Verhandlungen beziehen sich letztendlich alle auf das Zustandekommen einer Handelstransaktion.

Ein besonderes Merkmal ist auch, dass als Verhandlungsobjekte die angebotenen Produkte eines Unternehmens referenziert werden.

Bei Verhandlungen im Electronic Business kommt es weiterhin nicht zu echten Langzeitverhandlungen. Die üblichen Verhandlungen sind innerhalb weniger Sekunden bis maximal einigen Tagen abgeschlossen.

3.3 Basismodell für Verhandlungen im Electronic Business

Zur Entwicklung eines Basismodells für Verhandlungen wird zunächst auf die oben vorgenommene allgemeine Charakterisierung des Phänomens der Verhandlung zurückgegriffen. Das Modell wird unter Berücksichtigung der bisherigen Forschung dann durch Identifikation von einzelnen Verhandlungselementen und der Betrachtung des zeitlichen Ablaufs von Verhandlungen formal konkretisiert.

3.3.1 Basismodell

Im Anschluss an die allgemeine Analyse von Verhandlungen im Electronic Business sollen die gewonnenen Erkenntnisse in einem formalen Modell auf Basis relationaler Algebra detailliert festgehalten werden. Das Ziel dieser Formalisierung ist:

- die Exaktheit des Modells zu fördern,
- die maschinelle Verarbeitung des Modells zu erleichtern.

Das in der Folge entwickelte Modell basiert auf den bisherigen Analysen und Arbeiten und führt diese mit dem von Ströbel (Ströbel, 2003) im Frühjahr 2003 vorgestellten ‚formal model‘ zusammen. Weitere Beiträge sind im Zusammenhang durch die auf dem Basismodell beruhende Entwicklung einer systemtechnischen Verhandlungsinfrastruktur zu sehen (siehe auch Kapitel 5 und (Gräfe, 2003)).

Das formale Basismodell wird untergliedert in verschiedene inhaltliche Abschnitte vorgestellt. Durch formale Definitionen, die meist kurz erläutert werden, wird das Modell schrittweise aufgebaut.

Zu einer Definition kann es *weiterführende Anmerkungen* geben, die als solche gekennzeichnet sind. Sie stellen einen nicht formalisierten Teil des Modells dar. Die Anmerkungen dienen als ergänzende Informationen zur allgemeinen Verwendung des Modells sowie zur möglichen Weiterentwicklung. Teils treten die genannten Sachverhalte später im Modell formalisiert auf, dienen aber in diesem Falle bei der vorhergehenden informellen Nennung dem besseren Gesamtverständnis des Modells.

3.3.1.1 Verhandlungssituation

3.3.1.1.1 Verhandlungsakteure

Definition 1: Akteur (Synonyme: Verhandlungsakteur, -partner, -partei, -teilnehmer, Agent)

$$a \in A = \{a_1, a_2, \dots, a_n\}$$

Ein Akteur a ist eine autonome, aktive Entität in einer Verhandlung. Ein Akteur kann menschlich sein, maschinell repräsentiert werden oder auch maschinell automatisiert sein.

Weiterführende Anmerkungen:

Eine wichtige Informationen je Akteur ist beispielsweise:

- Anonymität: Wie anonym ist ein Verhandlungsakteur jeweils gegenüber den anderen Akteuren?

Bei den Akteuren speziell zu beachtende Gegebenheiten sind:

- der Ein- und Austritt einzelner Akteure während einer (multi-lateralen) Verhandlung.

Definition 2: Rolle (Synonyme: Verhandlungsrolle, Agent type)

$$r \in R = \{\text{Käufer}, \text{Verkäufer}, \text{Vermittler}, \text{Vführer}\}$$

Verhandlungsakteure können im Rahmen gewisser Kontexte, beispielsweise bei einer bestimmten Anwendungsdomäne oder bei Verwendung spezieller Ablaufprotokolle, bestimmte Rollen übernehmen. Akteure können in einer Verhandlung auch gleichzeitig mehrere Rollen übernehmen.

Im Rahmen dieses Modells, orientiert am Ablauf einer Handelstransaktion im Electronic Business, sind dies die Rollen ‚Käufer‘, ‚Verkäufer‘, ‚Vermittler‘ und ‚Vführer‘ (Verhandlungsführer).

Weiterführende Anmerkungen:

Grundsätzliche kann man drei Rollen im Electronic Business unterscheiden:

- Verhandlungsakteur, der ein direktes Interesse an den Verhandlungsobjekten hat, konkret sind dies:
 - Verkäufer und
 - Käufer,
- Vermittler (Mediatoren) können in eine Verhandlung eintreten. Sie sind interessensneutral zu den Verhandlungsobjekten. Ihre einzige Motivation sollte sein, den erfolgreichen Abschluss der Verhandlung zu unterstützen und aktiv zu fördern.
- Verhandlungsführer(Vführer), der eine leitende, d.h. aktive und organisierende / regulierende / überwachende Funktion übernimmt. Verhandlungsführer verfügen über besondere Rechte im Rahmen einer Verhandlung, die von den in eine Verhandlung eintretenden Akteuren anerkannt werden müssen.

Ergänzend können die hier definierten Rollen speziellere Ausprägungen annehmen. Als Beispiel sei die Verwendung des Ablaufprotokolls ‚Auktion‘ in einer Verhandlung genannt. Die Bieter sind eine spezielle Ausprägung der Rolle Käufer. Der Auktionator ist eine Ausprägung von Verhandlungsführer.

Wichtige Informationen je Rolle sind beispielsweise:

- *Berechtigungen* (wofür, welche): Beispiel: in einer Auktion darf ein Akteur in der Rolle des Auktionators kein Gebot abgeben

Definition 3: Rollen des Akteurs (Synonyme: Agent roles)

$$R_a \in 2^R$$

Verhandlungsakteure können eine oder mehrere Rollen übernehmen. Die Menge der Rollen eines Akteurs ist damit Element der Potenzmenge aller möglichen Rollen.

Weiterführende Anmerkungen:

Akteure können ihre Rollen im Laufe einer Verhandlung wechseln.

3.3.1.1.2 Verhandlungsobjekte

Definition 4: Attributname (Synonyme: property, terms&conditions)

$$an \in AN = \{an_1, an_2, \dots, an_n\}$$

Ein Attribut enthält nähere Informationen über ein Verhandlungsobjekt, wie beispielsweise den Preis eines zu verkaufenden Gutes oder der Ort der Anlieferung.

Definition 5: Attributoperator (Synonyme: attribute operator)

$$ao \in AO = \{<, \leq, =, \neq, \geq, >\}$$

Definition 6: Attributwert (Synonyme: attribute value)

$$aw \in AW = \{aw_1, aw_2, \dots, aw_n\}$$

Weiterführende Anmerkungen:

Es sind beliebige Werte zulässig, insbesondere auch Werte, die eine Maximierung oder Minimierung ausdrücken.

Definition 7: Attributeinheit (Synonyme: Einheit, Unit)

$$ae \in AE = \{_, ae_1, ae_2, \dots, ae_n\}$$

Das Symbol '_' steht für 'keine Einheit'.

Definition 8: Domäne (Synonyme: Domain)

$$DOM \subset AN \times AO \times AW \times AE \mid AN \neq \{\}$$

$$DOM_{an} \subset an \times AO \times AW \times AE$$

Weiterführende Anmerkungen:

Es gibt verschiedene Arten von Domänen, von denen einige spezielle näher vorgestellt werden:

- *Punkt-Domäne*: Nur ein Tupel mit einem Attributnamen und dem '=' Operator wird angegeben.
Beispiel: (Farbe,=,blau,_)
- *Diskrete Domäne*: Für einen Attributnamen werden mehrere Tupel mit dem '=' Operator angegeben.
Beispiel: ((Farbe,=,blau,_),(Farbe,=,blau,_))
- *Stetige Domäne*: Für einen Attributnamen werden ein oder mehrere Tupel mit einem anderen Operator als '=' angegeben.
Beispiel: ((Festplatte,>=,20,GB),(Festplatte,<=80GB,GB))
- *Ausschlussdomäne*: Die Punkt-Domäne und die Diskrete Domäne werden zu Ausschlussdomänen, wenn anstelle des Gleichheitsoperators der Ungleichheitsoperator verwendet wird.

Definition 9: Dynamiksymbol (Synonyme: Dynamic-Symbol)

$$dy = *$$

Definition 10: Dynamische Domäne (Synonyme: Dynamic-Domain)

$$DOM_{an,*} \subset an \times AO \times dy \times AE$$

Wenn ein konkreter Attributwert nicht angegeben werden kann oder soll, kann eine dynamische Domäne verwendet werden.

Dies ist beispielsweise der Fall, wenn Attributwerte erst durch weitere Angaben eines anderen Verhandlungsakteurs bestimmt werden können, wie bei Versicherungen, bei denen Prämien auf dem Alter oder der Fahrerfahrung des Versicherten beruhen.

Weiterführende Anmerkungen:

./.

Definition 11: Verhandlungsobjektnamen (Synonyme: ./.)

$$von \in VON = \{von_1, von_2, \dots, von_n\}$$

Definition 12: Verhandlungsobjekte (Synonyme: Issue)

$$VO \subseteq 2^{DOM}$$

$$vo_{von} \in 2^{DOM}$$

Die Menge der Verhandlungsobjekte ist der Grund für die Führung von Verhandlungen. Über sie soll eine Einigung erzielt werden. Ein Verhandlungsobjekt kann mit einem Namen bezeichnet werden und umfasst ein oder mehrere Attributnamen beziehungsweise Domänen, über die verhandelt wird.

Eymann (Eymann, 2003) fasst Verhandlungsobjekte als Güter auf, über die verhandelt wird. Der hier benutzte Begriff differiert von dieser zunächst intuitiven Auffassung.

In der eigentlichen Verhandlung sind die zu verhandelnden Eigenschaften einer Handelstransaktion und damit unter anderem auch die des dazugehörigen Handelsobjekts relevant. Sind Objekteigenschaften den Beteiligten selbstverständlich oder ausreichend unwichtig, so werden sie in der Verhandlung nicht erwähnt. Trotzdem können sie für die umschließende Abwicklung der gesamten Handelstransaktion von Bedeutung sein.

Ein Beispiel ist: $vo_{\text{Computerkauf}} = ((\text{RAM}, >256, \text{MB}), (\text{Preis}, <,600, \text{EUR}))$

Weiterführende Anmerkungen:

Ist in einem Verhandlungsobjekt mehr als ein Attributname mit der dazugehörigen Domäne enthaltenen, so spricht man von multiattributiven Verhandlungen.

Geht es in einer Verhandlung um mehr als ein Verhandlungsobjekt so spricht man auch von Multiple-Issue-Verhandlungen. Unterschiedliche Verhandlungsobjekte sind thematisch unterschiedlich, d.h. die Namen der Attribute in den Domänenmengen sind unterschiedlich.

In der Folge werden in der Regel nur Verhandlungen mit einem einzelnen Verhandlungsobjekt betrachtet.

3.3.1.1.3 Handels- und Intentionsraum, Handelskonfigurationen

Definition 13: Handelsraum (Synonyme: Deal-Space)

$$HR \subseteq A \times DOM$$

$$HR_a \subseteq a \times DOM$$

$$HR_{a,an} \subseteq a \times DOM_{an}$$

Der Handelsraum umfasst *alle möglichen* Zustände, bei denen ein Handel theoretisch zustande kommen kann.

Der Handelsraum eines Akteurs umfasst *alle möglichen* Zustände, bei denen aus Sicht des Akteurs ein Handel theoretisch zustande kommen kann und der Akteur davon betroffen ist.

Weiterführende Anmerkungen:

Der Handelsraum führt konzeptionell zum Intentionsraum, der mit Hilfe von Beschränkungen die für einen Akteur *akzeptablen* Handelstransaktionen darstellt.

Definition 14: Beschränkungsoperator (Synonyme: Constraint-Operator)

$$bo \in BO = \{<, \leq, =, \neq, \geq, >\}$$

Definition 15: Unäre Beschränkung (Synonyme: Unary-Constraint)

$$ub_{an} = DOM_{an}$$

Definition 16: Binäre Beschränkung (Synonyme: Binary-Constraint)

$$bb_{an^*, an^\wedge} \in BB_{an^*, an^\wedge} = (an^*, bo, an^\wedge)$$

Weiterführende Anmerkungen:

./.

Definition 17: Intentionsraum (Synonyme: Intention-Space)

$$IR_a = (a \times UB) \cup (a \times BB)$$

Der Intentionsraum eines Akteurs stellt mit Hilfe unärer und binärer Beschränkungen die für einen Akteur *akzeptablen* Handelstransaktionen dar.

Weiterführende Anmerkungen:

Im folgenden Beispiel eines Intentionsraumes stehen die ersten beiden Tupel für akzeptable Werte für zwei Attributnamen mit ihren Domänen, das dritte Tupel sagt aus, dass die Zahlung nach Erhalt der Ware erfolgen soll:

((X,(Platte,<,40,GB)),(Y,(RAM,>,128,MB)),(X,(zahl_zeitp,>,liefer_zeitp)))

Die Beschränkungen („Constraints“) begrenzen die Menge der möglichen Kombinationen von Attributnamen mit ihren Domänen auf eine kleinere Menge von angestrebten Kombinationen. Mit Hilfe der obigen Definition eines Intensionsraums kann ein „Constraint-Satisfaction-Problem“, kurz CSP (Willmott, et al., 2000), definiert werden. Die Lösung eines CSP ist die Zuweisung von Attributwerten zu Attributnamen, so dass keine der Beschränkungen verletzt wird. Die Lösung von CSP für stetige Domänen ist weiterhin sehr ineffizient. Daher wird oft von diskreten Domänen ausgegangen, um eine effiziente Lösung zu garantieren. Für die hier diskutierten CSP dürfen die Lösungen keiner der unären und binären Beschränkungen des Intensionsraums widersprechen.

Diese Überlegungen führen in der Folge zu den unten genannten Handelskonfigurationen.

Definition 18: Intensionsraum-Attributnamen (Synonyme: Intention-SpaceAttributes)

$$IRAN_a = \{an \mid \exists ub_{an} \in IR_a \vee bb_{an,an^*} \in IR_a \vee bb_{an^*,an} \in IR_a\}$$

Weiterführende Anmerkungen:

Die Intensionsraum-Attributnamen sind wichtig zur Bestimmung der später aufgeführten Handelskonfigurationen.

Erweiterung der binären Beschränkungen

Als Voraussetzung für die folgenden Optionen und Abhängigkeiten für Attributnamen sind zunächst die binären Beschränkungen zu erweitern:

$$bbk_{an^*,an^\wedge} \in BB_{an^*,an^\wedge} \subset (an^*, AO, AW, AE) \times (an^\wedge, AO, AW, AE)$$

Intensionsraum-Attributoptionen (Synonyme: Optionen, Options) (vgl. (Ströbel, 2003), S. 50)

Optionen im Intensionsraum beruhen auf den binären Beschränkungen. Die Ausprägung eines Attributnamens mit der dazugehörigen Domäne wird dabei im Intensionsraum mit mehreren alternativen Werten für eine andere Domäne kombiniert. Beispiel sind:

$$\begin{aligned} bb_{RAM,mouse}^1 &= (X, RAM, =, 128, MB), (X, (mouse, =, track, \_)) \\ bb_{RAM,mouse}^2 &= (X, RAM, =, 128, MB), (X, (mouse, =, stick, \_)) \end{aligned}$$

Weiterführende Anmerkungen:

./.

Intensionsraum-Attributabhängigkeiten (Synonyme: Abhängigkeiten, Dependencies) (vgl. (Ströbel, 2003), S. 50)

Intentionsraum-Attributabhängigkeiten repräsentieren die Abhängigkeiten von Attributnamen mit den dazugehörigen Domänen untereinander im Intentionsraum.

Beispielsweise könnte der Preis eines Computers von der Größe der Festplatte abhängen:

$$\begin{aligned} \text{bb}_{\text{Festplatte,Preis}}^1 &= (X, (\text{Festplatte}, =, 40, \text{GB})), (X, (\text{Preis}, = 500, \text{EUR})) \\ \text{bb}_{\text{Festplatte,Preis}}^2 &= (X, (\text{Festplatte}, =, 80, \text{GB})), (X, (\text{Preis}, = 550, \text{EUR})) \end{aligned}$$

Weiterführende Anmerkungen:

./.

Definition 19: Handelskonfigurationen (Synonyme: Deal-Configuration)

$$HK_a = \{(an, ao, aw, ae) \mid \forall an \in IRAN_a : \exists (a, an, =, AW, ae) \in IR_a \wedge |AW| = 1\}$$

Ein CSP kann mehrere Lösungen haben. *Handelskonfigurationen* repräsentieren Lösungen des Intentionsraumes eines Akteurs. In einer Handelskonfiguration darf nur exakt ein Wert für einen Attributnamen im Intentionsraum angegeben sein; damit ist die Kardinalität der Mengen der Attributwerte in allen Domänen gleich 1.

Geht es im Intentionsraum nur um das Vorhandensein eines Attributnamens, so kann dieses Tupel für die vorgeschlagene Darstellungsweise umformuliert werden, beispielsweise um anzugeben, dass ein DVD-Laufwerk im Computer enthalten sein soll: $(X, (\text{DVD-Laufwerk}, =, \text{wahr}))$.

Um das formale Modell konsistent zu halten, sollte entgegen Ströbel (Ströbel, 2003) die Einschränkung auf einen Attributwert erhalten bleiben. Wenn es um Wertebereiche für einen Attributnamen geht, kann entweder eine entsprechende Darstellungsweise der Attributwerte gewählt werden oder beispielsweise zwei Attributnamen mit den dazugehörigen Domänen eingeführt werden, die den Wertebereich begrenzen.

Weiterführende Anmerkungen:

./.

3.3.1.2 Der Weg zur Einigung

Im Folgenden wird ohne Einschränkung der Allgemeinheit von einer Situation mit zwei Akteuren ausgegangen, wobei ein Akteur die Rolle eines Verkäufers und der andere die Rolle des Käufers übernimmt.

3.3.1.2.1 Angebotsraum

Definition 20: Angebotsraum (Synonyme: Offer-Space)

$$AR_a \subseteq IR_a \cup IR_x$$

Der Angebotsraum umfasst alle möglichen Angebote, die ein Akteur anderen Akteuren macht. Er beruht einerseits auf dem Intentionsraum des Akteurs. Andererseits können in offenen Umgebungen wie dem Electronic Business Strategien eingesetzt werden, die zu Angeboten führen, welche jedoch nicht direkt zu Handelstransaktionen führen sollen (taktische Scheinangebote).

Weiterführende Anmerkungen:

./.

3.3.1.2.2 Kritik ‚Negotiable-Assignment‘

Ströbel (Ströbel, 2003) führt mit seiner Definition 19 ein ‚Negotiable-Assignment‘ ein, das dazu dienen soll einige Beschränkungen des Intentionsraumes so zu kennzeichnen, dass sie ihre ursprüngliche absolute Gültigkeit verlieren, während die anderen Beschränkungen vollständig erhalten bleiben. Dieses Vorgehen ist konträr zum ursprünglichen Charakter des Intentionsraumes.

Es ist ausreichend, wenn sich der Intentionsraum zeitlich im Verlauf einer Verhandlung verändert. Dies kann durch den sich verändernden inneren Zustand eines autonomen Akteurs geschehen, wie schon oben bei der Charakterisierung von Verhandlungen diskutiert wurde.

Soll tatsächlich eine mögliche zukünftige Veränderungsbereitschaft innerhalb des Intentionsraumes modelliert werden, so wäre die Angabe einer stetigen Funktion, die den Grad der Bereitschaft angibt sicherlich angebrachter als eine Ja/Nein-Kennzeichnung.

Sinnvoller erscheint jedoch das Vorgehen über die Modellierung von Zielen eines Akteurs, die zeitlich gesehen Wirkung auf den Intentionsraum des Akteurs ausüben. In der Folge werden Verhandlungsziele definiert.

3.3.1.2.3 Ziele

Definition 21: Ziele (Synonyme: Verhandlungsziele)

$$Z \subseteq 2^{IR}$$

Ziele werden von den Verhandlungsakteuren angestrebt.

Sie können auch alternative Ziele für dasselbe Verhandlungsobjekt geben.

Weiterführende Anmerkungen:

./.

Definition 22: Ziele eines Akteurs (Synonyme: ./.)

$$Z_a \in Z$$

Die autonomen Verhandlungsakteure streben während der Verhandlung jeweils nach der Erreichung der Menge ihrer eigenen Verhandlungsziele.

Weiterführende Anmerkungen:

Die Ziele können mit dem im Abschnitt 0 eingeführten Präferenzsystem bewertet und geordnet werden.

Definition 23: Ziele einer Rolle (Synonyme: ./.)

$$Z_r \in Z$$

Eine Rolle bedingt zwangsläufig bestimmte Ziele, die durch die Rolle bestimmt werden.

Ein Beispiel dafür ist das Ziel eines Käufers, das Eigentum an einem Handelsgut zu erwerben.

Weiterführende Anmerkungen:

Die Übernahme einer Rolle durch einen Akteur bedingt, dass die Menge der Ziele der Rolle eine Teilmenge seiner eigenen Ziele ist. Ansonsten kann es zu Zielkonflikten kommen.

$$Z_a \cup Z_r = Z_a$$

In Abbildung 3.1 werden wichtige Elemente des Basismodells und ihre Beziehungen untereinander skizziert.

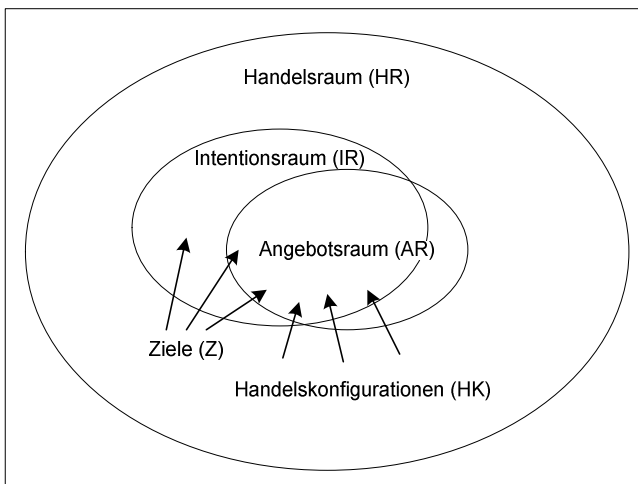


Abbildung 3.1: Wichtige Elemente des Basismodells

Aus dem Handelsraum ergeben sich alle aus der vorliegenden Situation möglichen Handelstransaktionen für einen Akteur. Der Intensionsraum

umfasst die Transaktionen, die der Akteur bereit ist bestimmte Bedingungen anzustreben. Im Intentionsraum liegen die Ziele des Akteurs. Der Angebotsraum umfasst die Angebote, die der Akteur äußert; sie müssen in einer offenen Umgebung nicht im Intentionsraum liegen.

Diese Elemente sind im Zeitverlauf einer Verhandlung veränderlich.

3.3.1.2.4 Strategien

Definition 24: Strategie (Synonyme: ./.)

$$s \in S = \{s_1, s_2, \dots, s_n\}$$

Die Verhandlungsakteure implementieren eine Strategie, mit der sie ihre Ziele zu erreichen suchen.

Konkret bedeutet dies, dass ein Akteur mit Hilfe seiner Strategie die nächste Verhandlungsaktivität bestimmt. Dies kann insbesondere die Auswahl einer Kommunikationsaktivität orientiert am gültigen internen Verhandlungsprotokoll sein. Dabei wird aus der Menge der möglichen Nachrichten eine bestimmte Nachricht mit ihrem Typ ausgewählt und der Inhalt festgelegt.

Weiterführende Anmerkungen:

Einerseits kann die Strategie durch einen menschlichen Verhandlungsakteur realisiert werden, andererseits gibt es Ansätze bei der Automatisierung von Verhandlungen, Strategien auch maschinell zu implementieren. Dabei gibt es neben dem Einsatz von Methoden der künstlichen Intelligenz, auf die hier nicht weiter eingegangen wird, derzeit verschiedene Ansätze, die für maschinelle Strategien wichtige Nutzenfunktion zu bestimmen, wie mathematisch-statistische oder heuristische Verfahren.

Die Behandlung von Strategien kann weitgehend unabhängig von den anderen Elementen des Verhandlungsmodells erfolgen.

Definition 25: Strategie eines Akteurs (Synonyme: ./.)

$$s_a \in S$$

Die Verhandlungsakteure implementieren jeweils eine eigene Strategie.

Weiterführende Anmerkungen:

./.

3.3.1.2.5 Protokolle

Definition 26: Protokollregel (Synonyme: ./.)

$$pr \in PR = \{pr_1, pr_2, \dots, pr_n\}$$

Durch Protokollregeln kann ein Protokoll näher bestimmt werden.

Weiterführende Anmerkungen:

Dabei kann es mehrere Kategorien von Protokollregeln geben. Beispiele dafür sind:

- Regeln, die aussagen, welche Nachrichten auf eine bestimmte Nachricht folgen kann,
- Regeln, durch die Nachrichten der Verhandlungspartner mitteilbar bestimmt werden:
 - Beispiele dafür sind:
 - die maximale Zeitdauer / Beendigungsbedingung der Verhandlung,
 - Vorschriften für den Ein- und Austritt von Verhandlungsparteien vor, nach oder während der Verhandlung.
 - Solche Aspekte wurden teils auch schon als so genanntes Meta-Protokoll bezeichnet (Tu, 2000), obwohl es sich genauso um Regeln für die Kommunikation handelt und nicht um Regeln über diese Regeln.

Für die verschiedenen Kategorien von Protokollregeln können unterschiedliche Beschreibungsformen, wie beispielsweise endliche Automaten, Zustandsdiagramme in UML, Petrinetze oder die Temporallogik angebracht sein.

Definition 27: *Protokoll* (Synonyme: Ablaufprotokoll)

$$p \in P \subseteq 2^{PR}$$

Ein Protokoll für eine Verhandlung bestimmt eine vorgesehene Abfolge von Nachrichten in der Verhandlungskommunikation zwischen den Verhandlungsakteuren.

Die Menge der möglichen Protokolle ist eine Teilmenge der Potenzmenge der Menge der Protokollregeln.

Weiterführende Anmerkungen:

Unter Protokoll ist in diesem Zusammenhang, aufbauend auf einem Zitat aus Merriam-Webster, eine Menge von Regeln für den Ablauf und die mögliche Abfolge der elektronischen Verhandlungskommunikation zu verstehen:

Protocol: "a set of conventions governing the treatment and especially the formatting of data in an electronic communications system" (Merriam-Webster, 2003).

Definition 28: *Internes Protokoll* (Synonyme: Lokales Protokoll)

$$ip \in IP \subseteq P$$

Ein autonomer Verhandlungsakteur kommuniziert mit den anderen Akteuren nach dem nur für ihn selbst gültigen internen Protokoll.

Weiterführende Anmerkungen:

Das interne Protokoll eines Akteurs steht in engem Zusammenhang mit der an anderer Stelle aufgeführten Strategie eines Akteurs. Das interne Protokoll sollte so gestaltet werden, dass es der Strategie entspricht. Mit Hilfe der Strategie wird dann wiederum aus der Menge der möglichen Nachrichten, die sich aus dem internen Protokoll ergibt, die jeweils nächste zu sendende Nachricht ausgewählt.

Wird nur das Zusammenspiel der Akteure in einer Verhandlung betrachtet und nicht die Details eines autonomen Akteurs selbst, so braucht das interne Protokoll nicht weiter detailliert zu werden.

Dieses interne Protokoll wird sich an dem für eine Verhandlung insgesamt gültigen globalen Protokoll orientieren müssen, damit eine möglichst erfolgreiche Kommunikation resultiert.

Definition 29: Globales Protokoll (Synonyme: Verhandlungsprotokoll)

$$gp \in GP \subseteq P$$

Das globale Protokoll in einer Verhandlung gibt die korrekte mögliche Abfolge von Nachrichten zwischen den Verhandlungsakteuren vor.

Ein Beispiel ist das Protokoll einer Holländischen Auktion.

Weiterführende Anmerkungen:

Die Einhaltung oder Verletzung des globalen Protokolls durch einen Akteur kann je nach Konfiguration der Verhandlung von anderen Akteuren sanktioniert werden. Beispielsweise könnte ein Akteur die Verhandlung abbrechen, wenn ein anderer Akteur sich nicht an ein vorher vereinbartes globales Protokoll hält.

Ein globales Verhandlungsprotokoll kann auf verschiedene Weisen etabliert werden:

- Die Verhandlungsaktuere einigen sich vor Verhandlungsbeginn auf ein globales Verhandlungsprotokoll. (Siehe dazu auch das später in dieser Arbeit vorgestellte Bootstrapping-Konzept.)
- Zu einer Verhandlung hinzutretende Verhandlungsaktuere akzeptieren das etablierte Protokoll.
- Wurde kein globales Protokoll etabliert, so ergibt es sich aus den für die jeweils anderen Verhandlungsaktuere ersichtlichen Teilen (Kommunikationsaktivitäten) der internen Protokolle der Verhandlungsaktuere und ist erst vollständig am Ende der Verhandlung vorhanden. (Gegebenenfalls Bildung

von Konventionen als Protokollregeln, an die sich alle Akteure halten.)

Das Zusammenspiel von internem und globalem Protokoll, sowie der Strategie eines Akteurs wird in Abbildung 3.2 verdeutlicht.

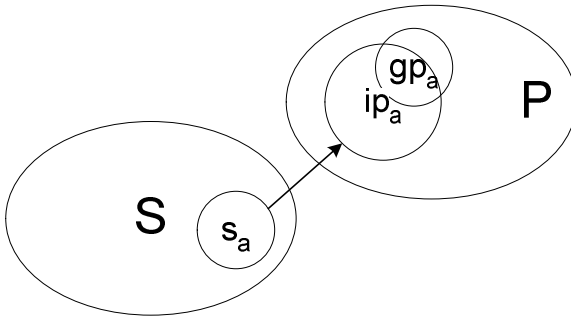


Abbildung 3.2: Zusammenhänge zwischen globalen und internen Protokollen sowie Strategien am Beispiel eines Akteurs

Das globale Protokoll und das interne Protokoll eines Akteurs setzen sich jeweils aus Protokollregeln zusammen. Diese beiden Mengen können sich überschneiden und sollten dies auch, damit sich ein Akteur im vereinbarten Rahmen des Globalen Protokolls verhält. Ein konkretes Internes Protokoll ip setzt sich aus den zugrunde liegenden Protokollregeln zusammen. Durch das Protokoll ip wird in jeder Kommunikationssituation bestimmt, welche Nachrichten gesendet werden können. Die Strategie s wählt daraus eine bestimmte Nachricht aus, die zur Erreichung der Ziele des zugehörigen Akteurs führt.

3.3.1.2.6 Angebote

Definition 30: Instanz des Angebotsraums (Synonyme: Angebot, Gebot, Anfrage, Offer-Space-Instance, Offer, Bid, Ask)

$$ar_a \in AR_a$$

Weiterführende Anmerkungen:

./.

Definition 31: Angebotsart (Synonyme: ./.)

$$aa \in AA = \{AzK, AzV\}$$

Es gibt Angebote zum Kauf (AzK) und Angebote zum Verkauf (AzV).

Weiterführende Anmerkungen:

./.

Definition 32: Angebotskennzeichnung (Synonyme: ./.)

$$AK = ar \rightarrow AA$$

Weiterführende Anmerkungen:

./.

Definition 33: Angebot zum Kauf (Synonyme: ./.)

$$AZK_a = AR_a : AK(azk_a) = AzK$$

Weiterführende Anmerkungen:

./.

Definition 34: Angebot zum Verkauf (Synonyme: ./.)

$$AZV_a = AR_a : AK(azv_a) = AzV$$

Weiterführende Anmerkungen:

./.

Definition 35: Angebotsinstanz-Attributnamen (Synonyme: ./.)

$$ARIAN_{ar} = \{an \mid \exists ub_{an} \in AR \vee \exists bb_{an,an^*} \in AR \vee \exists bb_{an^*,an} \in AR\}$$

Weiterführende Anmerkungen:

./.

3.3.1.2.7 Vergleich von Angeboten

In der Folge wird die Bindung bzw. Zuordnung der zu definierenden Begriffe zu den Akteuren aufgelöst, um gemeinsame Räume mit definierten Bedeutungen ableiten zu können. Abbildung 3.3 zeigt eine Übersicht:

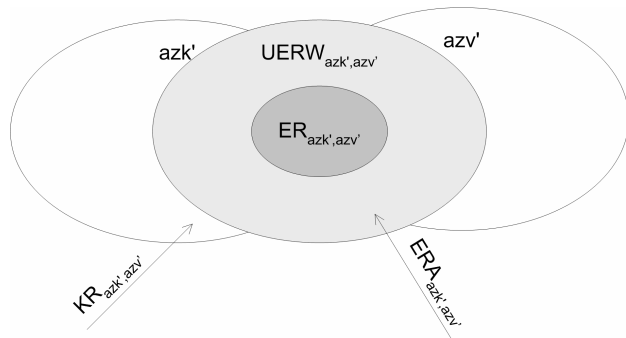


Abbildung 3.3: Angebots- und Einigungsmengen

Definition 36: Anonyme Instanz des Angebotsraumes (Synonyme: Anonymous-Offer-Space-Instance)

$$anar' = \{(an, ao, aw, ae) \mid (a, (an, ao, aw, ae)) \in AR_a\}$$

Weiterführende Anmerkungen:

./.

Definition 37: Einigungsraum (Synonyme: Agreement-Space)

$$ER_{azk', azv'} = azk' \cap azv'$$

Der aktuelle Einigungsraum besteht aus den Elementen der Handelskonfigurationen, die den Beschränkungen der Angebotsräume der Akteure entsprechen.

Weiterführende Anmerkungen:

./.

Definition 38: Einigungsraum der Attribute (Synonyme: Negotiation-Space)

$$ERA_{azk', azv'} = azk' \cup azv': ARIAN_{azk'} = ARIAN_{azv'}$$

Im Einigungsraum der Attribute stimmen die Akteure über die Attributnamen überein.

Weiterführende Anmerkungen:

./.

Definition 39: Uneinigkeitsraum der Attributwerte (Synonyme: Disagreement-Space)

$$UERW = ERA - ER$$

Im Uneinigkeitsraum der Attributwerte differieren die Attributwerte der Akteure. Der UERW ist Teilmenge des ERA.

Weiterführende Anmerkungen:

./.

Definition 40: Konfliktraum (Synonyme: Conflict-Space)

$$KR_{azk', azv'} = (azk' \cup azv') - ER_{azk', azv'}$$

Im Konfliktraum sind sich die Akteure nicht einmal über die zu verhandelnden Attributnamen einig.

Weiterführende Anmerkungen:

./.

Definition 41: Streitpunkte (Synonyme: Issues)

$$SP_{azk',azv'} = \{ir \mid (ir \in azk' \vee ir \in azv') \wedge ir \notin ER_{azk',azv'}\}$$

Die aus den konkreten Angeboten zum Verkauf und Kauf hervorgehenden Streitpunkte sind die Angebotselemente, die nicht Teil des Einigungsraumes sind.

Weiterführende Anmerkungen:

Im Verlauf der Verhandlung können natürlich durch die nachfolgenden Angebote wiederum völlig neue Elemente eingebracht werden.

Präferenzen

Sobald ein Einigungsraum festgelegt ist und mindestens ein Element enthält, können die Akteure die möglichen Handelskonfigurationen im Hinblick auf ihren Nutzen und ihre weitere Verhandlungsstrategie evaluieren.

Dazu sei angenommen, dass ein Akteur jeder dieser Handelskonfigurationen einen Wert zuordnen kann, wobei unterschiedliche Handelskonfigurationen auch denselben Wert erhalten können. Daraus ergibt sich eine Ordnungsrelation.

Auch die Ziele eines Agenten können mit dem Präferenzsystem bewertet werden.

Definition 42: Handels-Einschätzungswerte (Synonyme: Deal-Value)

$$hew \in HW = [0 .. 1]$$

Definition 43: Handels-Konfigurationswert (Synonyme: Deal-Configuration-Value-Assignment)

$$HKW_a = a \times HK \rightarrow HW$$

Weiterführende Anmerkungen:

./.

Definition 44: Präferenz (Synonyme: ./.)

$$PRF_a \subset HK \times HK$$

$$PRF_a = \{(hk^*, hk^\wedge) \mid hkw_{a,hk^*} > hkw_{a,hk^\wedge} \wedge (hk^*, hk^\wedge), (hk^\wedge, hk') \in PRF_a\}$$

Weiterführende Anmerkungen:

./.

3.3.1.2.8 Kritik ‚Fixed / Flexible-Deal-Value‘

Ströbel ((Ströbel, 2003), S. 53f, Definitionen 38 und 39) führt feste und variable Handels-Konfigurationswert-Zuweisungen (HKW) ein. Dies

ähnelt der Unterteilung von Attributen bei De Paula et al. (De Paula, et al., 2000) in Produkt- und Verhandlungsattribute.

Dabei wird jedoch die offene Umgebung, die für Verhandlungen im Electronic Business unabdingbare Voraussetzung ist, nicht berücksichtigt. In einer offenen Umgebung kann es keine vorhersehbaren festen Handels-Konfigurationswert-Zuweisungen geben, da sie auch von den jeweiligen anderen autonomen Akteuren oder dem Zeitverlauf abhängen können.

Als Beispiel nennt Ströbel den Ort eines Akteurs als nicht verhandelbares Attribut und impliziert damit den Anlieferungs- oder Abholungsort einer Ware, da nur diese Orte für die genannte Verhandlung relevant wären. Es ist jedoch durchaus möglich, dass die Akteure auf eigene Kosten zusätzliche Logistik übernehmen, um in den übrigen Punkten Verhandlungsvorteile zu erzielen.

De Paula et al. (De Paula, et al., 2000) nennt als Beispiele Länge, Gewicht oder Farbe eines Gutes. Doch ist es durchaus möglich, dass ein Verkäufer bei attraktivem Kaufpreisangebot auch ein Gut besorgt, das eine andere Farbe hat, als seine auf Lager liegenden Verkaufsgüter.

3.3.1.2.9 Opposition

Definition 45: Opposition (Synonyme: ./.)

$$OPS_{Käufer,Verkäufer,hk} = hkw_{Käufer,hk} \times hkw_{Verkäufer,hk}$$

Der Grad der Uneinigkeit zwischen den Akteuren kann mit Hilfe einer Maßeinheit geschätzt werden, die von Kersten und Noronha vorgeschlagen wurde (Kersten, et al., 1999).

Weiterführende Anmerkungen:

./.

3.3.1.2.10 Verläufe von Verhandlungen

Zu Beginn einer Verhandlung geben die beteiligten Akteure durch den Austausch von Angeboten ihre implizite Verhandlungsabsicht kund. Jeder Akteur verfügt dazu insbesondere über seinen jeweils aktuellen Intentionssraum und Angebotsraum.

Durch den Austausch der Angebote werden die Verhandlungsobjekte immer näher bestimmt. Schließlich bildet sich bei einer Einigung ein Einigungsraum, der Handelskonfigurationen enthält, denen alle Akteure zustimmen. Stimmen alle Akteure einer Handelskonfiguration zu, so ist die Verhandlung abgeschlossen und der Handel kann ausgeführt werden.

Kommt es zum Abbruch der Verhandlung, so kann die Kommunikation entweder völlig eingestellt werden oder es gibt eine Übereinkunft der Parteien, die Verhandlung abubrechen. Im letzteren Falle können auch

noch weitere Informationen, beispielsweise über eine eventuelle Vertagung der Verhandlung und spätere Fortsetzung ausgetauscht werden.

3.3.1.3 Distributive und Integrative Verhandlungen

Oft wird zwischen distributiven und integrativen Verhandlungen unterschieden. Aufgrund des Basismodells können diese Begriffe definiert werden.

Definition 46: Distributive Verhandlungen (Synonyme: ./.)

$$\forall hk^*, hk^\wedge \in ER_{X,Y} : hkw_{X,hk^*} > hkw_{X,hk^\wedge} \rightarrow hkw_{Y,hk^*} < hkw_{Y,hk^\wedge}$$

Führt eine neue Handelskonfiguration zur verbesserten Einschätzung durch einen Akteur und zu einer verschlechterten Einschätzung beim anderen Akteur, so spricht man von einem distributiven Verhandlungsschritt.

Weiterführende Anmerkungen:

./.

Definition 47: Integrative Verhandlungen (Synonyme: ./.)

$$\exists hk^* \in ER_{X,Y} \leq hk \sim ER_{X,Y} : hkw_{X,hk^*} > hkw_{X,hk^\wedge} \wedge \rightarrow hkw_{Y,hk^*} > hkw_Y$$

Führt eine neue Handelskonfiguration zur verbesserten Einschätzung durch beide beteiligten Akteure, so spricht man von einem integrativen Verhandlungsschritt.

Weiterführende Anmerkungen:

./.

Neben den nun vorgestellten Elementen des Basismodells, aus denen sich eine Verhandlung zusammensetzt, ist natürlich auch der weiter gefaßte zeitliche Ablauf des Verhandlungsprozesses, der im nächsten Abschnitt betrachtet wird, von besonderer Bedeutung.

3.3.2 Ablauf einer Verhandlung

Die in dieser Arbeit betrachteten Verhandlungen im Electronic Business sind der Vereinbarungsphase von Handels- bzw. Markttransaktionen zuzuordnen. Eine Markttransaktion gliedert sich grob in die drei Phasen Information, Vereinbarung und Abwicklung, wie auch schon oben im Kapitel 2 zum Electronic Business ausführlich diskutiert.

Die Vereinbarungsphase wird weiter unterteilt in die folgenden drei Phasen, um die eigentliche Verhandlung nicht mit anderen im Grunde typfremden Aktivitäten zu belasten:

[I.] In der *Intentionsphase* werden alle nötigen Vorbereitungen für eine Verhandlung getroffen. Dazu zählen beispielsweise:

Herstellung des Kontakts zwischen den als Ergebnis der Informationsphase passenden („Matching“) Partnern,

Etablierung einer (gegebenenfalls sicheren) Kommunikationsmöglichkeit,

eine grobe Bestimmung der Verhandlungsobjekte.

[V.] Eine weitere Unterteilung der *Verhandlungsphase* wird weiter unten vorgenommen.

[A.] Die *Abschlussphase* kann beispielsweise die folgenden Aktivitäten enthalten:

- Ausarbeitung eines bindenden Vertrages mit Sanktionsmöglichkeiten bei Nichteinhaltung,
- Sicherung der Dokumentation der Verhandlung, die gegebenenfalls im Rahmen eines Verhandlungsmonitorings angefertigt wurde,
- Einleitung einer neuen Verhandlung, falls alle beteiligten Parteien versuchen wollen eine bessere Einigung, möglicherweise ausgehend von der bereits erreichten Einigung, zu erzielen.

Eine alternative Einteilung bieten Kersten et al. (Lo, et al., 1999), (Kersten, et al., 2000) an. Sie unterteilen den Ablauf einer Verhandlung in die drei Phasen ‚Preparation‘, ‚Conduct‘ und ‚Post-Settlement‘:

- ‚*Preparation*‘: Zunächst wird die Verhandlungssituation analysiert, indem die einzelnen Parteien und ihre Interessen identifiziert werden. Es wird geklärt, über welche Verhandlungsobjekte verhandelt werden soll.
- ‚*Conduct*‘: Es werden solange Nachrichten und Angebote zwischen den Verhandlungspartnern ausgetauscht, bis eine Einigung erzielt wird beziehungsweise die Verhandlung abgebrochen wird.
- ‚*Post-Settlement*‘: Diese Phase bietet die Möglichkeit, nach einer schon erzielten Einigung, die Verhandlung erneut aufzunehmen, um eventuell ein besseres Verhandlungsergebnis zu erzielen.

Zugunsten einer detaillierten Aufteilung wird jedoch die zuerst vorgestellte Einteilung in Intentions-, Verhandlungs- und Abschlussphase weiter verwendet.

Auch die Verhandlungsphase kann wiederum weiter unterteilt werden. Dazu entwickelt Haft (Haft, 2000) ein idealisiertes ‚rationales Verhandlungsmodell‘ bei dem er sechs Phasen identifiziert.

Die *Eröffnungsphase*, in der es um die Herstellung einer persönlichen Beziehung zwischen den Partnern geht, um die spätere persönliche Kommunikation zu erleichtern.

Die *Rahmenphase* dient der Klärung und Einigung über die äußeren Umstände einer Verhandlung. Dazu zählen insbesondere die Kompetenzen und Zuständigkeiten der Verhandlungsparteien und der zur Verfügung stehende Zeitrahmen.

In der *Themenphase* wird das bisher nur grob definierte Thema der Verhandlung weiter geklärt, also der Gegenstand der Verhandlung.

Die *Sachverhaltsphase* dient zum Austausch von Informationen bezüglich des Sachverhalts des zu verhandelnden Problems, da die beteiligten Parteien durchaus eine unterschiedliche Sicht auf den Sachverhalt haben können. Haft (Haft, 2000) nennt diese Phase ‚Informationsphase‘. Sie wurde zur besseren Unterscheidung zu anderen Phasen in dieser Arbeit unter dem Namen ‚Sachverhaltsphase‘ übernommen.

Die *Argumentationsphase* ist die wichtigste Phase der Verhandlung. In dieser Phase werden Strukturen für die zu verhandelnden Probleme beziehungsweise Verhandlungsobjekte gebildet und in einzelnen Arbeitsabschnitten behandelt. Dazu werden relevante Aspekte gesammelt, geordnet und entschieden. Aufgrund dieser Argumentation kann es zu Zwischenergebnissen kommen.

In der *Entscheidungsphase* wird aus den Teilergebnissen der Argumentationsphase das Endergebnis abgeleitet.

Diese Einteilung ist, wie auch schon durch Haft (Haft, 2000) angesprochen, ein idealisiertes Modell und kann in der Realität andere Ausprägungen annehmen. Es dient jedoch als Leitbild und kann für die Erstellung von Systemen, die weiterhin Flexibilität bieten, herangezogen werden.

Für den Ablauf einer elektronischen Verhandlung im Electronic Business lässt sich die folgende Einordnung und Phasenstrukturierung zusammenfassend ableiten (siehe Tabelle 3.1):

Tabelle 3.1: Einordnung und Strukturierung von Verhandlungsaktivitäten im Electronic Business

Markttransaktion	Vereinbarungsphase	Verhandlungsphase
Informationsphase		
Vereinbarungsphase	Intentionsphase	
	Verhandlungsphase	Eröffnungsphase

		Rahmenphase
		Themenphase
		Sachverhaltsphase
		Argumentationsphase
		Entscheidungsphase
	Abschlussphase	
Abwicklungsphase		

Eine *Markttransaktion* gliedert sich in die Informationsphase, die *Vereinbarungsphase* und Abwicklungsphase (siehe erste Spalte). Die *Vereinbarungsphase* unterteilt sich wiederum in die Intentionsphase, die eigentliche *Verhandlungsphase* und die Abschlussphase (siehe mittlere Spalte). Letztendlich lassen sich die Aktivitäten in der *Verhandlungsphase* in eine Eröffnungsphase, eine Rahmenphase, eine Themenphase, eine Sachverhaltsphase, eine Argumentationsphase und in eine Entscheidungsphase untergliedern (siehe letzte Spalte).

Ergänzend zur obigen Darstellung der einzelnen Verhandlungsphasen, werden diese nun im Einzelnen im Kontext elektronischer Verhandlungen im Electronic Business adaptiert und dargestellt.

Eröffnungsphase:

Das Medium Internet erschwert die Pflege von persönlichen Beziehungen gegenüber der direkten Begegnung wesentlich.

Werden die Verhandlungspartner durch automatisierte Akteure realisiert, entfallen diese persönlichen Beziehungen und damit die Eröffnungsphase oder sie sind gegebenenfalls entsprechend nachzubilden, falls dieses als förderlich für die Verhandlung angesehen wird.

Menschlichen Akteuren können im Rahmen der Verhandlungsunterstützung Applikationen angeboten werden, mit denen sie entsprechend interagieren können. Dazu zählen insbesondere auch Applikationen des ‚Computer-Support-Cooperative-Work‘ (CSCW), wie gemeinsames Dokumentenmanagement, Chaträume, E-Mail oder ‚Instant-Messaging‘.

Rahmenphase:

Die Rahmenphase dient, wie oben angesprochen, der Klärung und Einigung über die äußeren Umstände der Verhandlung. Die Rahmenphase wäre bei der Einteilung von Kersten et al. (Lo, et al., 1999) in der Preparation-Phase einzuordnen.

Zur Klärung von Kompetenzen und Zuständigkeiten der Verhandlungsparteien sind gegebenenfalls Techniken der Identifikation, Authentifi-

fizierung und Autorisierung einzusetzen. Ist nicht die gesamte Kommunikation schon gesichert, so sind spätestens hier entsprechende Maßnahmen zu treffen.

In dieser Arbeit stehen jedoch die inhaltlichen Rahmenbedingungen einer Verhandlung im Vordergrund. Dazu zählen beispielsweise der Zeitrahmen der Verhandlung oder ein explizit zu verwendendes Protokoll. Dazu kann das Konzept des Bootstrappings eingesetzt werden, mit dem vor der eigentlichen Verhandlung genau diese Parameter dynamisch und flexibel geklärt werden können. Damit hat das Bootstrapping besonders intensive Auswirkungen auf die nachfolgenden Argumentations- und Entscheidungsphasen.

Das Bootstrapping wird detailliert in Kapitel 4 behandelt.

Themenphase:

Die zu verhandelnden Themen, also die Verhandlungsobjekte (VO), werden im Allgemeinen durch die überordneten Phasen Informationsphase und dann die Intentionsphase vorbereitet. Jedoch sind oft Konkretisierungen notwendig. Die Verhandlungsobjekte können im Verlauf der Verhandlung auch modifiziert werden, beispielsweise um ein weiteres Attribut in die Verhandlung einzubeziehen und so eine integrative Verhandlung zu erleichtern. Dazu dient die Themenphase.

Sachverhaltsphase:

In der Sachverhaltsphase werden die sachbezogenen Sichten der Verhandlungsparteien auf das zu verhandelnde Problem abgeglichen. Das heißt, es werden nicht nur Informationen zu den Verhandlungsobjekten (VO), wie in der Themenphase ausgetauscht, sondern auch über die Sachlage, durch die die VO bedingt werden. Darauf können die Verhandlungsparteien verschiedene Sichten haben.

In der Sachverhaltsphase können sich, philosophisch gesehen, der Nominalismus und der Begriffsrealismus gegenüberstehen. Der Begriffsrealismus geht von der Realität von Begriffen, also von der realen Existenz objektiver Wahrheiten aus. Diese Sichtweise liegt beispielsweise den deutschen gerichtlichen Prozessordnungen zu Grunde. Der Nominalismus verneint die Existenz objektiver Wahrheiten. Die Wahrheit wird als subjektiv, relativ und als bezogen auf den Menschen aufgefasst. Diese Auffassungen liegen beispielsweise den angelsächsischen gerichtlichen Prozessordnungen zugrunde.

Argumentationsphase:

Die Argumentationsphase dient dem Austausch der eigentlichen Argumente, wobei dies orientiert an einer aufzubauenden Strukturierung des

Problems geschehen sollte. Ebenfalls ist die Dokumentation von erreichten Zwischenergebnissen wichtig.

Bei Kersten et al. (Kersten, et al., 2000) werden die Aktivitäten der Argumentationsphase in die ‚Conduct‘-Phase eingeordnet.

Entscheidungsphase:

Auf der Basis der in der Argumentationsphase erreichten Teilergebnisse wird in der Entscheidungsphase das Endergebnis abgeleitet.

Diese Phasen müssen gemäß den jeweiligen Situationen nicht unbedingt nur einmal, vollständig oder streng sequentiell durchlaufen werden. Angepasst an die jeweiligen Situationen können einzelne Phasen übersprungen oder schnell vereinfacht durchlaufen werden sowie gegebenenfalls wiederholt werden.

Ergeben sich für die Verhandlungspartner *neue Situationen*, beispielsweise durch bedeutende externe Informationen, so kann in Übereinstimmung mit den anderen Verhandlungspartnern *in frühere Phasen zurückgegangen* und der Ablauf wiederholt werden.

Lo und Kersten (Lo, et al., 1999) schlagen eine Phase des ‚Post-Settlements‘ nach der eigentlichen Verhandlung vor, welche die Möglichkeit bietet, nach einer schon erzielten Einigung, die Verhandlung erneut aufzunehmen, um eventuell ein besseres Ergebnis zu erzielen. Dies ist ein Spezialfall des oben vorgeschlagenen Vorgehens. Zu solchen Situation kann es kommen, wenn die für die Verhandlung relevanten Aspekte nur lückenhafte Berücksichtigung gefunden haben, um zügiger zu einer ersten Einigung zu kommen. Die im weiteren Verlauf noch detailliert vorgestellte SNT-Methode steht zu diesem Vorgehen in Beziehung.

Kommt es zu einem *Abbruch* der Verhandlung, ist auf allen Ebenen der Interaktion ein *transaktionales Beenden* für die meisten Beteiligten von Vorteil.

Um das Ergebnis einer Verhandlung zu beschreiben, findet oft der Begriff des Vertrages Verwendung (Merz, et al., 1998), wobei teilweise auch berücksichtigt wird, dass sich dieses Ergebnis im Laufe der Verhandlung erst entwickelt (vgl.(Schoop, et al., 2001)).

Der Begriff Vertrag fokussiert jedoch einerseits nur den Zustand nach einer Verhandlung und andererseits die rechtliche Sicht auf den Abschluss der Verhandlung. Ein Vertrag hat eine Rechtsgrundlage, er enthält genaue Angaben über die Vertragspartner und die Vertragsgegenstände, die das Resultat der Verhandlung sind.

Es ist jedoch wichtig, den aktuellen Verlauf und Zustand der Übereinstimmungen zwischen den Verhandlungsparteien zu verfolgen und dokumentieren zu können (Quix, et al., 2002). Dies entspricht auch der

weithin akzeptierten Anforderung an Verhandlungen, erreichte Zwischen-einigungen zu konsolidieren (Haft, 2000).

Wird der Zustand des im formalen Basismodell eingeführten Einigungsraums archiviert, so liegt eine entsprechende Dokumentation vor. Der aus einer erfolgreichen Verhandlung resultierende Einigungsraum bildet dann die Grundlage für einen späteren Vertrag.

Eine Infrastruktur für Verhandlungen muss die in diesen Phasen vorkommende Kommunikation ermöglichen. Verhandlungsunterstützungssysteme sollten den Ablauf einer Verhandlung in dieser Form unterstützen können.

Bei dem bis hier in dieser Form vorgestellten Verhandlungsmodell handelt es sich um ein breit angelegtes Basismodell.

Die weitere Konkretisierung und zum Teil Spezialisierung erfolgt in den noch folgenden Abschnitten und Kapiteln.

Dabei kann das Basismodell als konzeptionelle Grundlage für die zu entwickelnden Verhandlungsinfrastruktur und -systeme dienen.

3.4 Verhandlungsstrategien

Eine Verhandlungsstrategie dient dazu, wie schon oben im Basismodell eingeführt, über die Verhandlungsaktivitäten und -nachrichten im Rahmen des vereinbarten Protokolls zu entscheiden. Diese Strategie kann durch *menschliche* Benutzer oder *maschinell* ausgeführt werden.

Bei der maschinellen Ausführung ist das Anliegen meist die Automatisierung des Verhandlungsprozesses. Derzeit sind solche automatisierten Verhandlungen trotz aller Bestrebungen in offenen verteilten Umgebungen jedoch weder effizient noch effektiv durchführbar. Dies liegt einerseits in den extrem vielfältigen und sich unvorhersehbar verändernden Anforderungen begründet, andererseits in den bislang für solche Situationen nicht verfügbaren Methoden der künstlichen Intelligenz. Trotzdem gibt es hier intensive Bemühungen. Ein Beispiel für einen solchen Ansatz findet sich bei Tu (Tu, 2000), der genetische Algorithmen zur Implementation von automatisierten Verhandlungsstrategien betrachtet.

Zur Unterstützung des menschlichen Benutzers bei der Bildung und Ausführung von entsprechenden Strategien sind weitere Lösungen, wie beispielsweise Ergebnisse der 'Negotiation-Analysis', entwickelt worden.

3.4.1 ‚Negotiation-Analysis‘

‚Negotiation-Analysis‘ ist eine Kombination von ‚Game-Theory‘ und ‚Decision-Analysis‘. Sie beschäftigt sich mit der Art der Konfliktlösungen durch Verhandlungen, analysiert deren Prozesse und erarbeitet mögliche Verhandlungsstrategien.

3.4.1.1 Grundlagen der ‘Negotiation Analysis’

In der ‚Game-Theory‘ werden Spielsituationen auf die Frage hin analysiert, wie man zu einem optimalen Ergebnis gelangen kann. Eine Voraussetzung ist dabei ein optimales Verhalten aller Spieler. Gleichzeitig analysiert die ‚Game-Theory‘ interaktive Prozesse von mehreren Spielern und bietet präskriptive Modelle, mit denen es die Ergebnisse des Spiels vorhersagt.

Im Gegensatz dazu beschäftigt sich die ‚Decision-Analysis‘ hauptsächlich damit, Entscheidungsprobleme asynchron aus der Sicht eines einzelnen Entscheidungsträgers zu modellieren. Dabei ist das zugrunde liegende Prinzip das Streben des Entscheidungsträgers, seinen Nutzen (‚Utility‘) zu maximieren.

‚Negotiation-Analysis‘ gehört eng zu diesen beiden Forschungsfeldern. Sie bietet zur Unterstützung des Verhandlungsprozesses außer einer deskriptiven Analyse auch präskriptive Modelle. Es wird davon ausgegangen, dass sich alle Parteien intelligent und zielgerichtet verhalten, jedoch nicht vollkommen rational, wie in der ‚Game-Theory‘ vorausgesetzt wird, (Sebenius, 1992).

Die ‚Negotiation-Analysis‘ geht davon aus, dass das Ziel der Verhandlung ein pareto-optimales Ergebnis ist. Folgende Sachverhalte sind dabei mit zu betrachten: Jeder Verhandlungsteilnehmer muss sich zunächst über seine eigenen Ziele und Präferenzen bewusst sein. Diese lassen sich in der so genannten Nutzenfunktion (‚Utility-Function‘) abbilden. Die Nutzenfunktion ist demnach eine subjektive Beurteilung des Wertes von Verhandlungsergebnisalternativen, beispielsweise auf einer numerischen Skala von 0 bis 100. Die niedrigste Bewertung kennzeichnet das am wenigsten gewünschte Ergebnis und die höchste Bewertung das am meisten gewünschte. Alle möglichen Ergebnisse bilden das ‚Set-Of-Feasible-Agreements‘ oder ‚Utility-Set‘. Nehmen z.B. zwei Partner an der Verhandlung teil, so lassen sich aus den beiden ermittelten Nutzenfunktionen alle gemeinsam möglichen Verhandlungsergebnisse ermitteln, wie in Abbildung 3.4 dargestellt.

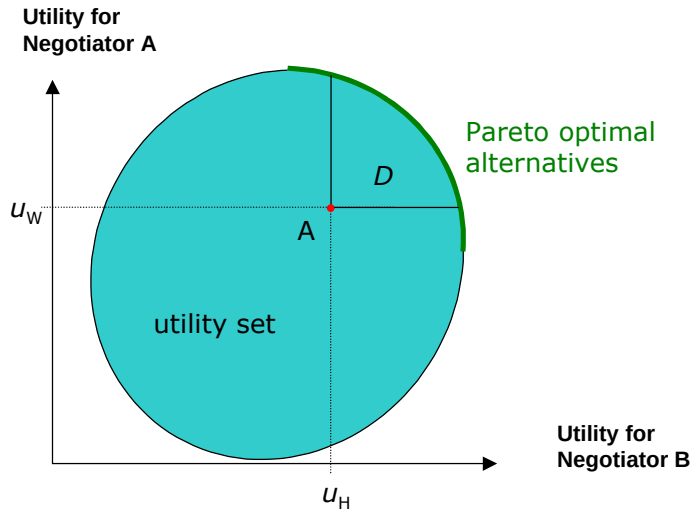


Abbildung 3.4: Utility Set und Pareto-Optimalität (nach (Ehtamo, et al., 2002))

Das Nash-Equilibrium, dargestellt durch den Punkt A, stellt eine Standardlösung in der Spieltheorie dar. Nash-Gleichgewichte sind Strategiekombinationen bzw. Verhandlungsergebnisse. Sie stellen die wechselseitig besten Antworten bei gegebener Strategie der anderen Spieler dar. Ein Verhandlungsergebnis kommt zu einem solchen Gleichgewicht, wenn kein Spieler, sofern alle Spieler bei ihrer Strategie bleiben, seinen erwarteten Nutzen weiter maximieren kann.

Das folgende Beispiel in Tabelle 3.2, entlehnt aus Thelen (Thelen, 2000), verdeutlicht dies:

Tabelle 3.2: Beispiel zum Nash-Equilibrium

Firma A	Firma B			
		p=1	p=2	p=3
	p=1	(0,0)	(50,-10)	(40,-20)
	p=2	(-10,50)	(20,20)	(90,10)
	p=3	(-20,40)	(10,90)	(50,50)

Zwei Firmen haben die Wahl, ihr Produkt zum Preis von $p=1$, 2 oder 3 Einheiten anzubieten. Aus der Tabelle sind die jeweiligen Gewinne nach Abzug aller anfallenden Kosten zu entnehmen. Es ist ersichtlich, dass der Gewinn einer Firma sowohl von der eigenen Wahl des Preises p , als auch von der Wahl der anderen Firma abhängig ist.

Das einzige Nash-Equilibrium wäre in diesem Fall die Lösung ($p=1$, $p=1$). Denn bei den anderen Ergebnissen könnte jeweils eine der beiden

Firmen nur durch eine Preissenkung profitieren, wenn der Verhandlungspartner bei seiner Wahl bleibt.

Wie man an diesem Beispiel sehen kann, muss ein Nash-Equilibrium nicht immer ein gutes, bzw. das beste Ergebnis sein. Es ist in diesem Beispiel ein Gleichgewichtszustand mit einem besonders niedrigen Preis für beide Parteien, jedoch mit sehr wenig Profit. Bessere Ergebnisse wären hier die pareto-optimalen Lösungen ($p=3, p=2$), ($p=2, p=3$) und ($p=3, p=3$). Ein pareto-optimales Ergebnis ist ein Ergebnis, bei dem sich kein Spieler verbessern kann, ohne dass sich ein anderer Spieler verschlechtert. Dies veranschaulicht das folgende Beispiel in Tabelle 3.3, angelehnt an das weithin bekannte Prisoners-Dilemma:

Tabelle 3.3: Beispiel zum Nash-Equilibrium (2 – Prisoners-Dilemma)

Firma A	Firma B	
	p=1	p=2
p=1	(10,10)	(50,0)
p=2	(0,50)	(40,40)

In diesem Fall wäre die Lösung ($p=1, p=1$) ein Nash Equilibrium. Diese Lösung ist jedoch keineswegs optimal, da sowohl die Lösungen ($p=2, p=1$) bzw. ($p=1, p=2$) und ($p=2, p=2$) einen höheren Profit versprechen. Ein solches Szenario entsteht häufig durch distributives Verhandeln, auch ‚Competitive-Bargaining‘ genannt. Beide Verhandlungspartner versuchen um jeden Preis, das für sie am besten erscheinende Ergebnis zu erreichen, hier einen Gewinn von 50. Sie scheitern, weil der Verhandlungspartner die gleiche Strategie der Gewinnmaximierung verfolgt. Durch ein kooperierendes Verhalten jedoch ist für beide Parteien eine bessere Lösung möglich. Die Lösung ($p=2, p=2$) stellt also ein optimales Ergebnis einer integrativen Verhandlung dar, ein pareto-optimales Ergebnis. Obwohl keine der beiden Parteien den größtmöglichen Profit, in diesem Fall 50, erlangt, ergibt die Summe der beiden Profite den größtmöglichen Profit innerhalb des ‚Utility-Set‘ ($40+40=80$).

In der ‚Negotiation-Analysis‘ existieren verschiedene Methoden, um ein Pareto-optimales Ergebnis zu ermitteln. Diese lassen sich nach (Kettunen, 1999) in zwei Hauptgruppen unterteilen:

- auf Konzessionen basierende Methoden (‚Concession-Based-Methods‘)
- Methoden mit gemeinsamen Verhandlungstexten (‚Single-Negotiating-Text-Methods, SNT-Methods‘)

Bei den auf Konzessionen basierenden Methoden nähern sich die Verhandlungspartner einander immer mehr an, indem sie sich gegenseitig

Zugeständnisse (,Concessions') machen. So gelangen sie schließlich zu einer gemeinsamen Lösung.

Bei den SNT-Methoden gehen die Verhandlungspartner von einem vorläufigen Verhandlungsergebnis aus (,Tentative-Agreement'), von dem aus sie nach gemeinsamen Lösungen suchen.

Weiterhin lassen sich jeweils die auf Konzessionen basierenden Methoden als auch die SNT-Methoden weiter unterteilen:

- die auf Nutzenfunktionen basierenden Methoden (,Value-Function-BasedMethods'),
- interaktive Methoden (,Interactive-Methods')

Methoden, die auf Nutzenfunktionen basieren, erfordern das spezifische Erstellen dieser Funktionen (,Utility-Functions'), d.h. eine Beschreibung der Nützlichkeit einzelner Attribute und deren Werte (,Options') für den Benutzer. Dieses Erstellen kann bei realen Szenarien äußerst aufwendig sein. Die ,Utility-Functions' der verhandelnden Parteien sind dann Grundlage für eine Berechnung pareto-optimaler Lösungen.

Interaktive Methoden erfordern nicht von den Parteien, vor der Verhandlung, ihre Präferenzen genau festzulegen. Vielmehr ergeben sich diese in einem interaktiven Prozess während der Verhandlung. Diese Methoden eignen sich daher besonders für Fälle, in denen die zur Verhandlung stehenden Objekte nur unzureichend definiert sind. Außerdem ist es oft schwierig für die Verhandelnden, ihre Präferenzen genau festzulegen. Dabei ist auch zu bedenken, dass bei der Erstellung einer ,Utility-Function' diese Präferenzen in numerischer Form vorliegen müssen oder aber so in ein System eingegeben werden müssen, dass dieses daraus eine ,Utility-Function' erstellen kann.

3.4.1.2 Konzessionsbasierte Methoden

Die Methode des Einschränkungsvorschlags (,Constraint-Proposal-Method') (vgl. (Kettunen, 1999)). und die auf Nutzenfunktionen basierende Methode (,Value-Function-Based-Method') sind Beispiele für auf Konzessionen basierende Methoden. Sie werden in der Folge vorgestellt.

3.4.1.2.1 Methode des Einschränkungsvorschlags

Bei der ersten Methode des Einschränkungsvorschlags schlägt das System oder ein Mediator eine Menge von Lösungen (,Constraint') vor, die alle ausgehend von einem Referenzpunkt auf einer virtuellen Fläche (bei zwei Attributen auf einer Linie) im Raum der möglichen Verhandlungsergebnisse liegen.

Jeder der Verhandelnden wählt aus dieser Menge die Lösung aus, die ihm am meisten zusagt. Hierauf schlägt das System oder der Mediator eine weitere begrenzte Menge an Lösungen auf einer weiteren Fläche (Linie) vor, jedoch mit demselben Referenzpunkt. Die Verhandelnden suchen wiederum das von ihnen bevorzugte Ergebnis aus. Dieser Prozess wird solange fortgesetzt, bis die Parteien sich auf eine gemeinsame Lösung geeinigt haben. Die Wahl der ‚Constraints‘ sollte so erfolgen, dass sich die Lösungen der Parteien mit jedem Schritt weiter annähern. Unter den richtigen Bedingungen ist das Ergebnis dann pareto-optimal. Mit der Wahl verschiedener Referenzpunkte können unterschiedliche pareto-optimale Lösungen gefunden werden.

3.4.1.2.2 Auf Nutzenfunktionen basierende Methode

Bei der auf Nutzenfunktionen basierenden Methode werden zunächst die ‚Utility-Functions‘ aller Verhandelnden festgelegt. Um zu einer pareto-optimalen Lösung zu gelangen, wird eine gewichtete Summe aller ‚Utility-Functions‘ gebildet und diese maximiert. Hierbei entsteht ein multikriterielles Optimierungsproblem, das mit Hilfe verschiedener Optimierungsalgorithmen beispielsweise durch evolutionäre Algorithmen (Hanne, 1999; Kolonko, et al., 2002) gelöst werden kann. Durch unterschiedliche Gewichtungen können unterschiedliche pareto-optimale Lösungen erzeugt werden, aus denen die Parteien dann die beste Lösung auswählen.

3.4.1.3 SNT-Methoden

Interaktive SNT-Methoden (‚Single Negotiating Text‘) basieren auf einer gemeinsamen vorläufigen Einigung und einem gemeinsamen Single Negotiating-Text, der eine Ausgangsbasis für alle Parteien bilden soll. Diese Art der Vermittlung wird z.B. auch in politischen Verhandlungen eingesetzt, vor allem wenn keine persönlichen Verhandlungen möglich sind oder das Misstrauen zwischen den Parteien groß ist (Fletcher-Forum, 2002). Der SNT, also die vorläufige Einigung, wird entweder vom System, einem Mediator oder einem der Verhandelnden selbst vorgeschlagen. Daraufhin versuchen die Parteien den SNT dahingehend zu verbessern, dass er ihren eigenen Vorstellungen am nächsten kommt. Der Mediator kann hierbei unterstützend eingreifen. Basierend auf den Verbesserungsvorschlägen der Parteien wird ein neues vorläufiges Ergebnis erstellt, das wiederum von den verhandelnden Parteien kritisiert und verbessert werden kann. Dies geschieht so lange, bis eine endgültige Einigung erzielt wird. Diese Einigung ist dann pareto-optimal. Wiederum können verschiedene pareto-optimale Ergebnisse durch Wahl von verschiedenen Ausgangs-SNT erreicht werden. Diese Methode kann auch in

einer Postsettlement-Phase der Verhandlung eingesetzt werden, wobei der erste SNT dann eine schon erreichte, erneut zur Verhandlung gestellte Einigung darstellt. In diesem Fall wird der SNT auch als Post-Settlement bezeichnet (Raiffa, 1982).

Für die Wahl des jeweils nächsten SNT existieren verschiedene Methoden. Eine davon ist die ‚Method-Of-Improving-Directions‘, die von Ehtamo und Hämäläinen (Ehtamo, et al., 1999) entwickelt wurde. Bei dieser Methode werden die Verbesserungsvorschläge der einzelnen Parteien als Richtungen im Raum der möglichen Einigungen gesehen. In Richtung der ‚Improving-Direction‘ gelangt der Verhandelnde zu immer besseren Lösungsalternativen. Legt man eine zuvor erstellte Nutzenfunktion zugrunde, können diese Richtungen mathematisch gesehen als die Steigungen der Nutzenfunktion der entsprechenden Parteien betrachtet werden. Um die gemeinsam gewünschte Richtung (‚Jointly-Improving-Direction‘) zu finden, muss genau die Richtung gewählt werden, die als Resultierende aus allen Richtungen zusammengenommen hervorgeht. Bei zwei Verhandlungspartnern gewinnt man diese Richtung einfach dadurch, dass man den Winkel, der sich zwischen den beiden gewählten Richtungen der Verhandelnden aufspannt, halbiert. In dieser neuen Richtung wird nun das neue vorläufige Ergebnis gewählt. Der Prozess wird wie oben beschrieben fortgesetzt. Dies geschieht so lange, bis nur noch sehr kleine Veränderungen in dem nächsten SNT auftreten. Dies ist in der Praxis der Fall, wenn die Richtungen der beiden Parteien beinahe in einem 180° Winkel zu einander stehen.

3.4.1.4 Conjoint-Analyse

Die Conjoint-Analyse stellt zur Ermittlung der Nutzenfunktion der Verhandelnden ein wichtiges Verfahren dar. Die Conjoint-Analyse stammt ursprünglich aus der Marktforschung und wird bei Produktbefragungen eingesetzt, also zur Analyse von Präferenzen sowie zur Prognose von Kaufabsichten.

In den 60er und 70er Jahren wurden Entscheidungsprozesse bei Befragungen wissenschaftlich untersucht. Man fand heraus, dass die Ergebnisse einer Befragung nicht unbedingt die wahren Ansichten der Befragten widerspiegeln. So gaben die Befragten Antworten, von denen sie dachten, dass der Interviewer sie hören wollte. Oder aber sie antworteten, wie es ihnen zuerst in den Sinn kam. Um diese Schwierigkeiten zu vermeiden, wurde ein spezielles Verfahren entwickelt, die Conjoint-Analyse (siehe (dobney.com, 2003)).

Für die Ermittlung der Präferenzen eines Käufers bzw. eines Verhandelnden existieren zwei komplementäre Vorgehensweisen:

- der kompositionelle Ansatz und

- der dekompositionelle Ansatz.

Bei einem kompositionellen Vorgehen wird der Befragte direkt nach Bewertungen einzelner Attribute eines Objektes gefragt. Die Bewertungen werden am Ende zu einer Gesamtbewertung zusammengefügt.

Die Conjoint-Analyse ist dagegen dekompositionell: Es werden Gesamtbeurteilungen von ausgewählten Objekten erfragt, die anschließend in Einzelurteile bezüglich der Attribute und Optionen dieser Objekte zerlegt, ‚dekomponiert‘, werden.

Die verschiedenen Ansätze sollen an einem Beispiel demonstriert werden (Adwiraah, 2003).

Beim *kompositionellen* Ansatz wird eine Befragung über ein Handy durchgeführt, mit den Attributen Gewicht, Lebensdauer des Akkus und Farbe. Es gibt jeweils zwei alternative Optionen bei den Attributen:

Gewicht:	200g	120g
Lebensdauer des Akkus:	21 Stunden	10 Stunden
Farbe:	rot	blau

Der Befragte soll nun die Attribute einzeln bewerten, also beispielsweise: „Das Gewicht des Handy ist mir sehr wichtig, aber die Farbe nicht besonders.“ und diese Bewertungen auf einer Skala von 0 bis 100 angeben: Gewicht: 80 Punkte, Farbe: 10 Punkte. Ebenso werden die Optionen einzeln bewertet, z.B. 200g: 30 Punkte, 120g: 50 Punkte. Aus diesen Einzelbewertungen werden dann die Bewertungen für ein ganz bestimmtes Produkt komponiert. Die einzelnen Werte für die Optionen eines Attributes werden mit der Bewertung des Attributes multipliziert und anschließend addiert. Dies ergibt dann die Gesamtbewertung für ein ausgewähltes Produkt.

Beim *dekompositioneller* Ansatz bekommt der Befragte schon von vornherein aus einzelnen Attributen und Optionen zusammengesetzte Produkte vorgegeben, die er dann insgesamt bewerten soll:

	Handy A	Handy B
Gewicht	200g	120g
Lebensdauer des Akkus	21 Stunden	10 Stunden
Farbe	rot	blau
Bewertung	39	85

Durch eine Reihe von solchen Befragungen können anschließend die Bewertungen der einzelnen Attribute und Optionen zerlegt werden und so eine Nutzenfunktion ermittelt werden.

Im Rahmen der Untersuchungen und Testimplementationen wurde von Adwiraah (Adwiraah, 2003) in Anlehnung an INSPIRE eine Kombination von kompositionellem und dekompositionellem Ansatz verwendet

und im Rahmen eines Prototypen implementiert. Für den dekompositionellen Teil wurde, basierend auf den zuvor kompositionell ermittelten Werten, ein orthogonales Design entwickelt (BYU-Marketing-Department, 2002).

3.5 Elektronisch unterstützte Verhandlungen

Derzeit erfolgt die Unterstützung von Verhandlungen im Electronic Business durch entsprechende Systeme nur ansatzweise. Man spricht hier von so genannten Verhandlungsunterstützungssystemen („Negotiation-Support-Systems“, NSS). Dieses Forschungsfeld befindet sich immer noch im Aufbau, daher sind die folgenden Betrachtungen zum Teil nur konzeptioneller und theoretischer Natur. Es werden jedoch auch konkrete existierende Ansätze vorgestellt und diskutiert.

Die Charakteristik der elektronischen Unterstützung resultiert wie insgesamt im Electronic Business im Wesentlichen aus dem neuen elektronischen Medium Internet. Auf Daten können in Bezug auf Verhandlungen besser zugegriffen werden; die Daten können schneller ausgetauscht und besser abgelegt werden. Die Vorgänge können zum Teil automatisiert durchgeführt werden.

Ein wichtiger Aspekt der elektronischen Unterstützung ist die *Komplexität von Verhandlungsformen* als Anforderung an Verhandlungssysteme.

Als ein weiterer Aspekt ist die *Art der Unterstützungsfunktionalität* als Charakteristik der Verhandlungssysteme, beispielsweise „Process-“, oder „Solution-Driven“, von besonderer Bedeutung und soll ebenfalls im Folgenden näher betrachtet werden.

3.5.1 Komplexität der Verhandlungsform

Von entscheidender Bedeutung für die Entwicklung eines Verhandlungssystems sind die Komplexität der durch das System zu unterstützenden Verhandlungen und die daraus resultierenden Anforderungen an das Softwaresystem.

Zum einen gibt es Systeme mit dem Ziel, komplexe Verhandlungen durch elektronische Hilfen bei der Handhabung der Komplexität zu unterstützen. Dazu wird unter anderem Dokumentenmanagement eingesetzt.

Zum anderen wird seit einigen Jahren mehr oder weniger erfolgreich versucht, einfache Verhandlungsformen auch direkt im breiten elektronischen Markt einzubringen. Beispiele sind hier Auktionen auf B2B-

Handelsplätzen oder zwischen Endverbrauchern (eBay), sowie die in Kapitel 2 erwähnten Kaufagenten (Shopping Agents).

3.5.1.1 Einfache Verhandlungsformen

Einfache Verhandlungsformen lassen sich dadurch charakterisieren, dass für sie nur wenige der im Basismodell genannten Elemente relevant beziehungsweise nur in einfacher Form ausgeprägt und die resultierenden Verhandlungen relativ einfach technisch beherrschbar sind. Beispiele sind bilaterale Verhandlungen über den Preis einer Ware, eine Ausschreibungsrunde oder auch eine englische Auktion über den Preis einer Ware, die den Ablauf mittels eines festen Protokolls regelt.

Systeme, die einfache Verhandlungsformen unterstützen, werden teilweise schon im Massenmarkt eingesetzt. Beispielhaft seien die Auktionen auf B2B-Handelsplätzen oder zwischen Endverbrauchern (eBay), sowie die in Kapitel 2 erwähnten Kaufagenten („Shopping-Agents“) in einfacher Form genannt.

Jedoch schon bei den „Shopping-Agents“ ist die Akzeptanz unter anderem gering, da sie bisher keine komplexeren Verhandlungen ermöglichen. Komplexere Verhandlungen können sowohl von den Nachfragern her sinnvoll sein, um die passenden Konditionen für die eigenen Präferenzen zu erreichen, als auch für die Anbieter, um die jeweiligen Stärken ihres Angebots und damit auch den angemessenen Preis gegenüber dem Nachfrager transportieren zu können.

3.5.1.1.1 Kommerzielle Anbieter

Inzwischen gibt es auch kommerzielle Anbieter, die Systeme zur Unterstützung von Verhandlungen zur Verfügung stellen. Einige dieser Anbieter kommen aus dem Bereich des „E-Contracting“, bei dem weniger die eigentliche Unterstützung einer Verhandlung, sondern das Management einer Vertragserstellung im Vordergrund steht.

Oft lässt sich aus den zugänglichen Quellen der Anbieter der Funktionsumfang nicht eindeutig erschließen. Normalerweise werden jedoch nur einfache Verhandlungsformen unterstützt.

Ein Überblick zu kommerziellen Anbietern findet sich auch bei der E-Negotiations-Community (E-Negotiations-Community, 2002).

3.5.1.2 Komplexe Verhandlungsformen

Bei komplexen Verhandlungsformen sind mehrere der im Basismodell genannten Elemente relevant und flexibel ausgeprägt. Die so resultierenden Verhandlungen sind häufig sowohl von Menschen als auch von Seiten der Technik nur schwer beherrschbar.

Eine Möglichkeit der technischen Unterstützung kann die Komplexitätsreduktion durch elektronische Hilfsmittel sein.

Hier kann allgemeine Unterstützung durch Methoden des CSCW, des Dokumentenmanagements oder eine allgemeine Kommunikationsunterstützung geboten werden. Dies entspricht, wie durch Yen und Ng (Yen, et al., 2002) vorgeschlagen, einer Unterstützung von Verhandlungen durch reinen Wechsel des Mediums auf das Internet. Konkret bedeutet dies etwa das Führen einer freien Verhandlung in einem Chatraum. Nachteilig ist bei einer solchen Herangehensweise dagegen die dann weitgehend fehlende Anpassung auf die Verhandlungsdomäne. Ebenso findet keine Abstimmung auf die fachliche Anwendungsdomäne, in der verhandelt wird, statt.

Komplexitätsreduktion kann auch durch die schon oben im Kontext der Strategien angesprochenen Entscheidungsunterstützungssysteme oder durch spezielle Kommunikationsinfrastrukturen, die auf Verhandlungen und die Anwendungsdomäne angepasst sind, erreicht werden..

Die Erstellung einer solchen Kommunikationsinfrastruktur ist auch ein Ziel dieser Arbeit.

3.5.2 Art der Unterstützungsfunktionalität

Verhandlungen können auf verschiedene Art und Weise unterstützt werden. Yuan et al. (Yuan, et al., 1998) geben eine wichtige Klassifizierung von Verhandlungsunterstützungssystemen (,NSS') an: eine Einteilung in lösungsorientierte (,Solution-Driven') und prozessorientierte (,Process-Support') NSS. Eine ähnliche Einteilung nehmen auch Weigand, Schoop et al. (Weigand, et al., 2003) vor, wobei in ihrer Klassifizierung die *prescriptive Modelle* den *Solution-Driven-Modellen* von Yuan et al. und *descriptive Modelle* den *Process-Support-orientierten-Modellen* entsprechen.

Ein weiteres Kriterium aktueller Systeme ist die webbasierte Architektur von NSS, die daher auch als WNSS (Web-Based-Negotiation-Support-System) bezeichnet (Shim, et al., 1999) werden und eine leichte Bedienbarkeit fördern.

Lo und Kersten (Lo, et al., 1999) merken an, ein NSS bestehe aus zwei Komponenten: einer *Decision-Support*-Komponente und einer *Communication-Support*-Komponente. Die *Decision-Support*-Komponente unterstützt die Parteien dabei, die ihnen zur Verfügung stehenden Informationen besser zu verarbeiten und diese in der Verhandlung einzusetzen. Die *Communication-Support*-Komponente vereinfacht den Austausch von Angeboten und Argumenten und beschleunigt so den Einigungsprozess

und unterstützt auf diese Weise den Weg zu einem zufrieden stellenden Verhandlungsergebnis.

Die beiden Komponenten sind nicht einzuordnen in die lösungs- und prozessorientierten Funktionalitäten, da beispielsweise auch prozessorientierte Funktionalitäten eine Decision-Support-Komponente nutzen, um Strategieentscheidungen zu fällen. Ebenso benötigen lösungsorientierte Systeme einen ‚Communication-Support‘ zur bestmöglichen Übertragung und Diskussion ihrer Lösungsvorschläge.

Unter Einfluss linguistischer Erkenntnisse eignen sich weiterhin Methoden aus der Sprechakttheorie und der Konversationsanalyse zur Unterstützung von Verhandlungskommunikation. Auf die Sprechakttheorie wird ausführlich in Kapitel 4 eingegangen.

Ebenfalls werden Automatisierungsaspekte als eine Möglichkeit der Unterstützung von Verhandlungen betrachtet.

Zusammenfassend lassen sich also die folgenden wichtigen Kriterien und möglichen Elemente für NSS nennen:

- lösungsorientierte Systeme
- prozessorientierte Systeme
- webbasierte Architektur
- Decision-Support-Komponente
- Communication-Support-Komponente
- sprechaktbasierte Unterstützung
- Unterstützung durch Methoden der Konversationsanalyse
- Automatisierungsaspekte

3.5.2.1 Lösungsorientierte Verhandlungsunterstützungssysteme

Lösungsorientierte NSS generieren Verhandlungsergebnisse oder bieten den Verhandelnden Lösungsalternativen an. Der Fokus liegt auf einer Einigung im Sinne von pareto-Optimierung. Hierfür werden verschiedene Techniken der schon im Abschnitt 3.4.1 angesprochenen ‘Negotiation Analysis’ eingesetzt.

Aus der Sicht der lösungsorientierten NSS besteht die Hauptaufgabe des Systems darin, den Benutzer bei der Zusammenstellung seiner Angebote und der Einschätzung der erhaltenen Gegenangebote zu unterstützen, und dadurch möglichst ein pareto-optimales Ergebnis zu erzielen.

Für Solution-Driven-NSS ist die Decision-Support-Komponente von besonderer Bedeutung. Sie ist für den Verhandlungsprozess unumgänglich, weil zum einen die Präferenzen des Verhandelnden ermittelt werden und zum anderen aus diesen dann die für den Verhandelnden am besten geeigneten Angebote generiert werden. So enthält die Intentionsphase innerhalb der Vereinbarungsphase einer Verhandlung bei Solution-Driven-NSS eine Phase, in welcher der Benutzer die Attribute des Ver-

handlungsgegenstandes seinen Präferenzen entsprechend bewerten und in das System eingeben muss. Hierfür existieren unterschiedliche Methoden, die meistens eine Nutzenfunktion zum Ergebnis haben. Einige dieser Methoden werden im Abschnitt 'Negotiation-Analysis' näher erläutert. Mit Hilfe dieser Funktion ist es nun möglich, pareto-optimale Verhandlungsergebnisse zu berechnen. Dadurch können den Benutzern während der Verhandlung Entscheidungshilfen geboten werden, um eines dieser pareto-optimalen Ergebnisse zu erreichen.

Ein Defizit lösungsorientierter Systeme ist die fehlende Unterstützung von komplexen Verhandlungsformen mit beispielsweise komplexen Verhandlungsgegenständen und multiplen Attributen.

Die Communication-Support-Komponenten in lösungsorientierten Systemen stellen oft ein weiteres Problem dar, da sie nur wenig ausgeprägt sind. Yuan et al. (Yuan, et al., 1998) äußern sich hierzu in Bezug auf das Negotiation-Support-System INSS von InterNeg folgendermaßen: „This System [INSS] can be categorised as a solution-driven NSS. ... Since it uses a rather simple communication channel, INSS does not provide full process support for negotiators to organise and negotiate complex issues.“

Ebenso sehen Weigand, Schoop et al. (Weigand, et al., 2003) präskriptive Modelle“, die den Solution-Driven-Modellen entsprechen, aus einer eher kritischen Perspektive und propagieren prozessorientierte Systeme. Als Kritikpunkt an Solution-Driven-Ansätzen heben sie hervor, dass diese vorwiegend auf der Erstellung einer Nutzenfunktion basieren und von einer axiomatischen Rationalität der Verhandelnden ausgehen. Es wird dabei davon ausgegangen, dass die Verhandelnden immer die Option wählen, die für ihre Interessen am nützlichsten sind. Diese Annahmen sind zwar theoretisch gesehen gültig, jedoch führen sie in der Realität zu schwerwiegenden Fehlern. Die Annahmen über die Interessen der Verhandelnden sind oft zu allgemein oder hypothetisch; sie sind zu vereinfacht und ziehen zu wenige Variablen in Betracht. Weigand, Schoop et al. werfen den Solution-Driven-Ansätzen vor, sie würden den Verhandlungsprozess an sich ignorieren. Sie würden sich zu sehr auf das Ergebnis einer Verhandlung konzentrieren und nicht darauf, wie man zu diesem Ergebnis gelangen kann (Weigand, Schoop et al., 2003).

3.5.2.2 Prozessorientierte Verhandlungsunterstützungssysteme

Prozessorientierte NSS konzentrieren sich auf den Verhandlungsprozess als solchen. Dabei ist die Aufgabe des Systems nicht, den Verhandelnden die bestmöglichen Verhandlungsergebnisse zu präsentieren, sondern sie auf dem Weg dorthin, von der Vorbereitung bis zur Einigung, bestmöglich zu unterstützen. Dazu existieren mannigfaltige Möglichkeiten, wie

die Unterstützung bei der Umsetzung einer Strategie oder in der Kommunikation.

Weigand, Schoop et al. (Weigand, et al., 2003) heben hervor, dass ein Verhandlungsprozess immer als eine Art von Kommunikation gesehen werden sollte. Daher ist in Process-Support-NSS die Communication-Support-Komponente von besonderer Wichtigkeit. Für ein NSS dieser Kategorie steht der Prozess der Verhandlung im Vordergrund. Weigand, Schoop et al. (Weigand, et al., 2003) sehen es als Ziel, die elektronische Verhandlung so zu unterstützen, dass sie einer persönlichen Verhandlung mit natürlich sprachlicher Kommunikation in der realen Welt nahe kommt.

Dies ist jedoch kritisch zu sehen, da so auch die besonderen Vorteile, die eine elektronische Unterstützung bieten kann, wieder vergeben werden. Hier ist insbesondere die durch die elektronische Unterstützung möglich werdende Strukturierung der ausgetauschten Informationen, beispielsweise durch explizite Sprechakte, zu nennen.

Beispiele für Process-Support-NSS sind WebNS von Yuan et al. (Yuan, et al., 1998) und Negoisst (Schoop, et al., 2002). Yuan et al. sehen als ein wichtiges Merkmal ihres NSS die Möglichkeit der Real-time-Kommunikation. Sie unterscheiden zwischen reichen und einfachen Kommunikationskanälen. Nach McGrath und Hollingshead (McGrath, et al., 1994) kann Kommunikation als eine Reihe von Kommunikationskreisen („Communication-Cycles“) gesehen werden. Jeder Kreis besteht aus einer Reihe von Operationen auf einer gegebenen Nachricht. Diese Operationen sind die Erstellung, die Übermittlung, der Empfang einer Nachricht, die Empfangsbestätigung und die Antwort auf eine Nachricht. Die Zeit um einen Kommunikationskreis zu vollenden, variiert für verschiedene Kommunikationskanäle und ist ein Maß dafür, inwiefern die Kommunikation über einen bestimmten Kanal reich oder arm an Informationen ist. Je länger die Kommunikationskreise sind, desto weniger Informationen können über den Kanal übertragen werden und desto einfacher ist der Kommunikationskanal. Beispiele für eher einfache Kommunikationskanäle sind z.B. Email-Kommunikation oder Briefe. Alle Operationen, die auf einer Nachricht ausgeführt werden können, brauchen relativ viel Zeit im Vergleich zu einer persönlichen Kommunikation. Außerdem entfallen alle visuellen Möglichkeiten einer solchen Kommunikation, wie z.B. Mimik, Gesten, Blicke usw. Diese zusätzlichen Möglichkeiten der Kommunikation können auch parallel zu der sprachlichen Kommunikation angewendet werden und unterscheiden sie von den meisten anderen Kommunikationsmedien. Dies und die Tatsache, dass persönliche Kommunikation über die kürzesten Kommunikationskreise verfügt, machen sie zu dem reichsten Kommunikationsmedium.

Webbasierte NSS sind derzeit auf schriftliche Kommunikation beschränkt. Um diesen relativ armen Kommunikationskanal zu verbessern und die Kommunikationskreise zu verkürzen, schlagen Yuan et al. Real-Time-Kommunikation und -Interaktion zwischen den Verhandlungspartnern vor. Schriftliche Kommunikation ohne Real-Time-Komponente bietet jedoch auch Vorteile, z.B. ist es möglich, alle Worte einer Nachricht noch einmal zu überdenken, bevor man sie übermittelt. Folglich ist die Kommunikation und Interaktion überlegter. Außerdem können schriftlich festgehaltene Nachrichten später leichter noch einmal eingesehen werden und bieten so eine bessere Dokumentation des Kommunikationsprozesses. Um diesen Aspekt der Kommunikation nicht zu vernachlässigen, sind folgende drei weitere Merkmale von Process-Support-NSS wichtig: der Dokumentenaustausch, das gemeinsame Arbeiten an Dokumenten und die Dokumentation des Verhandlungsprozesses in einer History-Funktion.

3.5.2.3 Konversationsanalyse

Im Rahmen der ganzheitlichen Betrachtung von Verhandlungen wurden in der Konversationsanalyse zwei geeignete Möglichkeiten der Unterstützung von Verhandlungskommunikation identifiziert: ‚*Adjacency-Pairs*‘ und ‚*Turn-Taking*‘.

Harvey Sacks (Auer, 1999), der als der Begründer der Konversationsanalyse gilt, nennt in seiner Untersuchung von Alltagskonversationen eine sequentielle Bindung von zwei Sprechakten eine *Paarsequenz* („Adjacency-Pair“). Paarsequenzen „sind dadurch charakterisiert, dass sie zwei Handlungstypen aneinander binden, die von unterschiedlichen Gesprächsteilnehmern ausgeführt werden, unmittelbar aufeinander folgen und durch eine Beziehung der konditionellen Relevanz miteinander verkettet sind“. In Alltagskonversationen ist ein solcher Ablauf normativ geregelt. Dadurch kann ein Gesprächspartner prüfen, ob der andere ihn verstanden hat. Folgt z.B. auf eine Nachfrage („Request“) ebenfalls eine Nachfrage, so ist anzunehmen, dass die erste vom Empfänger nicht richtig verstanden wurde, oder dass er diesen absichtlich ignoriert hat. Außer Paarsequenzen sind auch sequentielle Abläufe mit weniger starken Bindungen möglich. Dann muss zwar auf eine Äußerung eine zweite folgen, dies muss jedoch nicht unmittelbar geschehen und der Typ der Äußerung ist nicht unbedingt festgelegt (Auer, 1999; Dürr, et al., 1994).

Weitere Untersuchungen der Konversationsanalyse betreffen das ‚*Turn-Taking*‘ (Redewechsel). Im Allgemeinen sollten sich bei zwei Gesprächs-/Verhandlungspartnern die Sprecher abwechselnd äußern. Gibt es mehrere Parteien, so vergibt der augenblickliche Sprecher das Rederecht. Tut er dies nicht, so kommt derjenige zu Wort, der sich zuerst äußert („First-

Starter'). Nimmt keiner der anderen Gesprächsteilnehmer das Wort an sich, so fällt das Rederecht an den ersten Sprecher zurück.

Diese Regeln aus der Konversationsanalyse lassen sich nun zur Unterstützung von Verhandlungsprozessen verwenden. Denkbar wäre eine Plausibilitätskontrolle, die bewertet, wie sinnvoll eine Äußerung in Bezug auf eine vorhergehende Nachricht und ein gegebenes Verhandlungsprotokoll ist. Bei geringer Plausibilität könnte eine Fehlermeldung ausgegeben werden, die den Versender der Nachricht zu einer Korrektur auffordern. Eine weitere Unterstützung des Systems könnten Vorschläge in Form von Sprechakten für eine sinnvolle Fortsetzung der Verhandlung sein.

3.5.2.4 Spannungsfeld Automatisierung und Komplexität

Beim Thema der Automatisierung von Verhandlungen kommt bei Anwenden oft die intuitive Wunschvorstellung auf, ihre vielfältigen alltäglichen Verhandlungssituationen vollständig automatisiert und damit einerseits ohne jeglichen Aufwand und andererseits mit optimalem Ergebnis abzuwickeln.

Die Abbildung der Anwenderinteressen und die Komplexität der aus solchen allgemeinen Situationen resultierenden Verhandlungsformen sind mit dem in absehbarer Zeit zu erreichenden Stand der Forschung nicht umsetzbar (vgl. (Ströbel, 2003), S. 22).

Ein Beispiel dafür ist die oft schon schwierige Erstellung von Nutzenfunktionen als eine oft wichtige Grundlage für die Realisierung von Verhandlungsstrategien, wie oben diskutiert. Das *Strategie-Paradoxon* von Beam et. al. (Beam, et al., 1999) fasst diese Schwierigkeit ausführbarer, also formalisierbarer Strategien zusammen: „if a strategy is simple enough to be formalised, then it probably can be deciphered by the opposing agent – if it is too complex, it cannot be formalised.“

Dennoch lassen sich Teilbereiche von Verhandlungen automatisieren.

Sehr einfache Automatisierungen (Trivial-Automatisierung) sind zum Beispiel Bietagenten in einer Auktion, die bis zu einem vorgegeben Höchstpreis ohne Benutzerinteraktion bieten. Hier ist die Verhandlungsform mit nur dem zu verhandelnden Preis sehr einfach; das Protokoll der Verhandlung ist fest vorgegeben; die Strategie ist sehr einfach (bieten bis zum vorgegeben Höchstpreis) und die auszutauschenden Nachrichten denkbar simpel. So eine Trivial-Automatisierung ist beispielsweise bei eBay realisiert und trotz ihrer Einfachheit dennoch im Rahmen ihrer Möglichkeiten sinnvoll und nützlich.

In dieser Arbeit werden insbesondere weitere *Grundlagen* für die Automatisierung von Verhandlungen geschaffen.

Des Weiteren wird in Kapitel 5 eine Infrastruktur für Verhandlungen vorgestellt, die vollständig maschinell handhabbar ist.

3.5.3 Verhandlungsunterstützungssysteme

Ein Verhandlungsunterstützungssystem (NSS) ist im weitesten Sinne ein Softwaresystem zur Unterstützung von verschiedenen Verhandlungsaktivitäten (Kersten, et al., 2001).

Im Folgenden werden die Strukturen von NSS sowie konkrete Beispiele aus der Forschung vorgestellt.

3.5.3.1 Beispiele für NSS

Im Folgenden werden die aus der Forschung bekannten wichtigsten Beispiele und Typen von Verhandlungsunterstützungssystemen näher erläutert.

3.5.3.1.1 INSPIRE

INSPIRE ist ein webbasiertes NSS, das von der InterNeg Gruppe an der Carleton University in Kanada entwickelt wurde. Es unterstützt den Benutzer bei der Spezifikation seiner Präferenzen, der Erstellung von Angeboten sowie der Einschätzung von Gegenangeboten. Es verfügt außerdem über ein internes Messaging-System und eine grafische Anzeige des Verhandlungsprozesses. INSPIRE wurde vorrangig entwickelt, um interkulturelle Verhandlungen zu studieren. Das System kann jedoch auch zu anderen Zwecken eingesetzt werden, wie beispielsweise Verhandlungen im E-Commerce.

Das System bietet zwar Unterstützung für bilaterale Verhandlungen über multiple Attribute, jedoch keine Automatisierung und keine Mediationsmöglichkeiten. INSPIRE unterstützt den Benutzer in den folgenden Phasen einer Verhandlung:

‘Description of the negotiation case’: In dieser Phase werden die Benutzer zunächst einmal mit dem Verhandlungsfall vertraut gemacht. Da es sich um kein real eingesetztes System handelt, wird dem Benutzer ein virtuelles Handelsszenario vorgestellt. Hierbei sind die zu verhandelnden Attribute sowie deren mögliche Werte schon von vornherein festgelegt.

‘Prenegotiation phase’: Diese Phase bereitet die Verhandlung vor und besteht ihrerseits aus den Phasen: ‘Problem-Identification’, ‘Preference-Elicitation’ und ‘Evaluation of alternative packages’.

In der Problem-Identification-Phase werden die zu verhandelnden Attribute identifiziert und mögliche Werte (‘Options’) ermittelt.

Hierauf folgt die umfangreiche Preference-Elicitation-Phase, in der der Benutzer seine Präferenzen ermittelt und festlegt. Im ersten Schritt werden die zu verhandelnden Attribute gewichtet. Der Benutzer vergibt allen Attributen, ihrer Wichtigkeit nach, Punkte auf einer Skala von 0 bis

100. Im zweiten Schritt verfährt er ebenso mit den Werten für die einzelnen Attribute. INSPIRE wertet diese Angaben aus und präsentiert sie dem Benutzer im dritten Schritt zusammengefasst als Pakete von Attributen mit verschiedenen Werteangaben. Die Gewichtung der Pakete geschieht dabei automatisch, kann jedoch in der *Evaluation-Of-Packages*-Phase vom Benutzer wenn nötig geändert werden. Die Generierung der Pakete und ihrer Wichtigkeit sowie deren Evaluierung geschieht mit Hilfe der Conjoint-Analysis. Das Ergebnis dieser Auswertung mündet in einer Nutzenfunktion.

Die so gewonnenen Informationen werden von INSPIRE im nächsten Schritt genutzt, um dem Benutzer Hilfestellungen und Feedback bei der Erstellung von Angeboten zu geben und bei der Auswertung der Angebote des Verhandlungspartners.

„Conduct of the negotiation“: Diese Phase besteht aus dem Austauschen von Nachrichten und Angeboten, solange bis eine Einigung erzielt wird, einer der beiden Verhandlungspartner die Verhandlung abbricht oder bis der von INSPIRE festgelegte Endzeitpunkt der Verhandlung überschritten wird. INSPIRE bietet hierfür Menüs an, um die in der Pre-negotiation-Phase erstellten Angebotspakete zu verschicken und zeigt dann jeweils ein „Rating“, welches aus der Nutzenfunktionen gewonnen wird, für das aktuelle Angebot an. Außerdem ist es möglich, über eine Eingabefeld einfache Textnachrichten an den Verhandlungspartner zu schicken. Die Benutzer werden jeweils per E-mail benachrichtigt, wenn ein neues Angebot oder eine Nachricht des Verhandlungspartners eingeht.

Während der Verhandlung können die Benutzer jederzeit die Bewertungen der einzelnen Pakete und damit ihre Nutzenfunktion verändern.

Weiterhin ist es für beide Benutzer jederzeit möglich, einen Graphen einzusehen, der Dynamik und Verlauf der Verhandlung darstellt. Es werden die „Ratings“ der Angebote und Gegenangebote beider Verhandlungspartner dargestellt, jedoch jeweils nur aus der Verhandlungsperspektive des Betrachters. Es ist also nicht möglich, das „Rating“ der anderen Partei für ein bestimmtes Angebot zu sehen. Dies bewahrt die Benutzer davor Informationen preiszugeben, die dem anderen einen Vorteil verschaffen könnten. Es kann aber auch dazu führen, dass eine Veränderung im Graphen, für die eine Partei wie unnachgiebiges Verhalten des Verhandlungspartners aussieht, in Wirklichkeit aber und aus der Perspektive des anderen gesehen, ein Zugeständnis war. Solche Missverständnisse können sich dann negativ auf den weiteren Verhandlungsverlauf auswirken.

„Post-Settlement-Phase“: In dieser Phase besteht die Möglichkeit, die Verhandlung nach einer Einigung erneut aufzunehmen um eine andere, eventuell bessere Einigung zu erreichen. Sobald eine Einigung erzielt wurde, untersucht das System sie mit Hilfe der Nutzenfunktion der Be-

nutzer auf ihre Effizienz im Sinne von pareto-Optimalität. Wenn das Ergebnis nicht pareto-optimal ist, berechnet INSPIRE pareto-optimale Lösungen und bietet sie den beiden Verhandlungspartnern als Alternativen an. Von diesem Punkt aus können die Benutzer dann erneut in die Conduct-Phase treten, um die gegebenen Alternativen zu verhandeln. Anders als zuvor ist es ihnen jedoch nun nicht mehr möglich, ihre Utilities zu verändern.

3.5.3.1.2 'Joint-Gains'

'Joint-Gains' ist eine Entwicklung des Systems-Analysis-Laboratory an der Universität Helsinki, unter Zusammenarbeit von Eero Kettunen, Raimo P. Hämäläinen und H. Ehtamo. 'Joint-Gains' ist ein Solution-Driven-NSS, da es sich hauptsächlich auf die Optimierung des Verhandlungsergebnisses konzentriert. Es bietet Unterstützung für multilaterale Verhandlungen über multiple Attribute. Im Gegensatz zu INSPIRE ist der Verhandlungsprozess weitgehend automatisiert.

Für jede Verhandlung ist ein Vermittler (Mediator) notwendig, der das Verhandlungsszenario erstellt und den verhandelnden Parteien den Zugang zu dem System ermöglicht. In der Preparation-Phase legt der Mediator die Anzahl der Verhandlungsteilnehmer, die zu verhandelnden Attribute, sowie die niedrigste und oberste Grenze für deren Werte fest. Außerdem ist es möglich, so genannte „Criteria“ hinzuzufügen. Dies sind Verhandlungsattribute, die Abhängigkeiten zwischen einzelnen Attributen darstellen. Ein wöchentliches Gehalt beispielsweise ergibt sich aus dem Stundensatz und den Arbeitsstunden pro Woche. Schließlich werden alle Attribute mit einem Startwert initialisiert. Dieser wird von 'Joint-Gains' vorgegeben und kann vom Mediator trotzdem beliebig verändert werden. Diese Vorbereitungen werden in 'Joint-Gains' als Problemdefinition bezeichnet. Abschließend werden die angegebenen Werte den vorher eingetragenen Verhandlungsteilnehmern veröffentlicht. Danach kann die Verhandlung gestartet werden. Es können jedoch auch weiterhin jederzeit Änderungen an der Problemdefinition vorgenommen werden. Hierfür muss der Mediator die Verhandlungssitzung durch ein 'Unpublish' unterbrechen, die Werte ändern und dann die Verhandlung durch erneutes Veröffentlichen der Problemdefinition fortsetzen.

Im Prozess der Durchführung der Verhandlung ermitteln die Verhandlungsteilnehmer interaktiv mit dem System ihre Präferenzen und teilen sie dem System mit. 'Joint-Gains' benutzt zur Ermittlung von pareto-optimalen Verhandlungsergebnissen die 'Method-of-Improving-Directions', die auf der SNT-Methode basiert. Der Mediator wählt den 'Ausgangs-SNT'. Dem Benutzer werden in mehreren Schritten mehrere Vergleichsangebote zur Auswahl gestellt, aus denen dieser sein Meistgewünschtes auswählt. Kann sich der Benutzer nicht mehr klar für eine der

beiden ihm angebotenen Lösungen entscheiden, weil diese einander zu sehr ähneln, teilt er dieses dem System durch das Drücken des Buttons ‚Indifferent‘ mit. Aus den ermittelten Werten berechnet das System dann die ‚Improving-Directions‘. Dieses Vorgehen wird rundenweise für alle Verhandelnden durchlaufen. Erreicht der Prozess einen Punkt an dem die ‚Improving-Directions‘ der Parteien diametral auseinander laufen, steht das Ergebnis fest und die Verhandlung ist beendet. Um unterschiedliche Ergebnisse auf der Pareto-Grenze zu ermitteln, können mehrere Sessions mit unterschiedlichen Startwerten für die Attribute durchlaufen werden. Die einzelnen Sessions werden vom Mediator eröffnet und verwaltet.

3.5.3.1.3 CBSS / WebNS

CBSS (Collective-Bargaining-Support-System) ist ein webbasiertes Negotiation-Support-System, das auf der Übermittlung von unstrukturierten Textnachrichten basiert. Es wurde von Yuan, Rose, Archer und Suarga an der McMaster University in Kanada und der Hasanuddin University in Indonesien entwickelt. CBSS bietet Unterstützung von bilateralen Verhandlungen über mehrere Attribute. Eine Weiterentwicklung von CBSS ist WebNS. Dort ist für jede bilaterale Verhandlung außerdem ein Mediator vorgesehen. Sowohl CBSS als auch WebNS bieten keine Automatisierung.

CBSS ist im Gegensatz zu INSPIRE und ‚Joint-Gains‘ ein prozessorientiertes NSS, da es insbesondere den Verhandlungsprozess unterstützt. Es werden jedoch keine Vorschläge oder Lösungen im Sinne der Optimierung des Verhandlungsergebnisses generiert. Das System ist für viele Arten von Verhandlungen einsetzbar, einschließlich B2B-Verhandlungen.

CBSS unterstützt die Preparation-Phase und die Conduct-Phase einer Verhandlung durch verschiedene Systemeinheiten. So gibt es einen Pre-Session-Teil, mit der Möglichkeit, die zu verhandelnden Attribute festzulegen und Notizen für die Verhandlung vorzubereiten, und einen Session-Teil, der den eigentlichen Verhandlungsprozess unterstützt. Außerdem ist immer ein Hot-Line-Window geöffnet, in dem man in Echtzeit kurze Nachrichten mit seinem Verhandlungspartner austauschen kann.

Der Session-Teil besteht aus den unten beschriebenen Phasen. Für jede Phase gibt es ein Session-Window, welches wiederum aus drei Subfenstern besteht: ‚Our-Window‘, ‚Their-Window‘ und ‚Common-Window‘. In ‚Our-Window‘ werden die eigenen Angebote und Antworten ausgegeben, in ‚Their-Window‘ die des Verhandlungspartners. Einigungen werden im ‚Common-Window‘ dargestellt.

Die Phasen des Session-Teils sind:

‚General-Discussion‘:

Anders als INSPIRE und 'Joint-Gains' bietet CBSS die Möglichkeit, die Verhandlungsgegenstände in einer Art Vorverhandlung zu diskutieren und anschließend gemeinsam festzulegen. Nachfolgend werden, wie in der 'Perference-Elicitation' Phase bei INSPIRE, die Präferenzen der einzelnen Verhandlungspartner ermittelt. Hierbei können ein Zeitlimit für die Dauer der Verhandlung, 'Trade-Offs' und eine obere und untere Grenze für jedes Verhandlungsattribut festgelegt werden. Diese Phase ist somit noch zu der Preparation-Phase der Verhandlung zu zählen.

„Issue-Discussion“:

In dieser Phase tauschen die Parteien auf der Basis einfacher unstrukturierter Textnachrichten Angebote und Gegenangebote aus und suchen für jedes einzelne Verhandlungsattribut nach einer Einigung. Wird eine Einigung über ein Issue erzielt, so wird dies im Common-Window dargestellt. Während der Verhandlung werden alle ausgetauschten Nachrichten automatisch in verschiedenen Files auf dem CBSS-Server gespeichert. Die Nutzer können über ihren Nutzernamen alle ihre Verhandlung betreffenden Daten einsehen. Diese werden dann in den dazugehörigen Fenstern dargestellt, können jedoch nicht mehr editiert werden.

„Completing-The-Agreement“:

In der letzten Phase finden beide Parteien eine endgültige Einigung, die aus allen vorangegangenen Einigungen über die einzelnen Attribute besteht. Diese Einigung wird in einem speziellen Final Agreement Window dargestellt und kann von beiden Verhandlungsteilnehmern ebenfalls auf Textbasis und mit Hilfe von 'Cut-And-Paste' noch solange verändert werden, bis beide das Agreement durch das Drücken des 'Sign-Buttons' unterzeichnet haben.

3.5.3.1.4 Negoisst

Negoisst ist ein prozessorientiertes NSS, das an der Universität RWTH in Aachen, im Rahmen der „Research-Group Electronic-Business - Electronic-Markets“, von Schoop et al. entwickelt wurde. Es unterstützt den Verhandlungsprozess durch den Austausch von sprechaktbasierten, semi-strukturierten Nachrichten. Es bietet hauptsächlich Support für bilaterale Verhandlungen, wobei aber theoretisch auch multilaterale Verhandlungen möglich wären. Diese werden jedoch nicht, wie z.B. bei 'Joint-Gains', durch einen Mediator gesondert unterstützt. Es können beliebig viele Attribute zur Verhandlung gestellt werden. Das System bietet keine Automatisierung, wurde jedoch speziell für B2B-Verhandlungen entwickelt.

Negoisst besteht aus zwei Hauptkomponenten. Es unterstützt erstens den Austausch von Nachrichten, die auf 'Communicative-Acts' basieren, und erstellt zweitens einen Kontrakt, ein Dokument, in dem die Ergebnis-

se der Verhandlung festgehalten werden. Daher bietet es eine Kombination aus semistrukturiertem Nachrichtenaustausch und kooperativem Dokumentenmanagement. Die Nachrichten werden basierend auf Sprechakten in verschiedene Nachrichtentypen unterteilt. Der Typ einer Nachricht wird während der Verhandlung vom Nutzer festgelegt. Nachrichten können sowohl für den formalen wie auch für den informalen Nachrichtenaustausch benutzt werden. Der Austausch folgt einem Verhandlungsprotokoll, das auf der Kategorisierung und Semistrukturierung der Nachrichten nach Sprechakten basiert. Eine Nachricht, die versandt wird, kann je nach Typ dem Sender Verpflichtungen auferlegen, die dieser nach dem Versenden zu erfüllen hat. Ebenso kann vom Empfänger eine Antwort eines bestimmten Typs erwartet werden. Die Verhandelnden können sich eine Liste mit ihren eigenen Verpflichtungen und denen ihrer Verhandlungspartner anzeigen lassen. Daraus ist ersichtlich, welche weiteren Maßnahmen getroffen werden sollen.

Die Struktur der Nachrichten orientiert sich an natürlicher Sprache, bietet jedoch in sofern eine Semistrukturierung, als dass einzelnen Elementen des Inhalts ein Typ zugeordnet werden kann. Die Liste der Typen ist vom Programm vorgegeben und kann vom Nutzer beliebig erweitert werden. Weiterhin wird der Verhandlungsprozess durch die sprechaktbasierten Nachrichtentypen strukturiert. Das Verhandlungsprotokoll steuert den Verhandlungsprozess, indem es den Verhandelnden je nach Situation vorschreibt, wie die Verhandlung fortzusetzen ist, d.h. von welchem Typ die folgende Nachricht sein darf. Z.B. kann eine Verhandlung mit einer Nachricht vom Typ ‚Offer‘ oder ‚Request‘ begonnen werden. Ein ‚Request‘ kann wiederum nur mit einer ‚Counter-Offer‘, einem ‚Accept‘ oder einem ‚Reject‘ beantwortet werden. Die Verhandlung kann nur mit einem ‚Accept‘ oder einem ‚Reject‘ enden. Außerdem müssen die Nachrichten wechselseitig ausgetauscht werden, d.h. man kann seine eigenen Nachrichten niemals selbst beantworten.

Der Verhandlungsprozess wird kontinuierlich in einem Dokument, dem Vertrag, festgehalten. Außerdem werden alle Vorgänge, d.h. das Versenden von Nachrichten und Änderungen am Vertrag, dokumentiert, so dass der gesamte Verhandlungsprozess jederzeit nachvollziehbar ist. Der Vertrag wird automatisch aus den ausgetauschten Nachrichten erzeugt und gibt daher den aktuellen Verhandlungsstand wieder. Die Beziehung zwischen Vertrag und ausgetauschten Nachrichten kann jederzeit sichtbar gemacht werden, indem auf ein Item des Vertrages geklickt wird. Hierauf wird diejenige Nachricht angezeigt, die dieses Item nach dem aktuellen Stand des Vertrages erzeugt hat. Ebenfalls kann der gesamte Prozess von ausgetauschten Nachrichten, der zu diesem Vertragsitem gehört, sichtbar gemacht und zurückverfolgt werden.

Negoisst bietet teilweise auch eine Unterstützung für die Abwicklungsphase einer Markttransaktion. Dadurch dass der Kontrakt und die daraus resultierenden Verpflichtungen jederzeit sichtbar gemacht werden können, kann der Nutzer jederzeit überprüfen, ob die aktuellen, vertraglich festgesetzten Leistungen auch tatsächlich zu erbringen sind. Hiermit sollen nicht erfüllbare Verträge im Vorfeld vermieden werden. Weiterhin wird der Vertrag von Negoisst auf Konsistenz geprüft. Beispielsweise wird der Nutzer darauf hingewiesen, wenn für eine Zahlung ein bestimmter Termin angegeben wurde, der festgesetzte Zeitpunkt jedoch bereits in der Vergangenheit liegt.

3.5.3.2 Bewertung

Grundsätzlich sollte zwischen den Systemen mit Schwerpunkt auf Prozessorientierung und auf Lösungsorientierung unterschieden werden.

INSPIRE ist ein Solution-Driven-NSS, da es nur geringen ‚Process-Support‘ bietet und dem Benutzer bei der Erstellung seiner Nutzenfunktion behilflich ist. Jedoch bietet die Solution-Driven-Komponente während der Conduct-Phase nur wenig Unterstützung. Pareto-optimale Lösungen werden nur in der Post-Settlement-Phase generiert und den Verhandlungspartnern als BATNA (‚Best-Alternative-To-A-Negotiated-Agreement‘) angeboten. Der Hauptaspekt der Unterstützung liegt in der Pre-Negotiation-Phase, in den dort eingesetzten decision-analytischen Methoden. Die Conduct-Phase bietet dem Benutzer dagegen lediglich eine Orientierungsfunktion an, um Angebote besser einschätzen zu können. Die Entscheidung welche Angebote er verschickt und welches Angebot er annimmt, bleibt jedoch letztendlich dem Benutzer überlassen.

Negative Aspekte des Systems sind weiterhin die begrenzten Visualisierungsmöglichkeiten des Verhandlungsprozesses und der relativ einfache Kommunikationskanal.

Bei ‚*Joint-Gains*‘ werden die Präferenzen der Verhandelnden durch einen interaktiven Prozess während der Verhandlung ermittelt. Durch dieses Vorgehen ist es für den Benutzer nicht notwendig, seine Nutzenfunktion explizit anzugeben. Dies kann von Vorteil sein, wenn dem Benutzer zu Beginn der Verhandlung seine Präferenzen zum Teil noch unklar sind. Hier wird besonders der Entscheidungsprozess, der einer Verhandlung innewohnt, deutlich.

Obwohl es bei ‚*Joint-Gains*‘ möglich ist, multilaterale Verhandlungen zu führen, werden keine One-To-Many-Verhandlungsfälle unterstützt, die jedoch für den B2B-Anwendungsfall wünschenswert wären. Bei Verhandlungen mit mehreren Teilnehmern ist ‚*Joint-Gains*‘ eher als Group-Decision-System verwendbar, mit dessen Hilfe nach einer gemeinsamen optimalen Lösung für ein Problem gesucht wird.

Weiterhin ist problematisch, dass die Werte der Attribute als numerische, kontinuierliche Werte erfassbar sein müssen. Somit ist es nicht möglich, wie das bei INSPIRE der Fall war, Attribute mit nur zwei oder drei Alternativen anzugeben, beispielsweise ob eine Lieferung sofort oder später erfolgt, oder ob ein Produkt blau, rot oder gelb sein soll. Die History-Funktion von 'Joint-Gains' beschränkt sich darauf, die Initialwerte, die jeweiligen Zwischenwerte (SNT's) und die Endwerte der Attribute in einem Balkendiagramm darzustellen.

Der Kommunikationskanal ist wie bei INSPIRE eingeschränkt, obwohl eine Echtzeitkommunikation über kurze Textnachrichten möglich ist. Es fehlt jedoch eine Strukturierung der Nachrichten und eine Verbindung zum eigentlichen Verhandlungsprozess.

Bei CBSS liegt der Fokus auf der Echtzeitkommunikation und Interaktion, die das System bietet. Das System dokumentiert alle Verhandlungsschritte und speichert sie als Dateien auf dem Server. Somit ist es für die Verhandlungsteilnehmer jederzeit möglich, sich den Verhandlungsprozess im nach hinein anzuschauen und nachzuvollziehen. Hierfür stehen jedoch keine speziellen Tools zur Verfügung, die z.B. wie bei INSPIRE den Verhandlungsprozess über eine grafische Oberfläche erfassbar machen und so eine einfache Analyse ermöglichen.

Ein großer Unterschied zwischen den bisherigen Solution-Driven-NSS und dem Process-Support-NSS CBSS ist die Tatsache, dass bei den vorherigen Systemen alle Attribute einer Ware auf einmal verhandelt werden konnten. Bei CBSS werden dagegen alle Attribute einzeln und sequentiell verhandelt. Dies ist charakteristisch für Process-Support-Systeme. Die einzelnen Ergebnisse der Attributverhandlungen werden in einem Dokument festgehalten, welches gleichzeitig den Verhandlungsprozess strukturiert und dokumentiert, und in der Enderfassung einen Vertrag bildet.

Negoisst ist wie CBSS ein Process-Support-NSS, verfügt jedoch über eine bessere Unterstützung des Verhandlungsprozesses. Durch die Semi-strukturierung der Nachrichten und deren Kategorisierung nach Sprechaktkriterien wird die Kommunikation strukturierter und dadurch übersichtlicher. Die Bereitstellung eines Kommunikationsprotokolls, basierend auf den Sprechaktregeln, bildet eine wertvolle Unterstützung des Verhandlungsprozesses, die bei CBSS schlichtweg nicht vorhanden ist. Ebenso wie bei CBSS steht jedoch sowohl am Ende als auch während der Verhandlung der Kontrakt als zentrales Dokument. Zur Strukturierung der Verhandlung wird ein Rahmenvertrag zugrunde gelegt, welcher in jedem Verhandlungsschritt weiter vervollständigt wird. Durch die Aufzeichnung der einzelnen Schritte ist eine History-Funktion gegeben, die es möglich macht, den Verhandlungsprozess jederzeit nachzuvollziehen. Dies ist ebenfalls eine wichtige Eigenschaft von Process-Support-

Systeme, da die History-Funktion eine besondere Hilfe für die Unterstützung des Verhandlungsprozesses darstellt. Sie kann jedoch nicht als charakteristisch für Process-Support-Systeme erachtet werden, da Solution-Driven-Systeme ebenfalls von dieser Form der Unterstützung Gebrauch machen.

3.5.4 Funktionen zur Verhandlungsunterstützung

Zusammenfassend werden nun die Einzelaufgaben, die während eines Verhandlungsprozesses auftreten und die von einem Negotiation Support System unterstützt werden können, aufgeführt. Diese Auswertung beruht auf den oben erarbeiteten Erkenntnissen und durchgeführten Recherchen und stellt die Grundlage für den Entwurf weitergehender Systeme dar.

Funktionen von *lösungsorientierten Systemen* („Solution-Driven“):

- Festlegen der Präferenzen des Benutzers, in Bezug auf
 - Attribute
 - Werte der Attribute („Options“)
- Dies ist eine Hilfe beim Erstellen der Nutzenfunktion.
- Angebotspakete erstellen und verschicken
- Bewerten der Angebotspakete nach Nützlichkeit für den Benutzer (im Sinne von Pareto-Optimierung)
 - Eigene Angebote
 - Eingehende Gegenangebote des Verhandlungspartners
- Pareto-optimale Lösungen generieren
- Post-Negotiation
 - Hinweis, falls das Ergebnis nicht pareto-optimal ist, dies kann zu einer erneuten Verhandlung führen

Funktionen von *prozessorientierten Systemen* („Process-Support“):

- Unterstützung der Kommunikation durch Versenden von Nachrichten
 - unstrukturiert
 - semistrukturiert
 - beispielsweise nach Sprechaktkriterien
- Typisierung der Nachrichten durch Sprechakte
- Bereitstellung eines Verhandlungsprotokolls
- Kontraktbildung (Vertrag- / Einigung-)
 - Strukturierung des Verhandlungsprozesses durch den Kontrakt
 - Erstellung eines Rahmenvertrages
 - Weitere Vervollständigung des Kontrakts in jedem Verhandlungsschritt

- o Teile des Kontraktes können mehrmals verhandelt werden und sich dadurch ändern.

Generelle Funktionen zur Verhandlungsunterstützung:

- Austauschen von Textnachrichten zwischen den Verhandlungspartnern
 - o Echtzeitinteraktion (Chat-System, Austausch von kurzen Nachrichten)
 - o Messaging durch E-Mails (Austausch von längeren Nachrichten und Texten)
 - o bei lösungsorientierten Systemen meist unstrukturierte Nachrichten
- Überblick über den Verhandlungsprozess verschaffen (besonders wichtig für ‚Process-Support‘)
 - o Aufzeichnen der Verhandlung und Darstellen durch eine History-Funktion
 - o Anzeige aller ausgetauschten Nachrichten
 - o Grafische Anzeige der Bewertung von Angeboten und Gegenangeboten, Darstellen von Teilergebnissen
- Unterstützung des Verhandlungsprozesses durch das Aufstellen von Turn-Taking-Regeln nach der Konversationsanalyse

4 Kommunikationsaspekte in Verhandlungssystemen

Einer der wichtigsten Aspekte von Verhandlungssystemen ist die Realisierung der notwendigen Kommunikation.

Diese Kommunikation findet zwischen den beteiligten Akteuren in einer speziellen Verhandlungssituation statt.

Eine in der Praxis bereits vorhandene Grundlage für die interorganisationelle Kommunikation zwischen Electronic Business Systemen ist das Business Messaging, mit dem orientiert an standardisierten Formaten verbindlich geschäftliche Daten zwischen Unternehmen ausgetauscht werden. Diese Kommunikationsgrundlage sollte um Verhandlungsfunktionalitäten erweitert werden, um eine umfassendere Interaktion zwischen Unternehmen zu ermöglichen.

Als Ausgangspunkt für die Realisierung der Verhandlungskommunikation wird zunächst nach geeigneten Modellierungsmöglichkeiten gesucht. Hier erweist sich die Sprechakttheorie als ein nützliches und grundlegendes Instrument, da sie unter anderem den autonomen Interaktionscharakter der Verhandlungskommunikation gut abbildet.

4.1 Fallbeispiele für Verhandlungssituationen und Verhandlungskommunikation

Bei der Entwicklung von Lösungen zur Unterstützung von Verhandlungen und Verhandlungskommunikation muss zunächst im Detail geklärt werden, wo Verhandlungskommunikation auftritt.

Sollen im Software Engineering Lösungen entwickelt werden, so bieten die Methoden des klassischen Requirements Engineering entsprechende Vorgehensweisen und Modellierungsmethoden.

Für die Modellierung von Verhandlungssituationen und -kommunikation wird auf die in Kapitel 2 eingeführten Modellierungsmethoden zurückgegriffen. Insbesondere findet die E³-Value Methode für die Modellierung

von Geschäftsmodellen im E-Business Verwendung. Zur Darstellung der Kommunikationsabläufe kommen UML-Sequenzdiagramme zum Einsatz.

Im Kapitel 3 zu Verhandlungssystemen wurde auf die derzeit mögliche Unterstützung von Verhandlungen im Electronic Business eingegangen und Beispiele wie Auktionen unter Verbrauchern (z.B. eBay) vorgestellt.

Die folgenden Fallbeispiele setzen auf solchen möglichen realen Szenarien auf und erweitern sie introspektiv. Die Fallbeispiele stellen dabei lediglich typische Fälle dar und sollen einen guten Einblick in die Thematik ermöglichen. Inhaltlich orientieren sie sich am weiter oben vorgestellten Bereich des E-Procurement.

4.1.1 Ein grundlegendes Fallbeispiel: Preisagentur Spürhund GbR

Im Fallbeispiel ‚Preisagentur Spürhund GbR‘ geht es um die Beschaffung der DVD ‚Best Of All‘, wobei durch die ‚Spürhund GbR‘ eine Preisrecherche mit nachfolgender Preisverhandlung im Auftrag eines Kunden durchgeführt wird. Daran sind als Akteure im Wesentlichen beteiligt:

- Spürhund GbR (als Vermittler)
- AllMedia (als Verkäufer)
- EasyMedia (als Verkäufer)
- AllesBillig (als Verkäufer)
- BestChoice (als Verkäufer)

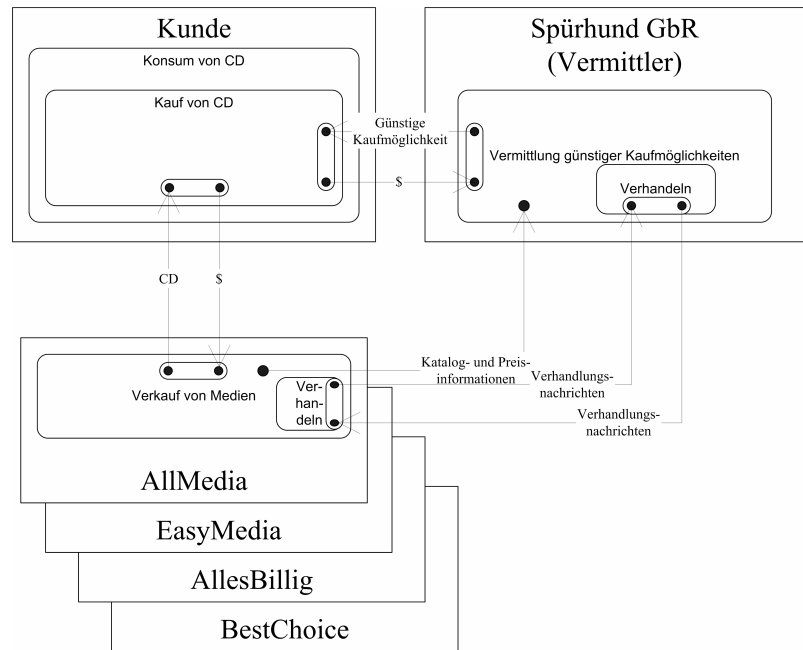


Abbildung 4.1: E³-Value Modell Fallbeispiel Spürhund GbR

Das an die e³-Value Modellierungsmethode angelehnte Diagramm in Abbildung 4.1 zeigt ebenfalls diese Akteure mit ihren dafür notwendigen Aktivitäten. Die Spürhund GbR kann von jedem der Anbieter die notwendigen Katalog- und Preisinformationen einholen. Bei Bedarf kann mit den Anbietern verhandelt werden. Ist eine günstige Kaufmöglichkeit gefunden, so kann diese dem Kunden mitgeteilt werden der dann den Kauf vollzieht.

In dem Modell ist der Kunde, in dessen Auftrag die Spürhund GbR tätig ist, der Vollständigkeit halber dargestellt. Er spielt jedoch für die weitere Betrachtung des Verhandlungsbeispiels zwischen Vermittler und Anbietern keine Rolle. Eine Vereinfachung des Modells ist die einheitliche Darstellung der Anbieter als Market Segment; wie im weiteren Verlauf deutlich wird, verfügt der Anbieter ‚AllesBillig‘ nicht über eine Verhandlungsfunktionalität.

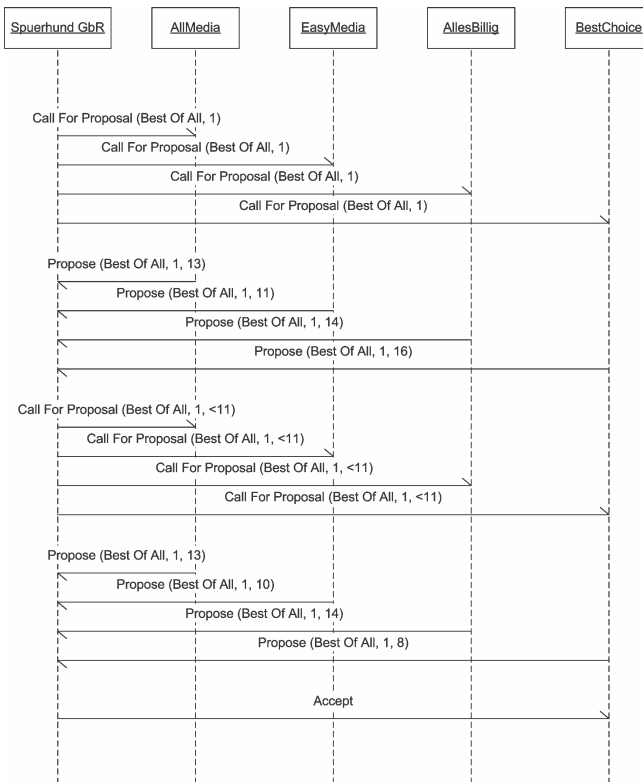


Abbildung 4.2: Ablauf Fallbeispiel Spürhund GbR

Die Kommunikation des Fallbeispiels hat den folgenden Ablauf, der auch durch Abbildung 4.2 deutlich wird:

- Die Spürhund GbR fragt zunächst bei den ihr bekannten vier Verkäufern der CD ‚Best Of All‘ nach dem Preis der CD an.
- Alle Verkäufer können die CD auch wirklich liefern. Die Verkäufer liefern jeweils ihre Preisangaben. Günstigster Anbieter ist EasyMedia mit 11 EUR.
- In einer zweiten Phase startet die Spürhund GbR eine erneute Preisanfrage an alle vier Verkäufer, jedoch mit der Zusatzangabe, ob ein Preis kleiner als 11 EUR für die CD möglich ist.
- Alle vier Verkäufer reagieren mit erneuten Preisangeboten:
 - o AllMedia bleibt bei ihrem vorherigen Preis von 13 EUR.
 - o EasyMedia senkt ihren bisher günstigsten Preis noch mal und bietet für 10 EUR an.
 - o AllesBillig ist nicht in der Lage die Anfrage ihrem Sinn nach richtig zu verarbeiten und reagiert nach Standardeinstellung mit ihrem regulären und auch bisherigen Preisangebot von 14 EUR.

- o BestChoice, der bisher teuerste Anbieter, ist sehr bemüht um den Handel und bietet den günstigsten Preis von 8 EUR.
- Die Spürhund GbR akzeptiert das Angebot von BestChoice.

An diesem Fallbeispiel wird in der ersten Phase deutlich, dass auch eine einfache Art der Verhandlung ohne besonders ausgestattete Client-Systeme möglich ist. Es ist lediglich eine Preisauskunft über die Katalogsysteme notwendig.

In der Vergangenheit haben sich einige Anbieter gegen eine reine Beurteilung über den Preis gesperrt und versucht Abwehrmaßnahmen gegen ein solches Vorgehen einzuleiten. Siehe dazu auch das Thema der Shopping Agenten in Kapitel 2.

In der zweiten Phase kommt implizit eine Auktionsart mit einer Bietrunde zur Ausführung. Es wurden jedoch vorher keinerlei explizite Regeln vereinbart.

4.1.2 Freie Verhandlungen bei der Fahrzeugpool GmbH

Im Fallbeispiel ‚Fahrzeugpool GmbH‘ geht es um den Kauf von 250 Litern Synthetik-Öl SF 17. Daran sind die folgenden Akteure beteiligt:

- Fahrzeugpool GmbH (als Käufer)
- SupplyService GmbH (als Verkäufer)
- MegaOil AG (als Verkäufer)

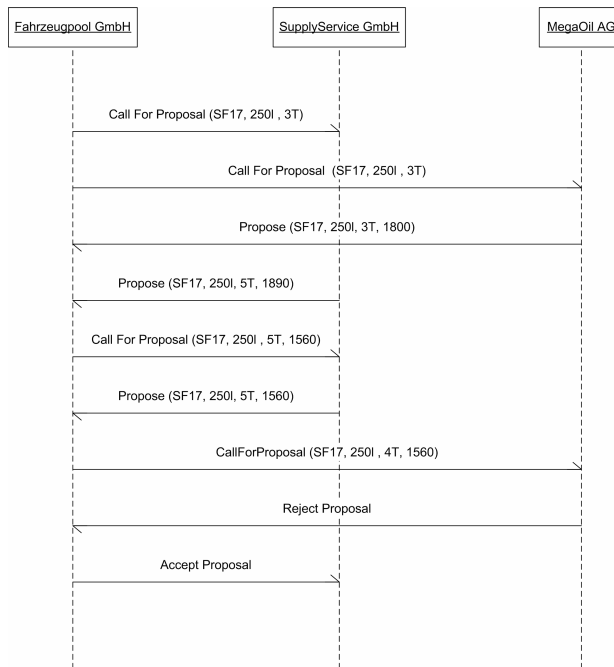


Abbildung 4.3: Ablauf Fallbeispiel Fahrzeugpool GmbH

Die Kommunikation des Fallbeispiels hat den folgenden Ablauf, der auch durch Abbildung 4.3 deutlich wird:

- Die Fahrzeugpool GmbH schickt eine unverbindliche Anfrage an die SupplyService GmbH und an die MegaOil AG über die Lieferung von 250 Litern Synthetik-Öl SF17 in 3 Tagen.
- Die SupplyService GmbH kann das geforderte Datum nicht einhalten und schickt ein modifiziertes Angebot mit 5 Tagen Lieferzeit und dem dazugehörigen Preis.
- Die MegaOil AG sendet der Fahrzeugpool GmbH ein Angebot entsprechend der Anfrage inklusive des Preises.
- Die Strategie der Fahrzeugpool GmbH ist es, einen möglichst günstigen Preis zu erzielen und so macht sie der SupplyService GmbH einen entsprechenden unverbindlichen Gegenvorschlag. Dabei akzeptiert sie die Änderung des Lieferdatums zugunsten eines nach unten korrigierten Preises, da sie gerade noch ausreichend Reserven für 5 Tage hat.
- Da die SupplyService GmbH an einem hohen Umsatz interessiert ist, akzeptiert sie die neuen Bedingungen und sendet ein identisches Angebot.
- Die Fahrzeugpool GmbH fragt nun bei der MegaOil AG an, ob diese nicht auch zu dem niedrigeren Preis, aber noch vor dem späten Liefertermin der SupplyService GmbH liefern könnte.

- Die MegaOil AG lehnt diese Konditionen ab.
- Die Fahrzeugpool GmbH akzeptiert das neu vorliegende Angebot der SupplyService GmbH.

An diesem Fallbeispiel wird deutlich das die Angebote von jeder der beteiligten Parteien gemacht werden können und keinem festen vorher vereinbarten Ablauf unterliegen müssen.

4.1.3 Feste Protokolle bei den Stadtwerken Tüchtigburg

In diesem Fallbeispiel kaufen die Stadtwerke Tüchtigburg, aufgrund eines Ausfalls ihres eigenen Kleinkraftwerks, kurzfristig innerhalb weniger Minuten ein zusätzliches Strompaket ein.

Dieses Fallbeispiel ist in einer fiktiven Zukunft angesiedelt, in der solche flexiblen Abschlüsse über Kleinkontingente und ohne große Rahmenvereinbarungen auch in der Strombranche möglich sind. Eine entsprechende Flexibilisierung zeichnet sich derzeit ab.

An dem Fallbeispiel sind die folgenden Akteure beteiligt:

- Stadtwerke Tüchtigburg
- MegaPower
- ÖkoStrom
- CityElectricNet

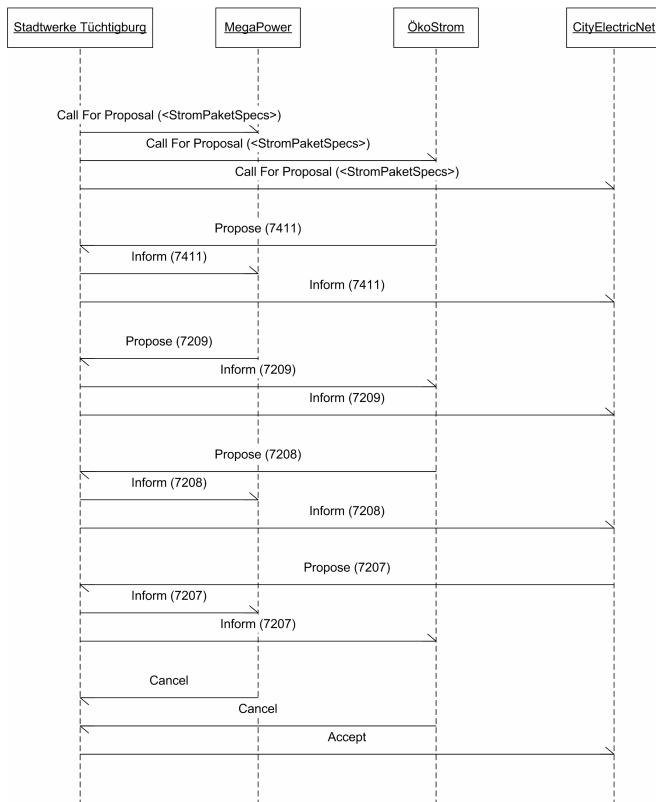


Abbildung 4.4: Ablauf Fallbeispiel Stadtwerke Tüchtigburg

Die Kommunikation des Fallbeispiels hat den folgenden Ablauf, der auch durch Abbildung 4.4 deutlich wird:

- Zunächst starten die Stadtwerke Tüchtigburg durch die Nachfrage des benötigten Strompaketes bei den möglichen Anbietern die vorher für solche Fälle vereinbarte umgekehrte englische Auktion.
- Durch ÖkoStrom werden zunächst 7411 Verrechnungseinheiten für die Lieferung des nachfragten Strompaketes geboten. Die Stadtwerke Tüchtigburg informieren die anderen Anbieter über dieses Gebot.
- MegaPower bietet nun 7209 Verrechnungseinheiten. Die Stadtwerke Tüchtigburg informieren die anderen Anbieter über dieses Gebot.
- ÖkoStrom bietet nun 7208 Verrechnungseinheiten. Die Stadtwerke Tüchtigburg informieren die anderen Anbieter über dieses Gebot.

- CityElectricNet bietet nun 7207 Verrechnungseinheiten. Die Stadtwerke Tüchtigburg informieren die anderen Anbieter über dieses Gebot.
- MegaPower und ÖkoStrom scheiden aus der Auktion aus.
- Die Stadtwerke Tüchtigburg nehmen das Angebot von CityElectricNet an.

In diesem Fallbeispiel wird das vorher für solche Verhandlungen explizit vereinbarte Protokoll einer umgekehrten englischen Auktion verwendet.

Als problematisch im Hinblick auf Falschinformationen kann sich die Übernahme der Rolle des Auktionators durch den Nachfrager erweisen.

Eine mögliche Lösung ist die Beauftragung eines neutralen Akteurs durch die Stadtwerke Tüchtigburg, der dann die Rolle des Auktionators übernimmt.

Die vorgestellten Fallbeispiele verdeutlichen Anforderungen an eine Infrastruktur für Verhandlungen im E-Business.

4.2 Business Messaging

Mit *Business Messaging* werden alle Aspekte des elektronischen Nachrichtenversands zwischen Handelspartnern bezeichnet. Es werden also die Kommunikation zwischen den Unternehmen und die Systeme die diese Kommunikation ermöglichen, betrachtet.

Der elektronischen Kommunikation zwischen Unternehmen liegen besondere Strukturen zugrunde, die zunächst dargestellt und analysiert werden. Darauf folgt eine Analyse der wichtigsten Standards im Business Messaging und deren Bezug zu Verhandlungen.

4.2.1 Strukturen des Business Messaging

Hauptgegenstand des Business Messaging sind einerseits die Geschäftsdokumente die zwischen Unternehmen ausgetauscht werden und auch als Repräsentation von Verhandlungsobjekten dienen können und andererseits die Messaging Frameworks, die dynamisch den Ablauf der Kommunikation gewährleisten.

Die konzeptionellen Ebenen der Kommunikation zwischen Unternehmen lassen sich einteilen in:

- Geschäftsprozesse
- Applikationen

- Daten
- Kommunikationsebene
- Physische Ebene

Grundlage einer sinnvollen Kommunikation zwischen Unternehmen ist die Interoperabilität der Nachrichten und Systeme auf jeder dieser Ebenen. Der in diesem Bereich verfolgte Weg, um Interoperabilität zu gewährleisten, ist Standardisierung.

Geschäftsprozesse sind letztlich individuell wie jede Organisation, in der sie ablaufen. Sie können damit eine Grundlage für den individuellen Wettbewerbsvorteil eines Unternehmens sein. Durch die Entwicklung des Electronic Business werden jedoch auch Geschäftsprozesse immer weiter automatisiert. Dabei geht es auch um die methodische und elektronische Unterstützung der Geschäftsprozesse. Und diese Unterstützung bietet Standardisierungspotential, während die Prozesse selber nicht standardisiert werden sollten.

Applikationen als Anwendungssoftware spiegeln das Wissen um wertvolle und gute Problemlösungen wieder, die ein Unternehmen entwickelt hat und anwendet. Daraus resultiert jedoch auch eine sehr inhomogene IT-Landschaft, die viel Potential für Standardisierung bietet. Allerdings sind unternehmensindividuelle Applikationen, wie auch die speziellen Geschäftsprozesse ein Wettbewerbsvorteil und können letztlich so nicht umfassend standardisiert werden.

Daten umfassen in diesem Kontext die Geschäftsdokumente aller Art die in standardisierten Nachrichten übertragen werden. Beispiele für Geschäftsdokumente sind Produktinformationen, Klassifikationen für Kataloge sowie Bestellaufträge. Hier ist auch eine inhaltliche Standardisierung möglich und sinnvoll. Als erstes sind die EDI Standards zu nennen, über die schon seit Jahrzehnten standardisierte Daten ausgetauscht werden, obwohl sich auch hier verschiedene ‚Standards‘ ausgebildet haben. Seit einigen Jahren bilden sich neue Standards heraus, die auf Internet- und XML-Technologie aufbauen (vgl. (Frank, 2001)), jedoch ansonsten von Funktion und Konzept die EDI-Technologie adaptieren.

Die *Kommunikationsebene* stellt nun auf Basis der physischen Ebene die Verbindung zwischen den kommunizierenden Unternehmen her. Das ISO/OSI Referenzmodell mit seinen sieben Schichten (Application, Presentation, Session, Transport, Network, Data Link, Physical) findet zwar praktisch kaum Anwendung, strukturiert den Bereich jedoch verständlich und ist ein sinnvoller Ansatzpunkt. Inzwischen werden oft nur noch die zwei großen Schichten der ‚Middleware‘ (z.B. CORBA) und der Transport Networks (WAN, z.B. Internet und LAN, z.B. Ethernet) betrachtet. Auf der Kommunikationsebene ist die Standardisierung in bestimmten Bereichen bereits weit fortgeschritten.

Die *physische Ebene* ist heutzutage weitgehend standardisiert und liegt nicht im Fokus der Betrachtungen in dieser Arbeit. Feste Anschlüsse an Telekommunikationsnetze, Inhouse-Installationen und mobile und wireless Zugänge sorgen für Zugang zu LAN's und WAN's oder speziellen Intra- und Extranets.

Neben diesen Standardisierungsbestrebungen sind bei der Analyse des Business Messaging auch Kriterien notwendig, die die derzeitige Situation wiedergeben können.

Von Masseti et al. (Masseti, et al., 1996) wurden Messkriterien zur Nutzung von EDI durch Organisationen vorgeschlagen, die auch allgemein auf den Dokumentenaustausch im Business Messaging anzuwenden sind.

Die *Tiefe* (depth) bezieht sich darauf wie weit elektronische Daten in bestehende Applikationen integriert werden. Die Erhöhung der Tiefe umfasst eine stärkere Integration des elektronischen Datenaustausches mit den Geschäftsprozessen einer Organisation. Resultierende Vorteile sind insbesondere die verstärkte gemeinsame Nutzung von Informationen und die verbesserte Koordination von Prozessen.

Der *Umfang* (breadth) bezieht sich auf die Anzahl der Handelspartner, die mit dem Unternehmen in elektronischem Datenaustausch stehen. Diese Anzahl lässt sich durch entsprechende Anregungen für die Handelspartner erhöhen. Ein höherer Umfang senkt die Kosten der Datenerfassung und des herkömmlichen Postversands oder Dokumentenaustauschs. Hilfreich bei der Erhöhung des Umfangs sind auch besonders die so genannten Hubs.

Die *Vielfalt* (diversity) bezieht sich darauf welche Spanne an verschiedenen elektronischen Dokumenten ausgetauscht werden kann. Hier ist die Identifikation von zusätzlichen Handelstransaktionen, die auch elektronisch abgewickelt werden können, wichtig. Oft ist dann zusätzliche Implementationen notwendig. Eine erhöhte Vielfalt reduziert jedoch wiederum die Transaktionskosten durch Nutzung des neuen und kostengünstigeren Mediums.

Der Einsatz von Business Messaging Lösungen orientiert sich, wie auch der Einsatz anderer Technologien, daran inwieweit die Technologie zur Erreichung der Unternehmensziele beitragen kann. Banerjee et al. (Banerjee, et al., 2002) entwickeln ausgehend von Unternehmenszielen und der vorgestellten Kategorisierung von Masseti et al. (Masseti, et al., 1996) ein Rahmenwerk in dem unter anderem die möglichen Technologien zum elektronischen Datenaustausch zur Einordnung dienen. Banerjee et al. legen den Fokus auf den Austausch elektronischer Dokumente in strukturierter und unstrukturierter Form und sprechen kurz von EBDI

(Electronic Business Document Interchange). Es wird die folgende Einteilung vorgeschlagen:

- EBDI mit Nutzung eigener Infrastruktur
 - VAN basiertes EDI
 - Web-Formulare durch dominanten Handelspartner bereitgestellt
 - XML basierter Dokumentenaustausch
 - Standardisierter Internet-/Web-EDI
- EBDI unter Nutzung von Infrastruktur von Dritten
 - EDI/Fax und EDI Übersetzer von Dritten angeboten
 - Web-Formulare von Dritten angeboten
- EBDI Infrastruktur und Dienstleistungen werden von Dritten direkt angeboten
 - Vertikale und funktionale Hubs
 - Handelsbörsen und Auktionshäuser

Nach der strukturellen Betrachtung des Business Messaging wird nun auf das Thema der Standards im Business Messaging eingegangen.

4.2.2 Klassifikation der Standards des Business Messaging

Die Zahl der im E-Business-Bereich existierenden Standards liegt im dreistelligen Bereich. Eine Quelle spricht sogar von 1200 (EIDX, 2003) Standards. Viele dieser Standards sind in so fern redundant, als das sie gleiche Problemfelder angehen.

Auch wenn unter dieser großen Anzahl viele der Standards spezielle Nischen bedienen und lediglich eine zweistellige Zahl einen größeren Bekanntheits- und Verbreitungsgrad haben, so ist das Dilemma offensichtlich: es erschwert die Interoperabilität zwischen verschiedenen Systemen, was insbesondere im B2B-Bereich ein Problem darstellt – im B2C-Bereich ist Interoperabilität von nicht ganz so großer Bedeutung, da Frontend-seitig der Kunde sich mittels Browser über das Angebot informiert, aber bisher in der Regel kein Bedürfnis hat, die ihm angebotenen Informationen weiter zu verarbeiten.

Bevor auf einige der vorhandenen Standards genauer eingegangen wird, soll erst einmal der Versuch unternommen werden, die Standards bezüglich dessen, was sie leisten (sollen) zu kategorisieren. Diese Strukturierung kann den Umgang mit der Vielzahl der Ansätze und Vokabulare erleichtern und darlegen, wie diese sinnvoll zusammenspielen können.

4.2.2.1 Standards für Geschäftsdokumente

Standards, die zu dieser Kategorie gehören, definieren Aufbau und Struktur von Geschäftsdokumenten, also:

- welche Informationen vorhanden sein müssen
- welche Informationen optional sind
- in welcher Reihenfolge diese Informationen angegeben werden
- Restriktionen bezüglich der Wertebereiche und Kardinalitäten
- wie diese Informationen repräsentiert werden, also beispielsweise durch die Angabe eines Datumsformates. Für einige dieser Informationen existieren internationale Standards, beispielsweise für Länder- (ISO 3166) und Sprachkürzel (ISO 639-1) oder Datumsformatierungen (ISO 8601, diverse andere Konventionen).

Die neueren Standards für Geschäftsdokumente nutzen alle die Repräsentation der Daten in XML und die Strukturdefinition über DTD oder XML-Schema.

Standards für Geschäftsdokumente definieren in der Regel nicht wie, also über welche Transportmechanismen diese Dokumente ausgetauscht werden. Sie stoßen daher in die gleiche Zielrichtung wie EDI, allerdings in unterschiedlichem Umfang.

Unter dem Gesichtspunkt Verhandlungen können Geschäftsdokumente auch als Repräsentation von Verhandlungsobjekten im Sinne des Basismodells für Verhandlungen dienen. Dieser Aspekt wird an späterer Stelle ausführlicher diskutiert.

4.2.2.2 Zusammenspiel der B2B XML-Standards

Die Standards, welche sich auf die Definition von Geschäftsdokumenten fokussieren sind in der Regel eher älter als solche, die Aspekte wie Messaging, Prozesse oder Verzeichnisdienste als Gegenstand haben. Historisch gesehen kann man also von einem „Bottom-Up“-Ansatz sprechen. Das trifft, wie in einem der folgenden Abschnitte noch genauer beschrieben wird, auch auf die „Mutter“ des elektronischen Geschäftsdatenaustausches, dem EDI, zu.

4.2.3 Überblick über die B2B-XML-Standards

Im den folgenden Abschnitten wird ein Überblick über die bekannteren XML-basierten Standards im B2B-Bereich gegeben. Dabei sollen neben einem allgemeinen Überblick Besonderheiten herausgearbeitet werden, die den entsprechenden Standard von anderen unterscheidet. Dabei soll

ein Augenmerk Insbesondere darauf liegen, ob und wenn ja, welche verhandlungsrelevanten Eigenschaften oder Konzepte in ihnen berücksichtigt sind und in wie weit diese Vorarbeiten sinnvoll in den Kontext dieser Arbeit einzubringen sind.

Ein für den Einstieg brauchbaren Überblick findet sich in (Fitzgerald, 2001). Für detaillierte Informationen ist das Studium der Standards unerlässlich:

- BMEcat (BMEcat, 2003)
- CXML (cXML)
- ebXML (ebXML, 2003; Schilling, 2002)
- openTRANS (openTRANS, 2003)
- UBL (OASIS)
- xCBL (xCBL)

4.2.3.1 EDI (UN/EDIFACT und ANSI/X12)

EDI (Electronic Data Interchange) ist die Mutter der elektronischen B2B-Kommunikation. Nach dem in den 70er und 80er Jahren die Möglichkeit der elektronischen Datenkommunikation (durch das Vorhandensein entsprechender EDV-Anlagen und großflächigen Verbreitung der Datenfernübertragung) prinzipiell gegeben war, erschien es sinnvoll, den mit dem Handel verbundenen Austausch von Dokumenten wie Aufträgen, Bestellungen, Bestätigungen usw. elektronisch abzuwickeln.

Hauptaufgabe der EDI-Aktivitäten war die Definition von standardisierten Geschäftsdokumenten und Geschäftstransaktionen (vgl. (Hill, 1989; Sokol, 1994; Stultz, 2001; Weitzel, et al., 2001)). Die Geschäftstransaktionen definieren eine Abfolge des Austausches von Geschäftsdokumenten, um ein bestimmtes Geschäftsziel, wie die Abwicklung einer Bestellung zu erreichen.

In einer ersten Phase entstanden eine Reihe branchenspezifischer Standards, zuerst in der amerikanischen Automobilindustrie. Schnell kam der Wunsch auf, die branchenspezifischen Standards zu vereinheitlichen und ein universell einsetzbares Grundgerüst bezüglich der Datenstruktur zu schaffen.

Die Definition eines übergreifenden Standards wurde für so sinnvoll gehalten, dass gleich zwei geschaffen wurden. 1983 entstand die erste Version des EDI-Standard X.12 der amerikanischen Standardisierungsorganisation ANSI. Zwei Jahre später, im Jahre 1985, begannen die Vereinten Nationen, genauer gesagt das UN/TRADE Komitee, mit der Entwicklung des internationalen EDI-Standards UN/EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport).

Die beiden Standards sind nicht interoperabel zueinander. Traditionell ist es so, dass in Europa EDIFACT und in den USA X.12 verwendet wird. Zwar gibt es Bestrebungen, vor allem von Seiten der UN, die beiden

Standards anzugleichen – allerdings werden diese Anstrengungen vor allem von Seiten der ANSI mit begrenzten Eifer vorangetrieben (Stultz, 2001) und in der genannten Quelle mit der schleppenden und bis zum heutigen Tage nicht vollzogenen Einführung des metrischen Systems verglichen.

4.2.3.1.1 Transport

Die offiziellen EDI-Standards definieren lediglich die Struktur der ausgetauschten Dokumente – sie klammern den Transportaspekt, also wie EDI-Dokumente von einem Teilnehmer zu einem anderen kommen, aus. Daher ergeben sich prinzipiell mehrere Möglichkeiten für die Wahl der Übertragungsmedien und Netzwerke.

Zu Beginn wurden mangels Verbreitung des Internets spezielle Netzwerke, so genannte Value-added Networks (VAN) eingerichtet, die von entsprechenden Dienstleistern betrieben wurden und gewissermaßen einen begrenzten (im Sinne von nicht offenen) Marktplatz implementierten.

Die zentrale Rolle spielt dabei ein Hub, der eine Vermittlung oder Zuordnung der Nachrichten in ein Postfach für den jeweiligen Empfänger implementiert (Store-and-Forward). Diese Topologie ist natürlich insbesondere für die Kommunikation bei einer größeren Anzahl beteiligter Unternehmen vorteilhaft gegenüber einer in einer kleineren Zahl der Fälle praktizierten Punkt-zu-Punkt Verbindung über Stand- oder Wählleitungen.

Nach dem eben gesagten ist es natürlich ebenso möglich, die Nachrichten per FTP oder E-Mail zu verschicken. Dies wird heutzutage auch tatsächlich praktiziert.

Nach und nach entwickelten sich Bemühungen, den B2B-Geschäftsdatenaustausch über gängige Internet-Technologien abzuwickeln, wobei man drei Ansätze unterscheiden kann:

Im ersten Ansatz werden herkömmliche EDI-Dokumente (also ASCII-basierte Textdateien) über E-Mail oder FTP ausgetauscht.

Der zweite Ansatz geht in sofern weiter, als das zwar bestehende EDI-Semantiken (also im wesentlichen die durch die entsprechenden Standards definierten Transaktion als Abfolge von definierten Geschäftsdokumenten) beibehalten werden, die Dokumente selbst jedoch in XML formatiert werden – bestes Beispiel für diesen Ansatz ist das in einem der folgenden Abschnitte besprochene XCBL.

Ferner gibt es Standards, die sowohl auf der Transport- als auch auf der Dokumentenebene auf Internetstandards basieren, aber keine Verbindung mehr zu dem klassischen EDI haben. Sie definieren entweder die Transaktionen neu, oder bilden nur Teilaspekte ab, wie die Standardisierung einer Teilmenge der denkbaren Geschäftsdokumente – ein Beispiel ist BMEcat, welches keine tief greifenden EDI-Wurzeln hat und

darüber hinaus lediglich ein Teilgebiet, nämlich den Austausch von Katalogdaten, in Betracht zieht.

Zuletzt sei noch WebEDI erwähnt – dieser Begriff wird nicht ganz einheitlich gehandhabt, in der Regel handelt es sich hierbei jedoch um die Möglichkeit, EDI Nachrichten über eine Webbrowser-basierte Oberfläche zu erstellen. Hier bieten größere Unternehmen (Automobilkonzerne (VW, 2003) oder Großhändler) kleineren Zulieferern oder Kunden die Möglichkeit, EDI-konforme Nachrichten zu erzeugen, ohne selbst große Investitionen in Hard- und Software tätigen zu müssen – ein Internetanschluss und ein Webbrowser genügt. Im Gegensatz zum klassischen EDI handelt es sich also um eine Client-Server Architektur.

4.2.3.1.2 Datenformat

EDI Nachrichten, gleich ob ANSI X.12 oder EDIFACT-konform, sind ASCII-basierte Textdateien, in denen die einzelnen Werte durch Trennzeichen beendet werden. Sie können also mit einem beliebigen Text-Editor geöffnet werden, sind jedoch der aufgrund Länge der Dokumente und der fehlenden Beschriftung der einzelnen Werte kaum vom Menschen lesbar.

Ein EDIFACT-Dokument grundsätzliche folgendermassen aufgebaut:

- Einzelne Zahlen oder Strings bilden ein Datenelement
- Mehrere Datenelemente bilden eine Datenelementgruppe
- Mehrere Datenelementgruppen bilden ein Segment
- Mehrere Segmente bilden Segmentgruppen
- Mehrere Segmentgruppen bilden eine EDIFACT-Nachricht, also ein Geschäftsdokument eines bestimmten Typs
- Die verschiedenen EDIFACT-Nachrichtentypen bilden letztendlich den EDIFACT-Standard bzw. einen Subset

Datenelemente entsprechen einer atomaren Informationseinheit. Ein Beispiel: es könnte sich um ein Datum handeln und der Standard definiert genau, wie dieses Datum formatiert sein soll, also beispielsweise durch eine Angabe wie YYMMDD 6/6, wobei die beiden letzten Zahlen die minimale und maximale Länge des Datenelementes kennzeichnen. Datenelemente werden durch ein Trennzeichen voneinander getrennt.

Mehrere Datenelemente können eine Datenelementgruppe bilden (beispielsweise Strasse und Hausnummer).

Ein aus mehreren Datenelementgruppen bestehendes Segment könnte beispielsweise eine Lieferadresse sein. Ein Segment wird durch einen Zeilenumbruch beendet.

Mehrere von der Bedeutung eng zusammengehörende Segmente (z.B. Absender-, Empfänger-, Rechnungsadresse) bilden eine Segmentgruppe, mehrere hier von bilden schließlich das komplette Geschäftsdokument.

Ein Beispiel für eine (teilweise gekürzte) EDIFACT-Nachricht findet sich im Anhang.

Als abschließende Bemerkung soll noch erwähnt werden, dass Rechnungen im elektronischen Format in vielen Ländern nicht rechtsgültig sind (in Deutschland ist dies jedoch seit dem 1.1.2002 der Fall, sofern die Rechnung elektronisch signiert ist (§ 14 Abs. 4 S. 2 des Umsatzsteuergesetzes)). Hier wird in der Regel zum Abschluss eine Rechnung in Papierform erzeugt. Dieser Umstand betrifft natürlich nicht nur das klassische EDI, sondern auch XML-basierte Geschäftsdokumente.

4.2.3.2 EbXML

EbXML ist modulares Framework, welche eine XML-basierende Infrastruktur für die elektronische Geschäftsabwicklung benutzt. Modular bedeutet in diesem Zusammenhang, dass es ca. ein halbes Dutzend von Spezifikationen gibt, die nicht zwangsläufig in ihrer Gesamtheit implementiert werden müssen.

Getrieben ist der Standard von folgenden zwei Organisationen:

- United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT). Bekannt geworden ist diese Organisation insbesondere durch den internationalen UN/EDIFACT EDI-Standard
- OASIS ist eine mitglieder-gestützte not-for-profit Organisation von etwa einhundert Firmen aus dem IT-Sektor, deren Ziel die Entwicklung von interoperablen Standards im XML-Umfeld ist

Die im November 1999 gestartete Initiative hatte eine vorher begrenzte Laufzeit von 18 Monaten - an ihrem Ende stand im Mai 2001 die Publikation der ersten Version der entsprechenden Standarddokumente.

Die ebXML-Initiative möchte insbesondere Unternehmen kleiner und mittlerer Größe (die sog. KMU oder englisch SMB - small and medium businesses) ansprechen. Einer der Väter von XML und im ebXML-Umfeld engagierten Personen, Jon Bosak, drückt dies folgendermaßen aus:

"Most of the emphasis so far has been on how to enable big multinationals to do business with each other, while relatively little attention has been paid to how we enable small companies to compete in the same virtual business environment. But most of the world's business is, in fact, done by small companies. I want to enable a five-person manufacturer of fabrics in Pakistan to bid on supplying a hundred units out of a purchase request for a million seat covers from General Motors. Seeing both parties to this transaction benefit equally is for me what this is all about."

So neu wie Initiativen wie ebXML sind, so unfertig erscheinen leider an der einen oder anderen Stelle auch die dahinter liegenden Absichten: Dieses Beispiel ist bezeichnenderweise nämlich eher ein schlechtes: ein Automobilhersteller wird sowohl aus Verwaltungsgründen als auch aus Gründen der Qualitätssicherung kein Interesse daran haben, ein derartigen Großauftrag an dutzende oder Hunderte von kleinen Firmen zu vergeben.

Neben zwei "Meta"-Standards, der ebXML Technical Architecture Specification und der ebXML Requirements Specification basiert das ebXML Framework auf folgenden fünf Spezifikationen:

- Business Process Specification Schema definiert ein XML-Schema für die Beschreibung von Geschäftsprozessen
- Registry Information Model beschreibt das interne Datenmodell für eine ebXML Registry
- Registry Services Specification definiert die Dienste, die eine ebXML-konforme Registry zur Verfügung stellt
- Collaboration-Protocol Profile and Agreement Specification behandelt Vereinbarungen zwischen zwei Partnern bezüglich der Fähigkeiten hinsichtlich technischer Art und unterstützten Geschäftsprozessen
- Message Services Specification spezifiziert einen zuverlässigen und sicheren Dienst für die Nachrichtenübertragung – er basiert mittlerweile auf SOAP

Die Nutzung des ebXML Standards kommt trotz der organisatorisch zentralen Förderung durch die UN nur schwer voran. Im Kontext dieser Arbeit entstand schon Anfang 2002 ein erster Prototyp zur praktischen Evaluation des ebXML Standards, der die grundsätzliche Einsetzbarkeit der Spezifikationen zeigte (Schilling, 2002).

4.2.3.3 CXML

CXML ist ein Standard für Katalogdaten, also Geschäftsdokumente. Eine besondere Eigenschaft von cXML ist die Möglichkeit, Kataloge im sog. Punch-Out Verfahren zu verteilen. Bei dieser Anwendung werden die Katalogdaten lieferantenseitig gehalten und die entsprechend relevanten Teilinformationen, beispielsweise qualifiziert durch eine Suchabfrage, online zum Kunden übertragen. Dieser benutzt dabei in der Regel ein Front-End, welches ihm vom Lieferanten zur Verfügung gestellt wird. Über diesen Wege kann auch die eigentliche Bestellung abgewickelt werden.

Vorteil dieser Lösung gegenüber dem Import in das System des Beschaffenden ist unter anderem die inhärente Aktualität von Preis- und Verfügbarkeitsinformationen.

Für eine detaillierte Diskussion der möglichen Szenarien der Kataloghaltung, sowie deren Vor- und Nachteile siehe (Böge, 2003), weitere Informationen zu CXML und Punch-Out Variationen siehe (cXML, 2003; Laka, 2002).

4.2.3.4 XCBL

xCBL (XML Common Business Library) ist einer der reiferen Standards für elektronische Geschäftsdokumente. Die erste Version stammt bereits aus dem Jahre 1997, seit März 2003 liegt xCBL in der Version 4.0 vor. Ursprünglich wurde xCBL von Veo Systems unter dem Namen CBL entwickelt, sowohl Firma als auch Standard wurden zwei Jahre später von Commerce One akquiriert und erlangten durch deren Produkte einen größeren Verbreitungsgrad. Commerce One ist bis heute federführend, der Standard kann jedoch von anderen Parteien kostenlos benutzt und implementiert werden.

Die XML Common Business Library kennt das Konzept sog. Transaktionen, die eine Abfolge des Austausches von bestimmten Geschäftsdokumenten definiert.

Es wurde sinnvollerweise nicht der Versuch unternommen, die Geschäftstransaktionen und damit die Art und Weise wie in der heutigen Praxis Geschäfte abgewickelt werden, neu zu erfinden. Stattdessen wurden diese aus den existierenden EDI Standards (bzw. einer Untermenge daraus) übernommen. Stattdessen wurden bestehende EDI-Semantiken internet-kompatibel gemacht und in XML gegossen.

Dieser Ansatz ist unter den in dieser Arbeit untersuchten Standards (abgesehen von der aus xCBL abgeleiteten UBL) einzigartig.

Einzigartig erschien zunächst auch die Unterstützung von Auktionen als ein ganz wesentliches Verhandlungsmodell. Allerdings erlauben die zur Verfügung stehenden Dokumententypen lediglich die Erzeugung und Verwaltung einer Auktion in einem externen System. Dabei kann beispielsweise die Auktionsart (DutchForward, Dutch Reverse, English Forward etc.) festgelegt werden.

Der eigentlich interessante Teil, also die Durchführung der Auktion mittels Gebote ist jedoch nicht Gegenstand des Standards.

Die Dokumente werden seit der Version 4.0 über XML Schema definiert und nicht mehr wie früher über DTDs und die proprietären SOX (Commerce One) und XDR (Microsoft) Standards.

Ferner existieren Excel-Sheets, die über Makros einige Subsets von xCBL in EDI (EDIFACT und ANSI X.12) übersetzen und zurückübersetzen können.

4.2.3.5 UBL

Auch die Universal Business Language (UBL) ist ein Standard für die Definition von Geschäftsdokumenten. Dabei wurde nicht von null begonnen, sondern xCBL 3.5 als Startpunkt genommen. Die Entwicklung der beiden Standards beeinflusst sich zwar gegenseitig (so sind Anregungen aus dem UBL-Umfeld in xCBL 4.0 eingeflossen), sind aber unabhängig voneinander. UBL ist unter OASIS (Organization for the Advancement of Structured Information Standards) aufgehängt, die wiederum eine der treibenden Kräfte von ebXML sind und streben an, das UBL "der" Standard für Geschäftsdokumente innerhalb des ebXML-Frameworks wird. Allerdings bleibt es dabei, das prinzipiell jeder beliebige Standard für die Erstellung der Geschäftsdokumente verwendet werden kann. Die Tatsache, das xCBL als Ausgangspunkt gewählt wurde, liegt in der Tatsache begründet, das dieser Standard sich der klassischen EDI-Semantiken bedient und diese vereinfacht ausgedrückt "nur" auf eine XML-Syntax abbildet.

4.2.3.6 BMEcat

Die seit 1998 gegründete BMEcat Initiative ist eine besteht aus einer Gruppe vorwiegend deutscher Unternehmen. Die fachlichen Entwicklungen wurden von Fraunhofer IAO in Stuttgart und den Universitäten Essen und Linz durchgeführt. Ziel war es, ein Katalogaustauschformat für elektronische Beschaffungssysteme zu schaffen.

Die XML-Dateien in BMEcat gestalten sich recht übersichtlich. Eine genaue Spezifikation des Standards findet sich unter (BMEcat, 2003). Das Katalogformat wird mittels einer XML-DTD definiert und neuerdings auch mit XML-Schema. Ein Katalog enthält Pflichtelemente sowie optionale Elemente.

BMEcat unterscheidet drei *Transaktionstypen*:

- Neuer Katalog (T_NEW_CATALOG)
- Aktualisierung von Produktdaten (T_UPDATE_PRODUCTS)
- Aktualisierung von Preisen (T_UPDATE_PRICES)

Ein BMEcat *Dokument* besteht aus zwei Abschnitten, dem Kopfdatenteil (Header) und dem Transaktionsteil.

Der Kopfdatenteil enthält folgende Informationen:

- Informationen über den Katalog (Catalog)
- Informationen über den Käufer (Buyer)
- Informationen über den Lieferanten (Supplier)

Der Transaktionsteil transportiert folgende Informationen:

- Beschreibung der Art der Transaktion
- Informationen über die Kataloggruppen

- Informationen über die Artikel
- Informationen über Abbildung der Artikel auf die Kataloggruppen
- Informationen über Abbildung der Artikel auf ein Klassifikationsschema

Neue Artikel können also mit ebenfalls mittels der Transaktion T_UPDATE_PRODUCTS übermittelt werden. Sie können bei Bedarf auch gleich Kataloggruppen zugeordnet werden, diese müssen aber bereits im System vorhanden sein, eine Übertragung der Gruppen selbst ist nur innerhalb der Transaktion T_NEW_CATALOG möglich, ebenso der Aufbau eines Klassifizierungsschemas (CLASSIFICATION_SYSTEM bzw. FEATURE_SYSTEM für die Merkmalsleisten).

Dem Klassifikationsschema können Merkmalsleisten zugeordnet werden. Darunter sind Attribute eines Produktes zu verstehen, die sich an der Art eines Produktes orientieren. So hat ein Produkt der Klasse „Niedervolthalogenlampe“ andere mögliche Eigenschaften wie Bettwäsche. BMEcat orientiert sich dabei an existierende Klassifikationsschema wie UNSPSC und ecl@ass, die genau dieses Konzept vorsehen. Es wird also kein bestimmtes Klassifikationsschema und darauf definierte Merkmalsleisten vorgesehen sondern die Möglichkeit geschaffen, eines dieser Systeme zu adaptieren.

BMEcat ist wie bereits erwähnt ein reiner Katalogstandard. Er wird durch den openTRANS Standard komplementiert, der zur Zeit (Juni 2003) in der Version 1.0 vorliegt. Er definiert folgende zusätzliche Geschäftsdokumente:

- DISPATCHNOTIFICATION (Lieferavis)
- INVOICE (Rechnung)
- ORDER (Auftrag)
- ORDERCHANGE (Auftragsänderung)
- ORDERRESPONSE (Auftragsbestätigung)
- QUOTATION (Angebot)
- RECEIPTACKNOWLEDGEMENT (Wareneingangsbestätigung)
- RFQ (Angebotsanforderung)

Damit deckt die Kombination BMEcat / openTRANS die Kerndokumente für die elektronische Informationsbeschaffung und Bestellabwicklung ab.

Im Sommer 2002 wurde vom BME (Bundesverband Materialwirtschaft, Einkauf und Logistik) ein Portal bereitgestellt, das zur führenden Plattform für Einkauf und Logistik im deutschsprachigen Raum werden soll (Kropp, 2002). Aufgrund der starken Verbreitung des Katalogformats und der zentralen Position des Bundesverbandes bestehen dafür durchaus realistische Chancen. Hier ist insbesondere zu sehen, welchen starken

Einfluss die Etablierung von Standards für den Datenaustausch auf die ökonomische Situation heutiger Unternehmen ausüben kann.

Verhandlungselemente werden jedoch auch in openTRANS nicht abgebildet. Es wäre beispielsweise denkbar, da die RFQ und QUOTATION entsprechende Prädikate für zu verhandelnde Attribute transportiert – dies ist jedoch nicht der Fall.

4.2.3.7 catXML

CatXML ist ein Katalogaustauschformat, welches im Rahmen einer Initiative des amerikanischen Verteidigungsministeriums geschaffen wurde, einen Marktplatz für die Beschaffung für die einzelnen Unterorganisationen (Luftwaffe, Heer, Marine) zu schaffen.

Das Hauptziel dieser Initiative war es, ein offenes und branchenneutrales Format zu schaffen.

Verhandlungsrelevante Aspekte sind in den verfügbaren Informationen nicht auszumachen.

4.2.3.8 Vergleich der gängigen XML basierten B2B-Standards

In der folgenden Tabelle 4.1 auf Seite 192f werden die gängigen der XML basierten Standards gegenübergestellt.

Tabelle 4.1: Einordnung und Vergleich gängiger B2B-Standards

	ebXML	Biztalk	RosettaNet	CXML	xCBL	BMEcat	catXML	openTRANS	UBL
ALLGEMEINES									
Treibende Kraft	UN/CEFACT, OASIS, alle großen Unternehmen (außer MS)	Microsoft	Diverse	Ariba (initial)	Commerce One	eBSC	U.S. DoD / DLA	eBSC	OASIS
Version	1.0x	2.0	2.0 (?)	1.2	4.0	1.2	1.1	1.0	
Website	www.ebxml.org	www.biztalk.org	www.rosettanet.org	www.cxml.org	www.xcbl.org	www.bmecat.org	www.catxml.org (offline ?)	www.opentrans.org	
FRAMEWORK									
Messaging-Funktionalität	Ja	ja	Ja	-	-	-	-	-	-
XML-Transport	SOAP w/ Attachments	SOAP		-	-	-	-	-	-
Transport-Protokolle	http, smtp, ftp	http, smtp	http, proprietär	-	-	-	-	-	-
Prozesse & Profile	Trading Partner Profile (TPP), Collaboration Partner Profile (CPP)	ja („Orchestration“)	ja, PIP – Partner Interface Processes, es werden Prozesse in sieben Kategorien (Order Management, Inventory Management etc.) definiert	-	-	-	-	-	-
Registry			zwei Dictionaries: technische Eigenschaften, Geschäftsprozesseigenschaften	-	-	-	-	-	-

GESCHÄFTSDOKUMENTE									
welcher Art	-	-	?	Setup Catalog Punchout Orders (create, edit, cancel) Order confirmation Ship notice Invoice	Catalog Order Availability Auction Payment Price Shipping und weitere	Kataloge	Kataloge	Dispatchnotification Invoice Order Orderchange Orderresponse Quotation Receiptacknowledgement Rfq (Request For Quote)	Catalog Order Availability Auction Payment Price Shipping und weitere
Werden Protokolle definiert	-	-	?	unklar -> Recherche	nein	nein, aber durch OpenTRANS	-	-	nein
Definitionsstandard	-	-	?	DTD	XML-Schema frühere Versionen: DTD, XDR, SOX	DTD XML-Schema	DTD	DTD XML-Schema	XML-Schema

[illegible]

4.2.3.9 Verhandlungsfunktionalitäten für das Business Messaging

Derzeit existieren nur wenige und kaum tiefer ausgearbeitete Ansätze, um den Austausch von Verhandlungsnachrichten im Electronic Business zu ermöglichen. Die ebXML-Initiative (ebXML-Initiative, 2003) sieht dies nur in äusserst allgemeiner und unspezifischer Weise vor.

Ein Ansatz von Thun und Rebstock (Thun, et al., 2002; Thun, et al., 2003) schlägt einen einfachen Ablauf vor, in dem einem Initialangebot weitere Gegenangebote folgen können aus denen dann letztendlich ein Vertrag resultiert. Dieser Ablauf wird konform zu ebXML konzipiert. Für die Repräsentation des Verhandlungsdokumentes wird ein Vorschlag unterbreitet.

Während die Einbettung von Verhandlungsnachrichten in den konkreten ebXML Standard ein möglicher Ansatz ist, hängt die erfolgreiche Umsetzung jedoch von der breiten Etablierung des ebXML Standards ab. Es erscheint nützlicher einen allgemeinen Vorschlag zu unterbreiten, der auf alle Standards anwendbar ist und entsprechende Anpassungen durchzuführen, wie es in dieser Arbeit geschieht.

Zusammenfassend lässt sich feststellen, dass bisher keiner der untersuchten Standards wesentliche Unterstützung für Verhandlungen anbietet.

4.3 Modellierung von Verhandlungskommunikation

Die im folgenden vorgestellte Sprechakttheorie bietet gute Voraussetzungen zur Modellierung von Verhandlungskommunikation, da sie einerseits Kommunikationsaktivitäten realitätsnah abbildet und andererseits eine formalisierbare Grundlage anbietet.

4.3.1 Grundlagen der Sprechakttheorie

Die Sprachphilosophie als Teildisziplin der Philosophie beschäftigt sich mit dem Ursprung und Wesen der Sprache. Neben logischen Aspekten werden auch die Bedingungen von Sprache, als Möglichkeit der Philosophie und anderer Wissenschaften Wissen niederzuschreiben, untersucht.

Seit dem 17. Jahrhundert wird Sprache durch das so genannte semantische oder auch semiotische Dreieck terminologisch fixiert: Es werden die Bereiche Syntax (insbesondere die Logik), Semantik (Bedeutungsleh-

re) und Pragmatik (Lehre vom Gebrauch der Zeichen) unterschieden. Von großer Bedeutung in der Sprachphilosophie war die Semantik, welche die Beziehung zwischen Zeichen und Bezeichnetem untersucht.

In jüngerer Zeit gewinnt die Pragmatik einen hohen Stellenwert. Der Umgang mit Zeichen bedeutet immer eine Handlung, d. h. durch die soziale, kommunikative Funktion der Zeichen teilt ein Sprecher durch seine Sprechakte einem Gesprächspartner etwas mit.

Austin und Searle entwickelten hieraus ihre Sprechakttheorie.

4.3.1.1 Sprechen bedeutet handeln

Austin führt als wichtiges Merkmal der Sprechakttheorie die Beobachtung an, dass Sprache nicht nur zur Beschreibung der Welt verwendet wird sondern dass einem Gesprächspartner etwas mitgeteilt oder sogar zu einer Handlung aufgefordert werden soll (Austin, 1962). Ein sprachlicher Ausdruck wird in seiner Bedeutung erst durch seinen Gebrauch definiert.

Wie Austin den Begriff Gebrauch definiert, wird durch folgendes Beispiel deutlich: Person A sagt zu Person B: „Morgen komme ich“. Dieser semantisch und syntaktisch korrekte Satz ist unabhängig von den äußeren Umständen zu verstehen. Jedoch treten Fragen auf, warum der Sprecher diesen Satz äußert und wozu er ihn gebraucht. Es genügt nicht zu wissen, welche Person zu welcher spricht und zu welchem Zeitpunkt dies geschieht. Es ist offen, ob Person A Person B ihr Kommen mitteilt, verspricht, androht, davor warnt oder anderes bezweckt. Austin spricht hierbei von den verschiedenen pragmatischen oder auch performativen Funktionen der Sprechakte und unterscheidet im Wesentlichen drei verschiedene: lokutionären, illokutionären und perlokutionären Akt.

Der *lokutionäre Akt* bezeichnet direkt den Vorgang der Äußerung. Er lässt sich weiter unterteilen in den phonetischen Akt (das Hervorbringen der zum Sprechen notwendigen Geräusche), den phatischen Akt (das Aneinanderreihen von Worten einer Sprache nach deren grammatikalischen Regeln) und den rhetischen Akt (die Verwendung des Produktes des phatischen Aktes mit einer Bedeutung in Bezug auf Objekte und Sachverhalte der Welt).

Der *illokutionäre Akt* legt die Intention einer Äußerung in einer bestimmten Kommunikationssituation fest. Gemeint ist damit das Stellen oder Beantworten einer Frage, das Aussprechen einer Drohung, Warnung oder Information. Der illokutionäre Akt unterteilt sich in zwei Bereiche: der illokutionären Kraft und der Proposition. Die Art der Kommunikationshandlung wird als illokutionäre Rolle oder Kraft bezeichnet. Der Akt nimmt Bezug auf äußere Objekte und Sachverhalte, die Austin Propositionen nennt. Bei einigen Sprechakten kann der illokutionäre Akt nicht direkt aus dem ihn erzeugenden lokutionären Akt abgeleitet werden. Tritt dieser Fall ein, wird von einem indirekten Sprechakt gesprochen. An

folgendem Beispiel kann verdeutlicht werden, wie der illokutionäre Akt nicht mit der eigentlichen Absicht erkannt werden kann: „*Würde es ihnen etwas ausmachen, die Tür zu schließen?*“ Die Äußerung ist wie eine Frage formuliert, jedoch ist der Vollzug des Aktes nicht die Bitte um eine Information sondern eine Aufforderung.

Die Wirkung des lokutionären und illokutionären Aktes beschreibt nun den *perlokutionären Akt*. Will der Sprecher mit einer Äußerung eine andere Person warnen, so ist diese gewarnt. Will er drohen, ist die andere Person eingeschüchtert. Nicht immer stimmt aber der perlokutionäre Akt mit dem des illokutionären überein. So kann ein Sprecher mit einer Information sein Gegenüber nicht überzeugen, oder die Warnung wird nicht ernst genommen. Während der illokutionäre Akt nur von der Äußerung selbst abhängt, hängt das Gelingen des perlokutionären Aktes davon ab, ob in Folge der Äußerung noch etwas anderes geschieht.

Austin hebt die performativen Äußerungen hervor. Sie werden auch *Performative* genannt. Er meint damit Äußerungen, die direkt auf eine bestimmte illokutionäre Rolle zugeschnitten sind. Wird der vollzogene illokutionäre Akt in der Äußerung direkt beschrieben, wird von *expliziten Performativen* gesprochen. Ein Beispiel für einen Befehl wäre die Äußerung „Ich befehle dir, ...“. Eine Warnung sähe folgendermaßen aus: „Ich warne dich, ...“. So kann der Satz: „Ich warne dich, der Hund ist bissig!“ nicht missverstanden werden und ist unabhängig von weiteren Umständen eine Warnung. Es gibt jedoch auch Performative, die von einer eindeutigen Beschreibung der äußeren Umstände abhängig sind. Wird der Beispielsatz ohne die explizite Formulierung „ich warne dich,“ geäußert, so kann dieser Satz ebenso eine Empfehlung sein, diesen Hund als Wachhund einzusetzen. Der Satz wird erst zur Warnung, wenn andere äußere Umstände, wie etwa deutliche Gesten oder ein besorgter Ton des Sprechers, dazukommen. Sind diese gegeben, handelt es sich um eine *primär performative* Äußerung. In beiden Fällen handelt es sich aber um Performative, denn durch sie wird ein illokutionärer Akt vollzogen.

Austin unterscheidet bei den Äußerungen vier Freiheitsgrade: (1) die Dimension von Glücken und Verunglücken¹, (1a) die illokutionäre Rolle, (2) die Dimension von Wahrheit und Falschheit und (2a) die lokutionäre Bedeutung.

Um Sprechakte näher klassifizieren zu können, greift er auf die illokutionäre Rolle zurück, die ihm am meisten als geeignet erscheint. Er unterscheidet fünf Klassen: die *verdiktive*, *exerzitive*, *kommissive*, *konduktive* und *expositive* Klasse. Verdiktive Äußerungen urteilen, schätzen oder bewerten, wobei das Urteil keinen endgültigen Charakter haben muss. Mit exerzitiven Äußerungen wird Macht oder Einfluss ausgeübt, sie drängen, warnen oder weisen an. Kommissive Äußerungen verpflichten den Sprecher zu etwas und legen ihn auf Handlungen fest. Konduktive

Äußerungen betreffen Einstellung und soziales Verhalten in der Gesellschaft, z.B. Entschuldigungen, Empfehlungen oder Herausforderungen. Expositive Äußerungen schließlich verdeutlichen ihre Stellung innerhalb einer Konversation, so wie Voraussetzen, Antworten oder Einräumen.

4.3.1.2 Taxonomie der Sprechakte

Austins Schüler Searle baut auf den geschaffenen Grundlagen auf und erweitert und verfeinert die Sprechakttheorie. Sein Anliegen ist es, die Theorie allgemeingültiger zu entwickeln, so dass sie unabhängig von einzelnen Sprachen ist. Er entwickelt ein fundiertes Ordnungsschema, dass eine Klassifizierung von Sprechakten ermöglicht. In (Searle, 1969) veröffentlicht er zunächst seine Ergebnisse, verbessert diese aber weiter und entwickelt in (Searle, 1979) eine Analyse, die im folgenden kurz beschrieben werden soll.

Searle betrachtet nicht nur den illokutionären Akt, sondern auch die dabei verwendeten Verben. Ein wichtiger Kritikpunkt an Austins Ausführungen ist, dass er nicht unterscheidet zwischen illokutionären Akten und illokutionären Verben. Er „(...) scheint anzunehmen, eine Klassifikation verschiedener Verben sei eo ipso eine Klassifikation von Arten illokutionärer Akte“ (Searle, 1979). Dabei entsteht das Problem, dass eine Betrachtung von Sprechakten jeweils nur in Bezug auf eine bestimmte Sprache geschehen kann. Außerdem wird eine Klassifikation erschwert. Nach Austin beschreiben zwei Verben mit unterschiedlicher Bedeutung stets verschiedene illokutionäre Akte. Dies muss aber nicht immer zutreffen. Derselbe illokutionäre Akt kann unter Umständen durch Äußerungen mit verschiedenen Verben vollzogen werden, welche ihrerseits beispielsweise die Art und Weise des Vollzugs kennzeichnen.

Aus diesem Sachverhalt und aus einigen anderen Punkten resultiert die entscheidende Kritik an der Taxonomie Austins, der ein klares und durchgängiges Prinzip fehlt. Austin räumt selbst ein, dass dies zu Überschneidungen zwischen den Kategorien und zu sehr heterogenen Inhalten einzelner Kategorien führt.

Searle entwickelt insgesamt zwölf verschiedene Merkmale, in denen sich Sprechakte unterscheiden können. Die drei wichtigsten verwendet er zur Klassifikation der Akte: den *illokutionären Witz*, die *Ausrichtung* und die *Aufrichtigkeitsbedingung*.

Der *illokutionäre Witz* (engl.: illocutionary point) bezeichnet den Zweck eines Illokutionstyps. Der Witz eines Befehls ist es, den Hörer dahingehend zu beeinflussen, dass er etwas Bestimmtes macht. Der Zweck eines Versprechens ist die Verpflichtung des Sprechers zu einer bestimmten Handlung. Von großer Bedeutung ist dabei, dass der illokutionäre Witz nicht mit dem perlokutiven Ziel gleichgesetzt wird, da jenes nicht für jeden Akt existiert. Auch wenn der illokutionäre Witz lediglich

einen Teil der illokutionären Rolle darstellt, so ist er doch der wichtigste. So haben beispielsweise eine Bitte und ein Befehl denselben illokutionären Witz, nämlich den Hörer zu einer Handlung zu bewegen. Sie unterscheiden sich aber deutlich in ihrer illokutionären Rolle.

Die *Ausrichtung* eines illokutionären Aktes beschreibt sein Verhältnis zur Welt und hängt mit dem illokutionären Witz zusammen. Es wird unterschieden, ob mit einem Akt versucht wird, die Wörter der Welt oder die Welt den Wörtern anzupassen. Zur ersten Ausrichtung gehören Feststellungen, zur zweiten Befehle und Versprechen. Searle zeigt diesen Unterschied an Hand zweier Menschen: Person A kauft ein, Person B beobachtet dies. Person A nimmt durch einen Einkaufszettel gelenkt nacheinander Waren aus den Regalen. Person B notiert sich die Reihenfolge. Wenn der Einkauf beendet ist, wird der Inhalt beider Listen identisch sein. Der Unterschied liegt aber in ihrer Funktion. Person A orientiert sich mit seinen Handlungen an der Liste, versucht also die Welt den Wörtern anzupassen (Welt-auf-Wort). Person B hat seine Liste gemäß den Handlungen der Person A niedergeschrieben, hat also die Wörter der Welt angepasst (Wort-auf-Welt). Der Unterschied kann auch verdeutlicht werden, wenn mögliche Fehler in der Liste betrachtet werden: Person A muss erneut das Geschäft aufsuchen, um die Ware zu korrigieren. Person B reicht es, den falschen Begriff durchzustreichen und den neuen daneben zu notieren.

Die *Aufrichtigkeitsbedingung* eines Aktes mit propositionalem Gehalt bezeichnet die Einstellung des Sprechers zu diesem Inhalt. Dabei wird sein psychischer Zustand zum Ausdruck gebracht. Die Handlungsabsicht wird durch ein Versprechen oder eine Drohung ausgedrückt. Durch eine Bitte oder einen Befehl wird der Wunsch zum Ausdruck gebracht, dass etwas geschehen möge. Selbst wenn der Sprecher nicht aufrichtig ist, bringt er seine Einstellung zum Ausdruck.

Die restlichen Unterscheidungsmerkmale für Sprechakte sind:

- die Stärke des illokutionären Witzes
- die Stellung der Beteiligten zueinander
- die Interessen der Beteiligten
- der Bezug zum restlichen Diskurs
- der propositionale Gehalt unter Berücksichtigung seiner Abhängigkeit von der illokutionären Rolle
- die Möglichkeit, einen Akt nicht als Sprechakt zu vollziehen
- die Abhängigkeit von außersprachlichen Einrichtungen
- das Vorhandensein eines entsprechenden Verbs mit performativem Charakter
- der Stil, in dem der Akt vollzogen wird

Searle baut auf diesen ersten drei Dimensionen seine Taxonomie auf und entwickelt daraus fünf verschiedene Klassen, in die er die Sprechakte

einteilt: *Assertive*, *Direktive*, *Kommissive*, *Expressive* und *Deklarationen*. Im Folgenden steht S für den Sprecher, H für den Hörer und p für den propositionalen Gehalt.

Der illokutionäre Witz der *Assertiven* ist, den Sprecher darauf festzulegen, dass der propositionale Gehalt der Äußerung wahr ist. Ihre Ausrichtung ist Wort-auf-Welt und der zum Ausdruck gebrachte psychische Zustand, also die Aufrichtigkeitsbedingung „glauben, dass p“.

Bei *Direktiven* besteht der illokutionäre Witz darin, dass sie Versuche des Sprechers sind, den Hörer dazu zu bewegen, das zu tun, was der propositionale Gehalt beschreibt, und der Gehalt ist immer, dass H eine künftige Handlung h vollzieht. Die Ausrichtung ist Welt-auf-Wort und die Aufrichtigkeitsbedingung „wollen, dass H macht h“. Fragen bilden eine Unterklasse der Direktiven: In ihrem Fall ist h „H gibt eine Antwort“.

Die Definition der *Kommissiven* entspricht der Austins. Ihr illokutionärer Witz ist, den Sprecher auf ein bestimmtes Verhalten festzulegen und ihre Ausrichtung Welt-auf-Wort. Ähnlich wie bei den Direktiven besteht der propositionale Gehalt immer darin, dass S eine künftige Handlung h vollzieht. Die Aufrichtigkeitsbedingung ist „beabsichtigen, dass S tut h“.

Der illokutionäre Witz der *Expressiven* ist, den psychischen Zustand zum Ausdruck zu bringen, der durch die Aufrichtigkeitsbedingung in Bezug auf eine im propositionalen Gehalt genannte Sachlage spezifiziert ist. Die Expressiven haben keine Ausrichtung, die Wahrheit der Proposition wird bei dieser Klasse vorausgesetzt, und ihre Aufrichtigkeitsbedingung ist variabel.

Bei den *Deklarationen* besteht der illokutionäre Witz darin, allein durch die erfolgreiche Vollbringung des Aktes eine Übereinstimmung zwischen propositionalem Gehalt und der Realität herbeizuführen. Die Ausrichtung der Elemente dieser Klasse ist somit bidirektional, also sowohl Wort-auf-Welt als auch Welt-auf-Wort. Eine Aufrichtigkeitsbedingung gibt es nicht aber es muss eine außersprachliche Einrichtung geben, die den Vollzug des Aktes ermöglicht. Die *assertiven Deklarationen* werden als spezielle Unterklasse betrachtet und zeichnen sich dadurch aus, dass sie – genau wie die *Assertiven* – eine Aufrichtigkeitsbedingung haben. .

4.3.2 Konstitutive Regeln für Sprechakte

Austin beobachtete, dass beim Vollbringen illokutionärer Akte Fehler gemacht werden können. Sie können nach sich ziehen, dass ein Akt nicht gut oder überhaupt nicht vollzogen wird. An Hand von Akten, die durch explizit performative Äußerungen vollbracht werden, untersucht er dieses und schematisiert die Ergebnisse in Form von sechs Regeln, die mindes-

tens erfüllt sein müssen, damit eine performative Äußerung wie erwartet gelingt.

A.1 Die Äußerung muss sich auf ein per Konvention existierendes Verfahren berufen, das zu einem bestimmten Ergebnis führt.

A.2 Die äußeren Umstände müssen den Voraussetzungen für das Verfahren genügen.

B.1 Das Verfahren muss korrekt eingehalten werden.

B.2 Das Verfahren muss vollständig durchgeführt werden.

Ä.1 Die Beteiligten müssen sich ehrlich äußern, d.h. ihre Äußerungen müssen ihren Handlungen und Überzeugungen entsprechen.

Ä.2 Die Beteiligten müssen sich konsequent verhalten, d.h. ihre Handlungen müssen ihren Äußerungen entsprechen.

Der illokutionäre Akt überhaupt wird nicht vollzogen und die entsprechende Handlung findet nicht statt, wenn gegen eine der A- oder B-Regeln verstoßen wird. Ein Verstoß gegen die Ä-Regeln lässt den Akt zwar zustande kommen. Er ist aber weniger gelungen.

Dieser Aspekt der Sprechakttheorie wird von Searle weiter vertieft und systematisiert. Er präsentiert eine Reihe konstitutiver Regeln für den Vollzug illokutionärer Akte. Der Vollzug wird in den meisten Fällen erst durch das Vorhandensein von Regeln ermöglicht. Damit vergleichbar sind Spiele wie Eishockey oder Mensch-ärgere-dich-nicht, die durch ihre Regeln definiert sind.

Die *normalen Ein- und Ausgabebedingungen* betreffen sinnvolles Sprechen und Verstehen, also z.B. eine gemeinsame Sprache, die physischen Voraussetzungen, diese zu verwenden und das Wissen um damit verbundene Konventionen.

Der Inhalt einer Äußerung unterliegt den *Bedingungen des propositionalen Gehalts*, d.h. bei einem Versprechen muss eine zukünftige Handlung und bei einem Bericht müssen Umstände der Gegenwart oder Vergangenheit beschrieben werden.

Die *Einleitungsbedingungen* beschreiben Eigenschaften der Welt, die vorhanden oder erfüllt sein müssen, damit ein Akt sinnvoll angewendet werden kann.

Die *Bedingung der Aufrichtigkeit* verlangt eine Übereinstimmung zwischen den zum Ausdruck gebrachten Überzeugungen und Absichten des Sprechers und seinen tatsächlichen Eigenschaften.

Der *wesentlichen Bedingung* zu Folge muss die Handlung, die der Sprecher durch seine Äußerung vollzieht, auch von ihm beabsichtigt sein.

Schließlich muss es noch in der Absicht des Sprechers liegen, einen bestimmten illokutionären Effekt dadurch zu erzielen, den Zuhörer zur

Erkenntnis zu bringen, dass er eben jenes beabsichtigt. Diese Bedingung basiert auf dem von Grice (Grice, 1957) geprägten Begriff der „nicht-natürlichen Bedeutung“.

Für den erfolgreichen und vollständigen Vollzug eines Sprechaktes sind die aufgeführten Bedingungen sowohl notwendig als auch hinreichend. Jedoch führt die Verletzung einer Bedingung nicht in jedem Fall dazu, dass der gesamte Akt ungültig ist. Searle bezeichnet einen Akt, bei dem nicht alle Bedingungen erfüllt sind, als unvollständig.

Habermas betrachtete in seiner Theorie des kommunikativen Handelns (Habermas, 1981) systematisch die Voraussetzungen, unter denen ein Sprechakt glücken kann. Sein Ausgangspunkt dabei ist, dass ein Sprecher durch jeden Sprechakt drei Geltungsansprüche erhebt: den Anspruch der Wahrheit, der Richtigkeit und der Wahrhaftigkeit. Ursprünglich rechnet Habermas den Anspruch der Verständlichkeit hinzu. Diesen betrachtet er später aber als minimale Grundlage jeder Form der Kommunikation und nicht mehr speziell als Geltungsanspruch.

Die drei Geltungsansprüche beziehen sich auf drei unterschiedliche Welten. Der Anspruch der Wahrheit bezieht sich auf die objektive Welt, auf alle Dinge und Sachverhalte, zu denen überprüfbare Aussagen gemacht werden können. Der Richtigkeitsanspruch betrifft die soziale Welt, die legitime Ordnung der geregelten interpersonalen Beziehungen. Die subjektive Welt ist die „innere“ Welt des Sprechers, sie umfasst seine subjektiven Einstellungen und Erlebnisse und ist Gegenstand des Wahrhaftigkeitsanspruchs.

Der Zuhörer kann für jeden Geltungsanspruch entscheiden, ob er diesem zustimmt. Ein Sprechakt ist nur dann erfolgreich, wenn Einigkeit in Bezug auf alle Ansprüche besteht. Widerspricht der Zuhörer einem Anspruch, so ist der Sprecher gezwungen, diesen weiter zu begründen. Im Verlauf dieser Begründung kommt es zu einer Diskussion zwischen beiden Parteien, die in dem Moment beendet ist, wenn eine endgültige Einigkeit oder Uneinigkeit erreicht ist.

Zwar sind die aktuelleren Arbeiten von Habermas für die Sprechakttheorie ebenfalls von Bedeutung, ihre Zielrichtung ist jedoch stärker gesellschaftsphilosophisch angelegt. Habermas verfolgt einen utopischen Entwurf einer idealen Kommunikationsgemeinschaft. Er konzentriert sich dabei auf eine Sachverhaltsdarstellung und klammert Willen und Handlungsziele aus. So haben seine Arbeiten in dem hier relevanten Kontext nur geringe Bedeutung und finden daher keine weitere Beachtung.

4.3.3 Modellierungen auf Basis der Sprechakttheorie

Die Anwendungen der Sprechakttheorie in der Informatik werden im Bereich der ‚Language Action Perspective‘ (LAP) zusammengeführt (Schoop, 2000).

Es folgt ein kurzer Überblick zu den relevanten Anwendungen im Verhandlungs- und B2B-Bereich.

Eine wichtige Anwendung der Sprechaktkommunikation findet sich in der Agentenkommunikation, insbesondere bei der FIPA (FIPA, 2002a) (Poslad, et al., 2001).

Die für den Bereich der Agententechnologie entworfenen Lösungen sind jedoch nicht adäquat für die Verhandlungskommunikation im B2B-Bereich. Daher werden in der Folge entsprechend angepasste Lösungen entwickelt.

Eine früher Ansatz zum Business Messaging, neben den in Abschnitt 4.2.2 erwähnten, wurde von Scott A. Moore und Steven O. Kimbrough mit der ‚Formal Language for Business Communication‘ (FLBC) entwickelt. Er setzt im Wesentlichen auf KQML auf, bietet jedoch keine Integration in bestehende B2B-Standards und Umgebungen. Der Ansatz hat keine implizite Verbindung zum Bereich der Verhandlungen.

Traum (Traum, 2000) diskutiert allgemein Kriterien für den Entwurf von Taxonomien für „Dialogue Acts“ und damit auch für Anforderungen an eine Taxonomie von Sprechakten für Verhandlungen. Traum gibt an, das es sinnvoll sein kann, sprechaktbasierte Kommunikationssysteme sogar während Mensch-zu-Mensch Kommunikation bei computerunterstützter Kollaboration einzusetzen.

Dabei ist es wichtig Flexibilität in den Ausdrucksmitteln zu wahren. Diese Anforderung machen Bichler et al. (Bichler, et al., 2002) bei der flexiblen Formulierung von Nachfragen und Angeboten deutlich, insbesondere bei der dynamischen Preis- und Konditionenfindung.

Chang et al. (Chang, et al., 1994) stellen ein Verhandlungsprotokoll auf Basis von Sprechakten vor, das ‚Speech Act based Negotiation Protocol‘ (SANP). Ihre Arbeit basiert auf der Klassifikation von Sprechakten durch Ballmer et al. (Ballmer, et al., 1981). Hier fehlt jedoch auch wieder eine Integration in den B2B-Bereich.

Negoisst (Schoop, et al., 2002) ist ein Verhandlungsunterstützungssystem, bei dem die natürlichsprachlichen Verhandlungsnachrichten in Ausschnitten als explizite Sprechakte gekennzeichnet werden können.

Es fehlt eine tiefgreifende Integration zwischen der Verhandlungsschematik und der Darstellung durch Sprechakte.

5 Systemtechnische Infrastrukturen zur Durchführung von Verhandlungsinteraktionen

In derzeitigen E-Commerce Systemen fehlt es weitgehend noch an einer geeigneten systemtechnischen Infrastruktur zur Unterstützung von elektronischen Verhandlungen wie in den vorhergehenden Kapiteln deutlich wurde.

Eine interoperable Verhandlungsinfrastruktur ist jedoch Voraussetzung für eine effektive Unterstützung der Vereinbarungsphase im Electronic Business. Durch eine solche Verhandlungsinfrastruktur werden effiziente teilautomatisierte Lösungen erst ermöglicht.

Die sprechaktbasierte Kommunikation bietet innovative und für eine Verhandlungssprache adäquate Grundlagen. Die Sprechakttheorie hat auch in der Agententechnologie Verwendung gefunden; die dortigen Lösungen sind jedoch in der Umgebung des Electronic Business nicht praktikabel umsetzbar, da die domänenspezifischen Anforderungen unberücksichtigt bleiben. So werden als Infrastrukturelemente eine Ontologie des oben entwickelten Basismodells für Verhandlungen und eine B2B-Ontologie im Kontext der verschiedenen B2B-Standards entworfen, umgesetzt und in B2B-Standards integriert und deren Vermittlung untereinander garantiert. Somit können in der Folge E-Business-Applikationen auf die durch die Infrastruktur angebotenen Funktionalitäten zurückgreifen.

Eine erste und besondere Anwendung findet die Verhandlungsinfrastruktur bei der Unterstützung verhandelbarer Verhandlungsprotokolle in Form von Protokollselektion und dynamischem Bootstrapping, die jedoch gleichzeitig noch Teil der neuen systemtechnischen Infrastruktur sind.

Die im Folgenden entworfenen Ontologien und vorgestellten systemtechnischen Komponenten sind als Teil einer prototypischen systemtechnischen Infrastruktur zu sehen, die unter der Bezeichnung ‚Electronic Procurement Negotiations System‘ (EPN-System) später in Kapitel 6 umgesetzt wird.

5.1 Ontologien als Infrastrukturen

Zunächst wird ein kurzer Einblick in das Thema Ontologien gegeben. Ontologien und die Systeme zu ihrer Handhabung können als gemeinsame Infrastruktur in verteilten offenen Systemen dienen.

Im Anschluss an die Einführung zum Thema Ontologien folgt die Betrachtung von Entwurfsmethodologien für Ontologien.

5.1.1 Grundlagen von Ontologien

Der aus dem Griechischen stammende Begriff *Ontologie* bedeutet übersetzt „die Lehre vom Sein“. Er bezeichnet ursprünglich eine philosophische Grunddisziplin der allgemeinen Metaphysik, die „nach der erkenntnistheoretischen und metaphysischen Struktur der Wirklichkeit, nach den formalen Gesetzen unseres Realitätsverständnisses und nach der relativen Gültigkeit des Seins in Ansehung seiner Abhängigkeit vom erkennenden Subjekt“ fragt (Brockhaus, 1996-99). Speziell die materiale Ontologie beschäftigt sich mit dem organischen Aufbau der Welt, die es allgemein zu klassifizieren gilt. Sie ordnet deren Elemente anhand ihrer Eigenschaften nach Einzelercheinungen, Gruppen, Arten und Gattungen und ist klassischer Weise bemüht, ein Stufen- oder Schichtenmodell zu konstruieren, das das Existierende als systematisch angeordnetes Ganzes beschreibt.

In vielen Bereichen der Informatik ist es notwendig, Erkanntes oder Erdachtes zu repräsentieren, betreffend Fakten, Regeln, Zusammenhänge oder Sachverhalte in unterschiedlichen Anwendungsbereichen (Geschäftsprozesse, Dokumentinhalte oder technische Anwendungen um nur einige zu nennen). So gelangt der Begriff über die Bereiche der künstlichen Intelligenz und der wissensbasierten Systeme in die Informatik, vor deren Hintergrund er neu interpretiert wird. Aus Sicht eines wissensbasierten Systems ist das, was existiert, genau das, was beschrieben werden kann. Formal repräsentiertes Wissen eines solchen Systems basiert auf einer Konzeptualisierung, einer abstrakten und vereinfachten Sicht auf die Welt, die beschrieben werden soll, bestehend aus Objekten, Konzepten und anderen Entitäten sowie den Beziehungen zwischen ihnen. In diesem Kontext versteht man unter einer Ontologie die explizite Spezifikation einer solchen Konzeptualisierung (Genesereth, et al., 1987; Gruber, 1993).

In der ursprünglichen Form ist der Plural des Wortes *Ontologie* nicht vorhanden. Er wird erst durch die Verwendung des Begriffes in der Informatik eingeführt und erhält hier einen Sinn.

Da nicht nur jedem wissensbasierten System, sondern generell jedem System oder Programm explizit oder implizit eine Konzeptualisierung zugrunde liegt — mag sie auch noch so primitiv oder vage sein — bleibt die Anwendung der im folgenden verdeutlichten Konzepte nicht auf den Bereich der künstlichen Intelligenz beschränkt.

Definition 1: Eine Ontologie ist eine explizite Spezifikation einer Konzeptualisierung, einer abstrakten und vereinfachten Weltsicht, die die Entitäten eines bestimmten Bereichs und deren Relationen beschreibt.

Alle Objekte und die Relationen zwischen ihnen, die sich auf Basis einer Ontologie beschreiben lassen, finden ihre Entsprechung im Vokabular der selben in Form von Bezeichnern. Dies ermöglicht die Beschreibung einer Ontologie durch die Definition einer Menge von Termen. Die Definitionen ordnen den Bezeichnern der Entitäten umgangssprachliche Texte zu, die die Bedeutung dieser Namen beschreiben, und sie bilden Axiome, die die Interpretation und wohlgeformte Anwendung solcher Terme eingrenzen. Dazu können beispielsweise Definitions- und Wertebereiche, Einordnung in Hierarchien, Vererbung von Klasseneigenschaften an Instanzen und Ähnliches gehören (Gruber, 1991).

Definition 2: Das Vokabular einer Ontologie ist die Menge aller Bezeichner, die den Objekten und Relationen zugeordnet sind, die sich auf Basis der Ontologie beschreiben lassen.

Die Beschreibung einer Ontologie erfolgt unabhängig von der internen Wissensrepräsentation und Struktur konkreter Systeme. Für Systeme, Programme und auch menschliche Nutzer, im folgenden als handelnde Einheiten betrachtet und als Agenten bezeichnet, sind Ontologien ein Mittel, um sich über die Bedeutungen von Objekten und Relationen zu einigen, auf deren Basis sie interagieren und kommunizieren. Dabei ist es weder notwendig, daß die beteiligten Agenten ihre internen Weltmodelle auf die selbe Weise repräsentieren, noch daß sich diese in Umfang und Mächtigkeit entsprechen. Je nach Art und Umfang der Interaktion genügt eine Verpflichtung der Beteiligten zu einer begrenzten gemeinsamen Ontologie. Damit ein Agent eine solche ontologische Verpflichtung erfüllt, müssen lediglich seine sichtbaren Handlungen mit den Spezifikationen der Ontologie logisch übereinstimmen.

Diese Definitionsweise basiert auf der von Newell entworfenen „knowledge-level perspective“ (Newell, 1982).

Umgekehrt bedeutet dies, dass eine Ontologie die Funktionsweise eines Agenten nicht vollständig spezifizieren muss. Die Begrenzung der Ontologie besteht auch darin, dass i.d.R. nur einige der formalen Beschränkungen definiert werden, die für die Objekte gelten, die Gegenstand der Kommunikation sind. Die ontologische Verpflichtung eines

Agenten betrifft also nur die Folgerichtigkeit und Übereinstimmung, nicht aber die Vollständigkeit seiner Handlungen. Durch die unabhängige, explizite Beschreibung inhaltlicher Konzepte und Zusammenhänge stellt eine gemeinsame Ontologie somit eine semantische Kommunikationschnittstelle dar. So wie eine Methode durch ihre Signatur, ihre Liste von Argumenten, ihre interne Arbeitsweise kapselt und nach außen verbirgt, gestattet eine gemeinsame Ontologie die Interaktion mit einem Agenten, ohne sich dessen interne Wissensrepräsentation zu eigen machen zu müssen.

Definition 3: Eine gemeinsame Ontologie beschreibt die ontologische Verpflichtung mehrerer Agenten zu einer begrenzten gemeinsamen Welt-sicht. Dieser müssen die sichtbaren Handlungen der verpflichteten Agenten entsprechen.

Bereits für den Entwurf isoliert arbeitender Programme kann die Strukturierung von Teilen ihrer Semantik in Form von Ontologien vorteilhaft sein, obwohl hier u.U. der dazu nötige Aufwand in einem ungünstigem Verhältnis zum erzielten Nutzen steht. Für die Entwicklung komplexer, heterogener, verteilter oder offener Systeme, deren Komponenten untereinander oder gar mit menschlichen Benutzern auf einem höheren Abstraktionsniveau kommunizieren müssen, spielt diese Methode eine sehr wichtige Rolle. Ein herausragendes Anwendungsbeispiel hierfür ist die sprechaktbasierte Kommunikation zwischen autonomen Software-Agenten. Solch Kommunikation auf Wissensebene lässt sich in mehrere Bereiche untergliedern. In (Gruber, 1993) werden derer drei identifiziert die getrennt voneinander spezifiziert werden können: das Format der Beschreibungssprache (z.B. KIF), das Protokoll für die Kommunikation zwischen Agenten (z.B. KQML) und die inhaltliche Spezifikation des gemeinsamen Wissens. Für letztere bieten sich Ontologien als Mittel zur Beschreibung an, wodurch ihre Bedeutung für die Kommunikation umrissen wird und sie in Relation zu deren anderen Komponenten gestellt werden. Diese Unterteilung korrespondiert zu der in den Spezifikationen der FIPA vorgeschlagenen, und unter Verwendung der dort gebrauchten Terminologie ergeben sich folgende Bestandteile der Kommunikation:

- Die *Kommunikationssprache* definiert das Protokoll, dem jede Kommunikation zwischen Agenten folgen muss. Ein Beispiel hierfür ist die FIPA-ACL (FIPA, 2002b; FIPA, 2002d).
- Die *Inhaltssprache* ist eine formale Sprache, in der der eigentliche Inhalt der zwischen Agenten ausgetauschten Nachrichten verfasst ist. Beispiele hierfür sind FIPA-SL (FIPA, 2002f) und KIF (Genesereth, et al., 1992).

- Die *Ontologie* schließlich beschreibt Konzepte und Zusammenhänge der gemeinsamen Weltsicht der Agenten. Soweit wie möglich geschieht dies unabhängig von Kommunikations- und Inhaltssprache.

In der Praxis werden die meisten Ontologien derzeit auf Basis eines frame-basierten Modells beschrieben. Daher stehen Klassen im Fokus des Entwurfs und der Definitionen. Sie beschreiben die Konzepte einer Domäne und der Begriff 'Klasse' findet sich oft als Synonym für 'Konzept'. Slots (auch 'properties', 'roles' oder 'parameters') beschreiben die Eigenschaften einer Klasse, Facetten (auch 'role restrictions') wiederum die Beschränkungen, denen die einzelnen Slots unterliegen und das Verhältnis zwischen Ober- und Unterklassen charakterisiert den Unterschied zwischen allgemeinerer und speziellerer Ausprägung eines Konzepts.

Im Kontext der wichtigsten Beschreibungsformen für Ontologien sind die enge Beziehung zwischen Logik und Ontologien zu nennen, Frames und konkrete Notationen wie KIF, RDF, RDF-Schema, DAML und DAML+OIL.

Neben der Kommunikation gibt es eine Reihe weiterer Anwendungsfelder und Bereiche, in denen Ontologien als nützliche Werkzeuge dienen können. Dazu gehören u.a. Systementwurf, Repräsentation und Wiederverwendung von Wissen, automatisches Schließen und Interoperabilität (Gruninger, et al., 2002; Uschold, et al., 1996)

5.1.2 Entwurf von Ontologien

Als wichtige Kriterien für Ontologien angesehen werden Klarheit, Kohärenz, Erweiterbarkeit, Unabhängigkeit, minimale Verpflichtung und Konsistenz mit Bestehendem (Uschold, et al., 1996; Uschold, et al., 1995).

Trotz der Bedeutung, die Ontologien für verschiedene Bereiche der Informatik zunehmend gewinnen, gibt es bisher keine standardisierte Methode zu ihrem Entwurf. Immerhin wurden bereits Versuche unternommen, vorhandene Erfahrungswerte aus diesem Bereich in Form von Richtlinien und Hilfestellungen zu sammeln, um den Entwurf neuer Ontologien zu erleichtern.

Uschold et al. geben (Uschold, et al., 1996; Uschold, et al., 1995) allgemeine Hinweise bezüglich einer geeigneten Vorgehensweise und ein möglicher Entwurfsvorgang wird in einzelne Aufgaben und Unteraufgaben zerlegt. Die oben genannten Kriterien für die Beurteilung von Ontologien dienen in jedem Schritt als Maßstab und Entscheidungshilfe.

Der erste Schritt besteht darin, Zweck und Umfang der zu entwickelnden Ontologie zu bestimmen. Ihre Anwendungsziele müssen bestimmt und im Raum der möglichen Anwendungsfelder für Ontologien positioniert werden.

Darauf folgt die Konstruktion der Ontologie, die sich wiederum in die Unterschritte *einfangen*, *codieren* und *integrieren* untergliedern läßt. Einfangen einer Ontologie bedeutet, ihre Schlüsselkonzepte und -zusammenhänge zu identifizieren. Zu diesen werden präzise, eindeutige textuelle Beschreibungen verfaßt so daß anschließend einzelne Terme isoliert und definiert werden können, die als Bezeichner für die Konzepte und Zusammenhänge dienen. Hierzu erfolgt zunächst ein 'brainstorming', durch das potentiell relevante Begriffe gesammelt werden, die die ihnen zugrundeliegenden Konzepte repräsentieren und so vorläufig noch mögliche Mehrdeutigkeiten und Widersprüche verbergen. Die gewonnenen Begriffe werden lose strukturiert, in dem sie unterschiedlichen Arbeitsbereichen zugeordnet werden und so in sich aus den Bereichen in natürlicher Weise ergebende Gruppen unterteilt werden. Anschließend werden semantische Querbeziehungen zwischen den Bereichen identifiziert, also Konzepte die sich voraussichtlich auf Konzepte in anderen Gruppen beziehen oder durch solche referenziert werden. Hieraus ergeben sich Anhaltspunkte für die Reihenfolge des weiteren Vorgehens: Uschold und Gruninger (Uschold, et al., 1996) empfehlen, Bereiche mit starker semantischer Überlappung zu erst zu behandeln. Diese müssen sauber umgesetzt werden, da sich sonst in allen anderen Bereichen Komplikationen und zusätzlicher Arbeitsaufwand ergeben.

Zum *Codieren* gehört die Festlegung auf grundlegende Begriffe (z.B. Klasse, Objekt, Relation, usw.), die als Basis für die Spezifikation der Ontologie dienen sollen, und die eine Metaontologie bilden. Es muß eine geeignete Repräsentationssprache gewählt werden, die die Konzepte der Metaontologie unterstützt, damit anschließend die textuelle Beschreibung der Konzepte in diese Sprache überführt bzw. durch Axiome in dieser ergänzt werden kann. Im Verlauf beider Phasen muß u.U. die Frage geklärt werden, ob es bestehende Ontologien mit ähnlichen Konzepten gibt, und ob und wie diese in die neue Ontologien *integriert* werden können und sollen.

Für den Codierungsschritt empfiehlt sich eine *middle-out*-Vorgehensweise an Stelle der klassischen top-down- und bottom-up-Ansätze. Das bedeutet, in jedem Arbeitsbereich zunächst die zentralen Begriffe zu definieren, bevor die abstrakteren oder spezielleren behandelt werden. Der bottom-up-Ansatz führt i.d.R. zu einer großen Detailfülle, die es erschwert, Gemeinsamkeiten zwischen verwandten Konzepten aufzufinden, was wiederum zu inkonsistenten Spezifikationen führen kann.

Diese müßten dann überarbeitet werden und der in jedem Fall mit diesem Ansatz verbundene hohe Arbeitsaufwand wächst zusätzlich. Bei einer top-down-Vorgehensweise hingegen lässt sich die Detailfülle besser kontrollieren. Sie kann jedoch dazu führen, dass auf oberster Ebene künstliche und willkürliche Kategorien eingeführt werden, die die Stabilität des gesamten Entwurfs gefährden und dann mit zusätzlichem Aufwand neu geordnet werden müssten. Der middle-out-Ansatz aber führt zu einem Kompromiss: Die Detailfülle ist kontrollierbar, da zusätzliche Details nur bei Bedarf durch Spezialisierung der grundlegenden Konzepte hinzukommen. Übergeordnete Kategorien ergeben sich in natürlicher Weise, da sie mit Hilfe der zentralen Begriffe definiert und beschrieben werden können.

Im Anschluß an die Konstruktionsphase sollte eine Evaluierung der Ontologie erfolgen, bei der die ursprünglich festgelegte Zielsetzung bezüglich des Einsatzbereiches, der angestrebte Umfang der Ontologie und ggf. auch ein praktischer Einsatz als Maßstäbe dienen können. Der letzte Entwurfsschritt ist die Dokumentation der Ontologie, die deren Wiederverwendbarkeit verbessert.

5.2 Verhandlungsontologie

Die vorgestellten Entwurfsmethoden für Ontologien werden zunächst auf den Projektkontext angepasst und danach wird die so entwickelte Verhandlungsontologie, die ein wesentliches Element von Verhandlungsaktivitäten und Verhandlungssprachen ist, in wichtigen Ausschnitten dargestellt.

5.2.1 Anpassung der Entwurfsmethoden auf die Verhandlungsontologie

Auch wenn die bisher beschriebenen Methoden hilfreiche, nützliche Richtlinien für den Entwurf von Ontologien liefern, sind nicht alle Aspekte unmittelbar auf den Entwurf der allgemeinen Verhandlungsontologie übertragbar. Vielmehr erfordern die speziellen Umstände eine Adaption, die aber nicht im Widerspruch zu den vorgestellten Konzepten steht, da diese sich als Arbeitshilfen und nicht als dogmatisches Schema verstehen und bereits davon ausgehen, dass sie von Fall zu Fall der Anpassung bedürfen.

Zu den im ersten Entwurfsschritt erwähnten Software-Voraussetzungen gehören die Agentenplattform JADE, die damit verbundene Entwicklung eines MAS und die gewünschte Berücksichtigung klassischer E-Business-Standards und -Systeme. Um die zu entwerfende Ontologie unter JADE praktisch sinnvoll einsetzen zu können, ist es nötig, sich am Ontologiekonzept der FIPA im Allgemeinen und an dem der konkreten Plattform im Speziellen zu orientieren. Dies bedeutet eine Vorentscheidung bezüglich der im Codierungs-Schritt festzulegenden Metaontologie. Die FIPA definiert ein frame-basiertes Ontologiemodell, das aus der von ihr spezifizierten Sprache SL hervorgeht.

Die Metaontologie wird von der FIPA nur implizit spezifiziert, durch ihre Verwendung z.B. für die „Agent Management Ontology“ und durch Aufbau und Mächtigkeit der Sprache SL.

Die JADE-Plattform implementiert das oben genannte Modell in objektorientierter Weise in der Sprache Java. Wesentliche Teile der grundlegenden Begriffe — der Metaontologie — stehen also schon im Vorfeld des Entwurfs fest: Die Ontologie muß auf Basis von Klassen, Slots und Facetten entworfen werden. Verschiedene Eigenschaften und Ausprägungen dieser drei Elemente ergeben sich zusätzlich aus den bekannten Voraussetzungen. Lediglich die Beschreibungsweise der Facetten ergibt sich nicht zwingend aus den Vorgaben. Allerdings empfiehlt sich auch hier die Verwendung von Standards aus „einem Hause“, nämlich den Einsatz der Sprache SL zur Beschreibung der Axiome, die die Slots betreffen.

Zusätzlich zu den Implikationen für die Metaontologie ergeben sich aus den FIPA-Spezifikationen auch Vorgaben und Ausgangspunkte für die eigentliche Ontologie. So fließen beispielsweise Konzepte wie 'Agent' und 'Nachricht' in die zu entwerfende Ontologie ein. Die FIPA-Spezifikationen können also im Sinne des oben angeführten Integrations-schritts als bestehende Ontologie verstanden werden, auf der die Verhandlungsontologie aufbaut.

Viele Entwurfsmethoden für Ontologien zielen darauf ab, eine Domäne möglichst vollständig zu beschreiben. Dies gilt in gewissem Maße auch für die oben vorgestellten. Zwar wird auch hier das Prinzip der minimalen Verpflichtung verfolgt, aber es steht in ständigem Widerstreit zu den anderen beschriebenen Kriterien, denen im Zweifelsfall oft Vorrang gewährt wird, was insgesamt doch in einer möglichst umfassenden Ontologie resultiert. DiLeo et.al. legen dar (DiLeo, et al., 2002), dass eine vollständige Beschreibung einer Domäne durch eine Ontologie für die praktische Verwendung in einem MAS oft nicht angebracht ist. Die Ontologie sollte sich auf die für den Betrieb des Systems notwendigen Aspekte der Domäne beschränken. Zum einen begrenzt dies die Komplexität des Systems auf ein handhabbares Maß. Zum anderen ist die Ontologie eine

Voraussetzung für die Wiederverwendbarkeit des Systems, und jedes System, das das Entwickelte weiterverwenden soll, muss sicherstellen, dass seine eigene Ontologie nicht in Konflikt mit der für das ursprüngliche System entworfenen steht. Das Potential für solche Konflikte ist um so größer, je umfassender die Ontologien gehalten werden. Beim Entwurf der Verhandlungsentologie soll also das Prinzip der minimalen Verpflichtung vorrangig eingehalten werden.

Die oben genannten Kriterien und die beschriebenen Entwurfsmethoden mit den hier aufgeführten Anpassungen sind die Werkzeuge, mit denen die EPN-Ontologie im Folgenden entworfen wird.

5.2.2 Konzepte der Verhandlungsentologie

Die entwickelte Verhandlungsentologie basiert im Wesentlichen auf dem in Kapitel 3 vorgestellten formalen Basismodell für Verhandlungen. Das Grundgerüst dieses Modells bilden eine Reihe formaler Definitionen auf der Basis von relationaler Algebra. Es enthält jedoch noch Teile die in umgangssprachlicher Form beschrieben werden. Die in der Verhandlungsentologie definierten Konzepte, Prädikate und Axiome orientieren sich an den Definitionen des Modells. Sie verstärken dessen formalen Charakter, indem sie weitere Bestandteile und Aspekte des Modells, die zuvor nur umgangssprachlich oder implizit vorlagen, formalisieren. Die Bestandteile der Ontologie sind in eine hierarchischen Klassenstruktur eingeordnet, die auf dem Content Reference Model (CRM) aufbaut. Sie entsprechen damit den FIPA-Spezifikationen zugrundeliegenden Ontologiekonzept und lassen sich so mit Hilfe von JADE in einfacher Weise praktisch umsetzen. Zu detaillierteren Erläuterung der Semantik der Konzepte wird ihre umgangssprachliche Beschreibung durch formale Axiome in SL (FIPA, 2002f) ergänzt.

In den nachfolgenden Abschnitten werden die einzelnen Konzepte der Verhandlungsentologie durch Klassen definiert. Eine Übersicht über ausgewählte zentrale Konzepte und Klassen der Verhandlungsentologie wird in Abbildung 5.1 auf Seite 215 gegeben.

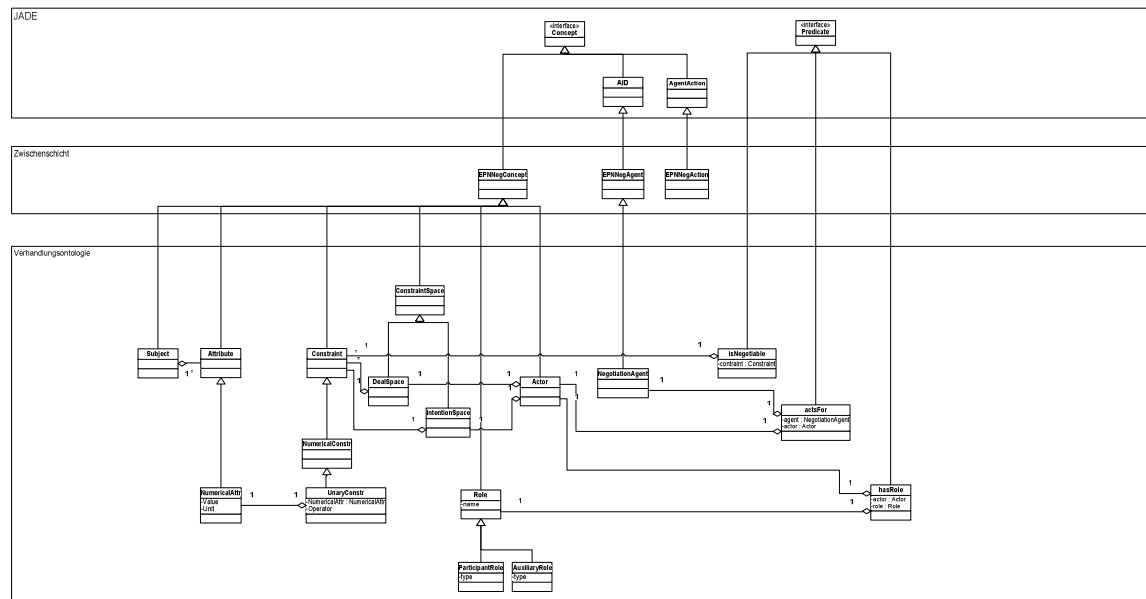


Abbildung 5.1: Zentrale Konzepte und Klassen der Verhandlungsontologie

Jede im Folgenden detailliert dargestellten Definitionen dieser Klassen beginnt mit einer schematischen Darstellung der Klasseneigenschaften, die sich in ihrer Struktur an der in (FIPA, 2002c) eingesetzten orientiert. Dabei werden folgende Bezeichnungen verwendet:

Klasse: Der Name des Konzeptes, der in der Repräsentation jeder Instanz der Klasse verwendet werden muss.

Oberklasse: Der Name der Oberklasse, von der die Klasse in der Ontologiehierarchie erbt.

Ontologie: Bezeichner der Ontologie, zu der die Klasse und ihre Slots gehören.

Parameter: Die Namen der Slots, die zu der Klasse gehören.

Beschreibung: Eine natürlichsprachliche Kurzbeschreibung der Semantik jedes Parameters.

Präsenz: Die Präsenz ist eine einfache Facette jedes Parameters, die bestimmt, ob dieser vorgeschrieben oder optional ist.

Typ: Der Typ der Werte, die ein Parameter annehmen kann.

Reserviert: Vordefinierte und reservierte Konstanten, die der Parameter als Wert annehmen kann.

Im Anschluss an die schematische Darstellung folgt jeweils eine natürlichsprachliche Beschreibung der Semantik der gesamten Klasse und ihrer Bestandteile, die ggf. durch formale Axiome in SL ergänzt wird.

Im Folgenden werden die Kernkonzepte der Verhandlungsontologie exemplarisch vorgestellt.

Attribute

Klasse	attribute			
Oberklasse	epn-negotiation-concept			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserviert
Name	Bezeichner des Attributs	Obligato- risch	string	

Ein Attribut ist eine isolierte Eigenschaft eines Verhandlungsgegenstandes, die für sich betrachtet werden kann und einen Aspekt des Gegenstandes durch ihren Wert beschreibt. Ein Attribut kann unterschiedliche Werte annehmen und die Einigung auf einen von diesen kann Ziel der Verhand-

lung sein. Der Verhandlungsgegenstand wird in Bezug auf das Verhandlungsergebnis durch seine Attribute vollständig beschrieben.

Dies Konzept ist eine abstrakte Oberklasse, die der im Basismodell beschriebenen Attribute übergeordnet ist.

Klasse	numerical-attribute			
Oberklasse	attribute			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
Name	Bezeichner des Attributs	obligatorisch	string	
Value	Numerischer Wert des Attributs	obligatorisch	number	
Unit	Maßeinheit des Attributwertes	optional	unit	

Ein numerisches Attribut ist eine Eigenschaft eines Verhandlungsgegenstandes, die sich durch einen numerischen Wert und eine Einheit quantifizieren läßt.

Beispiele hierfür sind der Preis (in Euro) oder die Geschwindigkeit (in MHz) eines Prozessors, über dessen Kauf verhandelt wird.

Dies Konzept ist eine spezialisiertes Attribut und entspricht sinngemäß dem im Basismodell beschriebenen Attribute, mit dem Unterschied, daß hier Wert und Einheit dem numerischen Attribut direkt zugeordnet werden, während dies ursprünglich nur durch das Konzept der Domain geschah.

Beschränkungen

Klasse	constraint			
Oberklasse	epn-negotiation-concept			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.

Beschränkungen spezifizieren die Menge der möglichen Werte, die ein Attribut annehmen kann. Mengen von Beschränkungen wiederum beschreiben die Handels- und Intentionsräume der Akteure.

Klasse	numerical-constraint
Oberklasse	constraint

Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.

Numerische Beschränkungen spezifizieren die Menge der Werte, die ein numerisches Attribut annehmen kann. Im einfachsten Fall handelt es sich dabei um eine unäre Beschränkung, aber es können auch mehrere Attribute durch binäre Beschränkungen zueinander in Beziehung gesetzt werden.

Klasse	unary-constraint			
Oberklasse	numerical-constraint			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
attribute	Das numerische Attribut, das durch diese Beschränkung beschrieben wird.	obligatorisch	numerical-attribute	
operator	Der Operator beschreibt die Art der unären Beschränkung, die dem Attribut auferlegt wird.	obligatorisch	string	<, <=, =, =, >=, >

Unäre Beschränkungen beschreiben die möglichen Werte eines einzelnen numerischen Attributs unabhängig von anderen Attributen mit Hilfe der definierten Operatoren. Dies Konzept entspricht oben und in (Ströbel, 2003) definierten Domain. Bei Verwendung des Operators „=“ ergibt sich eine diskrete Beschränkung, bzw. eine Punktbeschränkung, falls für ein Attribut nur eine einzelne solche Instanz existiert. Ein stetige Beschränkung ergibt sich durch die restlichen Operatoren und einen Spezialfall der stetigen bilden die Ausschlußbeschränkungen, bei denen der Operator „|=“ verwendet wird.

Strukturelle Unterschiede zu der ursprünglichen Domain ergeben sich aus der geringfügig abweichenden Definition der numerischen Attribute.

Beschränkungsräume

Klasse	constraint-space			

Oberklasse	epn-negotiation-concept			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.

Ein Beschränkungsraum ist eine allgemeine Menge von Instanzen unterschiedlicher Beschränkungen. Er beschreibt die Menge der möglichen Werte aller Attribute und somit die möglichen Konfigurationen des Verhandlungsgegenstandes.

Klasse	deal-space			
Oberklasse	constraint-space			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
constraints	Die Menge von Beschränkungen, aus denen sich der Handelsraum zusammensetzt.	obligatorisch	Set of constraint	
actor	Die Menge von Beschränkungen, aus denen sich der Handelsraum zusammensetzt.	obligatorisch	actor	

Ein Handelsraum ist ein Beschränkungsraum, der die aus Sicht eines Agenten möglichen Konfigurationen des Verhandlungsgegenstandes beschreibt.

Klasse	intention-space			
Oberklasse	constraint-space			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
constraints	Die Menge von Beschränkungen, aus denen sich der Intentionsraum zusammensetzt.	obligatorisch	Set of constraint	

actor	Die Menge von Beschränkungen, aus denen sich der Handelsraum zusammensetzt.	obligatorisch	actor	

Ein Intentionsraum ist ein Beschränkungsraum, der die für einen Agenten akzeptablen Konfigurationen des Verhandlungsgegenstandes beschreibt. Die Menge der Beschränkungen eines Intentionsraumes eines Agenten ist eine Untermenge der Beschränkungen des entsprechenden Handelsraumes: (exemplarisch in SL:)

```
(forall ?c
  (forall ?isc
    (forall ?a
      (implies (and (intention-space :constraints
?isc :actor ?a)
                    (member ?c ?isc))
                (and (deal-space :constraints ?dsc
?actor ?a)
                    (member ?c ?dsc)) ))))
```

Akteure und Rollen

Klasse	actor			
Oberklasse	epn-negotiation-concept			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.

Ein Akteur ist eine Person, die an einer Verhandlung beteiligt ist. Sie kann dabei eine oder mehrere Rollen annehmen und diese u.U. auch im Verlauf der Verhandlung wechseln. Im Kontext des EPN-Systems wird ein Akteur durch einen Verhandlungsagenten vertreten.

Klasse	role			
Oberklasse	epn-negotiation-concept			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
name	Optionaler Bezeicher der	optional	string	

	Rolle			
type	Typisierung der Rolle im Verhandlungskontext; bestimmte Typen sind in den Unterklassen reserviert.	obligatorisch	string	

Eine oder mehrere Rollen werden im Verlauf einer Verhandlung mit einem Akteur assoziiert. In Abhängigkeit von diesen Rollen unterliegen die Akteure bzw. die sie vertretenden Agenten bestimmten Einschränkungen. Die Rollen eines Akteurs bestimmen seine Rechte und Pflichten und die Erwartungen anderer Akteure bzgl. seiner Handlungen. Dieses Konzept entspricht sinngemäß der ‚Agent role‘, wird jedoch durch seine Unterklassen zunächst in zwei unterschiedliche Kategorien unterteilt.

Klasse	participant-role			
Oberklasse	role			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
name	Optionalen Bezeichner der Rolle	optional	string	
type	Typ der Teilnehmerrolle. Derzeit existieren reservierte Werte für Käufer und Verkäufer.	obligatorisch	string	buyer seller

Dieses Konzept beschreibt die Rollen der Verhandlungsteilnehmer, also jener Akteure, die mit einem eigenen Interesse am Verhandlungsgegenstand an der Verhandlung beteiligt sind. Reservierte Typen von Teilnehmern sind derzeit Käufer und Verkäufer (buyer und seller).

Klasse	auxiliary-role

Oberklasse	role			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
name	Optionaler Bezeicher der Rolle	optional	string	Intermediary, supervisor
type	Typ der Zusatzrolle. Derzeit existieren reservierte Werte für Vermittler und Verhandlungsführer.	obligatorisch	string	buyer seller

Dieses Konzept beschreibt die Rollen von Akteuren die ohne eigenes Interesse am Verhandlungsgegenstand aber mit Interesse an der Verhandlung selbst beteiligt sind. Dabei kann es sich um Verhandlungsführer oder Vermittler handeln, z.B. um einen Auktionator, der Gegenstände für einen anderen versteigert.

Klasse	negotiation-agent			
Oberklasse	epn-negotiation-agent			
Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.

Ein Verhandlungsagent im Sinne des EPN-Systems ist ein FIPA-Agent, der einen Akteur in mehr oder weniger autonomer Weise vertritt. Im Kontext des EPN-Systems interagieren Akteure nie direkt, sondern immer durch die sie repräsentierenden Agenten.

5.2.2.1 Prädikate der Verhandlungsontologie

Hat Rolle

Klasse	has-role
Oberklasse	epn-negotiation-predicate

Ontologie	epn-negotiation			
Parameter	Beschreibung	Präsenz	Typ	Reserv.
actor	Der Akteur, der die zugeordnete Rolle inne hat.	obligatorisch		
role	Die Rolle, die der zugeordnete Akteur inne hat.	obligatorisch	role	

Dieses Prädikat ordnet einem Akteur eine oder mehrere Rollen innerhalb einer Verhandlung zu. Die Zuordnung der Rollen kann sich u.U. im Verlauf der Verhandlung ändern.

5.3 Vermittlung zwischen B2B-Standards

Akzeptiert man die Realität der verschiedenen Standards im Business Messaging, so ist ein wesentliches Problem dieses Bereichs die „Vermittlung“ zwischen diesen Standards.

Da die Verhandlungsnachrichten mit dem Business Messaging integriert werden sollen, muss auch in diesem Zusammenhang mit vielen differierenden Standards umgegangen werden.

Dazu werden einige derzeitige erfolgversprechende Methoden zur Vermittlung betrachtet.

5.3.1 Ebenen der Vermittlung

Bei der Vermittlung auf der Transportebene geht um die Fragestellung, wie eine Nachricht von System A zu System B kommt und über welche Adressierungsmechanismen eine Zuordnung zusammengehörender Nachrichten möglich ist.

Vereinfachend (aber in heutigen Szenarien mit den dort verwendeten Transportprotokollen durchaus realistisch) ist die Annahme, dass diese Ebene nicht mit der Vermittlung auf Nachrichtenebene in Berührung kommt.

Die Zahl der auf den Bereich des Electronic Commerce spezialisierten XML-Dialekte geht in die hunderte. Auch wenn viele davon bestimmte Nischen abdecken, liegt die Zahl der horizontalen, also nicht für spezielle

Branchen oder Einsatzgebiete geschaffenen Standards immer noch im mittleren zweistelligen Bereich. Da diese bereits in tatsächlich existierenden Systemen eingesetzt werden und zum Teil signifikante Verbreitung gefunden haben, ist es äußerst unwahrscheinlich, da sich in näherer Zukunft diese Situation zugunsten eines oder einiger weniger Standards ändert.

Folglich ist für die Interoperabilität von unterschiedlichen Systemen eine wie auch immer geartete Vermittlung nötig. Müssen in der Regel etliche Hürden genommen werden. Die Unterschiede können auf syntaktischer Ebene, als auch auf semantischer Ebene liegen.

Ein System wird eine bestimmte Anzahl von Standards unterstützen wollen. Eine direkte Vermittlung zwischen jeweils einem zu allen anderen Standards erweist sich allerdings als ungünstig. Geht man von einem bidirektionalen Mapping aus, dann sind bei n Standards $n * (n-1)$ verschiedene Umsetzungen nötig.

Dieser Ansatz skaliert also schlecht. Um die Zahl der nötigen Umsetzer klein zu halten, wird daher, so fern mehr als zwei Standards involviert sind, in der Regel ein Zwischendialekt verwendet, so das von den allen anderen Standards in dieses Format sowie daraus zurück in alle betroffenen Standards übersetzt wird. Daraus ergeben sich $2 * n$ unterschiedliche Umsetzungen, also eine deutliche Reduzierung des Aufwandes gegenüber einem quadratischen Wachstum.

Orientiert an der Topologie wird der erste Ansatz auch Peer-To-Peer, der zweite als ‚Hub-and-Spoke‘ bezeichnet.

Da die Vermittlung auf diesen Ebenen völlig verschiedene Lösungsansätze bedingen, ist es sinnvoll, zu Beginn eine Klassifizierung der möglichen Probleme vorzunehmen.

In der Literatur existieren bereits mehrere Ansätze, mögliche Ebenen der Heterogenität zu beschreiben. Leider unterscheiden diese sich nicht nur in der Granularität, sondern auch in einem unterschiedlichen Verständnis der gleichen Begriffe. Stellvertretend sein hier nur zwei Ansätze vorgestellt.

(Ouksel, et al., 1999) definiert vier Ebenen der Heterogenität:

- die *Systemebene* (system level) betrifft unterschiedliche Hardware und Betriebssysteme. Damit sind beispielsweise unterschiedliche, zu einander inkompatible Netzwerktechnologien, unterschiedliche Repräsentationen auf Systemebene (wie Big Endian und Little Endian für die Kodierung von Zahlen) oder sehr grundsätzliche Unterschiede zwischen Betriebssystemen gemeint.

- die *syntaktische Ebene* (syntactic level) betrifft die Serialisierung der Daten, also die Repräsentation als Zeichenkette. In diese Ebene fällt beispielsweise die mögliche unterschiedliche Darstellung der Daten in XML-Schreibweise, wie heutzutage üblich, gegenüber der Formatierung als ASCII-Zeichenkette, definierten Trennzeichen zwischen einzelnen Elementen, wie dies im klassischen EDI der Fall ist.
- die *Strukturebene* (structural level) umfasst unterschiedliche Datenmodelle, die hinter den modellierten Konzepten stehen. Hier werden im Unterschied zu syntaktischen Heterogenität zwar grundsätzlich die gleiche Art der syntaktischen Repräsentation verwendet, allerdings werden die konzeptuellen Artefakte unterschiedlich strukturiert, obwohl nicht zwingend Unterschiede auf der nachfolgend besprochenen semantischen Ebene auftreten. Beispielsweise könnte eine Adresse in einem System Strasse und Hausnummer als zwei getrennte Attribute darstellen, werden diese bei einem anderen System zusammengefasst werden.
- auf der *semantischen Ebene* (semantic level) können Unterschiede durch die Verwendung von synonymen Begriffen für gleiche Konzepte auftreten. Dies ist ein recht einfaches Beispiel, in dem Sinne, das diese Unterschiede, zumindest durch einen Domänenexperten, recht einfach aufzulösen sind. Schwieriger sind Fälle, in denen ein Konzept lediglich in einem der betroffenen Systemen abgebildet ist.

Die Systemebene wird hier nicht näher betrachtet. Es wird davon ausgegangen, dass die miteinander kommunizierenden Systeme Internet fähig sind, also direkt oder mittels entsprechender Gateways über IP-basierende Netze miteinander sprechen. Unterschiedliche Byte Order spielen dadurch keine Rolle mehr, da TCP/IP in sämtlichen Headern Big Endian (auch Network Byte Order genannt) vorschreibt. Ein System, welches ursprünglich eine andere Byte Order verwendet, muss und wird in seiner Netzwerksoftware diese Anpassung explizit vornehmen – die Nutzdaten werden auf der syntaktischen Ebene behandelt.

Ein anderer Ansatz zerlegt die Problematik in drei Schichten (Melnik, et al., 2000; Omelayenko, et al., 2001):

- Die *Syntaktische Schicht* (Syntactic Layer) beschreibt die Serialisierung der Dokumente in eine Folge von Zeichenketten (ist also konzeptuell auch hier nicht auf XML festgelegt)
- Die *Objektschicht* (Object oder Data Model Layer) soll einen objekt-orientierten Zugriff auf die Informationen ermöglichen, da-

zu werden die Daten aus dem syntaktischen Layer in ein normiertes Tripel-Format (konkret: RDF) übertragen, um syntaktisch unabhängig von dem Ursprungsdokument zu werden. Hier werden bereits strukturelle Unterschiede aufgelöst, soweit diese sich aus der syntaktischen Ebene ableiten lassen

- Die *semantische Schicht* (Ontology oder Semantic Layer) - hier wird eine generische Ontologie für jedes mögliche Geschäftsdokument (beispielsweise Katalog oder Bestellung im B2B Kontext) definiert. Diese Ontologie wird idealerweise mehr Informationen ($\approx$ Obermenge) enthalten, als ein Geschäftsdokument eines bestimmten Formates, wenn für die Mediation eine verlustfreie Abbildung der Objektschicht auf die semantische Schicht erwünscht ist.

Die Unterschiede auf den einzelnen Ebene können weiter aufgegliedert werden.

Es sei an dieser Stelle vorweggenommen, das für die Implementierung des Prototypen der letztere Ansatz gewählt wurde – hauptsächlich aus dem Grunde, weil diese Aufteilung intuitiv eine etwas natürlichere Abbildung des theoretisch formulierten Konzeptes auf objekt-orientierte Programmiersprachen im Zusammenhang mit der Behandlung von XML-Dokumenten darstellt.

5.3.2 Repräsentationstechniken von Konzepten im B2B Umfeld

Das hinter den zur Zeit gängigen B2B XML-Standards liegende konzeptuelle Modell ist in der Regel nicht explizit noch formal formuliert – vielmehr sind die Dokumenttypen der Standards über geeignete XML-Mechanismen, also DTDs oder Schema-Spezifikationen, so wie eine begleitende Prosa Dokumentation definiert.

Aus der Definition der Schemata lassen sich jedoch trotzdem mehr oder weniger gut gewisse Informationen, wie beispielsweise Beziehungen zwischen einzelnen Artefakten ableiten. Allerdings haben DTDs im Vergleich zu XML Schema in diesem Bereich erhebliche Mängel, da sie ursprünglich aus der Dokumentenverarbeitung stammen und viele, sowohl für den programmier-sprachlichen Umgang als auch aus Sicht der konzeptuellen Ausdrucksstärke, wichtige Merkmale vermissen lassen. So fehlt beispielsweise die Möglichkeit, Datentypen festzulegen ebenso wie das Konzept der Vererbung (Generalisierung und Spezialisierung).

Abhilfe in dieser Situation können Ontologien schaffen.

Ontologien sind die explizite Konzeptualisierung einer Domäne (Gruber, 1993).

Die Betonung liegt besonders auf dem Wort „explizit“. Für die sinnvollen Umgang von Rechnern mit Ontologien müssen diese in einer geeigneten, also maschinenlesbaren und formalen Beschreibung vorliegen - da der Begriff ursprünglich nicht aus der elektronischen Datenverarbeitung stammt, ist dies per se nicht zwingend der Fall.

Formalisierte Ontologien bzw. Ontologiesprachen basieren in der Regel auf einer bestimmtem Logik-Ausprägung wie der Beschreibungslogik (Untermenge der Prädikatenlogik erster Stufe) oder Frame-Logik ((Kifer, et al., 1995), eine Logik-Art, welche die Konzepte der Objekt-Orientierung formalisiert).

Dabei können die Unterschiede auf ganz verschiedenen Ebenen liegen:

- Syntaktische Unterschiede
 - o Techniken zur Transformation
 - o Behandlung von nicht XML-konformen Standards (Bsp. EDI)
- Semantische Unterschiede
 - o Ontologien
 - o andere Techniken (vorkonfigurierte Adapter)
- Klassifizierung und typische Problemstellungen bei der Transformation
- Automatisierbarkeit der Transformation

Wichtig ist, auf welche Art und Weise die Vermittlung zwischen den unterschiedlichen Standards stattfindet. Neben der Wahl bestimmter Ansätze oder Algorithmen geht es vor allem um die Entscheidung, ob die nötigen Abbildungen zwischen den Konzepten der einzelnen Standards statisch oder zur Laufzeit der Vermittlung festgelegt werden.

Die Problematik ist nicht neu – schon vor der Existenz der gängigen, XML-basierten Standards im B2B Bereich war sie Gegenstand wissenschaftlicher Untersuchungen, beispielsweise im Datenbankbereich seit Beginn der achtziger Jahre. In klassischen Datenbankanwendungen ergibt sich häufiger das Problem, aus mehreren Datenquellen, deren Schemata ähnliche Konzepte abbilden, eine einheitliche Ansicht zu präsentieren (Forschungsgegenstand seit den 80er Jahren).

Ein Anwendungsbeispiel ist die Erzeugung unternehmensweiter Reports, wo lokal an den einzelnen Standorten ähnliche, aber unterschiedliche Anwendungen mit unterschiedlichen dahinter liegenden Datenbank-Schemata zur Anwendung kommen (Data Warehousing). Die entspre-

chenden Untersuchungen und Lösungsansätze wurden unter dem Begriff „Schema Matching“ bekannt.

Eine Übersicht findet sich in (Rahm, et al., 2001). Sie geht auf die wichtigsten Klassifikationsmerkmale ein und stellt eine Reihe von aus dem wissenschaftlichen Umfeld hervorgegangenen Prototypen vor.

Dem Matching können unterschiedliche Strategien zugrunde liegen. Grundlage der folgenden Taxonomie ist (Rahm, et al., 2001). Die Taxonomie wurde jedoch teils erweitert:

- Design-time vs. Run-time: Werden die Mappings initial definiert oder jedes mal automatisch zur Laufzeit ermittelt (dort können Lernfaktoren eine Rolle spielen)?
- Instanz vs. Schema: Werden tatsächliche Datenwerte und ihre per Schema definierten Konzepte abgebildet?
- Element vs. Struktur: Werden einzelne Elemente, also atomare Artefakte, auf jeweils ein anderes Element abgebildet, oder werden komplette Strukturen zusammengesetzt aus mehreren Elementen betrachtet?
- Linguistisch vs. Constraints: „Linguistisch“ kann bedeuten, Wörterbücher, Thesauri oder auch Häufigkeiten, Constraint-based Mapping könnte Matchkandidaten anhand ihrer Datentypen (in Kombination mit Schema-basiert) oder anhand auftretender Intervalle (bei Betrachtung der auftretenden Instanzen) oder auch ihrer Struktur mit Hilfe von Graphen abzubilden
- Kardinalitäten der Matching-Abbildung (1:1, 1:n/n:1, n:m): Welche sind möglich? Kann beispielsweise Vorname, Nachname auf Empfänger abgebildet werden (n:1-Mapping)? Dieser Punkt steht im Zusammenhang mit dem dritten Punkt.
- Individual vs. Combinational (Hybrid Matchers, Multiple Matchers): Kommen mehrere Matcher kombiniert zum Einsatz (hybrid), oder werden die Ergebnisse mehrerer einzelner Mapping-Vorgänge am Ende vereinigt?

Ein Mapping wird häufig durch eine Ähnlichkeitsrelation ($\sim=$, similarity relation) ausgedrückt:

$$\text{Schema } S_x = (e_1, e_2, \dots, e_n)$$

wobei e_i jeweils die in dem Schema auftretenden Elemente bezeichnet.

Die Ähnlichkeitsrelation entspricht dann der Abbildung:

$$\text{Pot}(S_1) \sim= \text{Pot}(S_2)$$

welche eine Menge von Elementen eines Schemas eine konzeptuelle Ähnlichkeit mit einer Menge von Elementen des anderen Schemas zuordnet.

Die Ähnlichkeitsrelation gibt jedoch kein Maß für die Güte der vermuteten Übereinstimmung an.

Eine Quantifizierung der Güte der Übereinstimmung erzeugt die Affinitätsfunktion: Sie ordnet zwei Konzepten unterschiedlicher Quellen einen Wert zwischen 0 und 1 zu. Der Wert entspricht einer Wahrscheinlichkeit, das die beiden Elemente das gleiche Konzept modellieren.

Die Affinitätsfunktion ist insbesondere bei der Verwendung von kombinierten Ansätzen, in denen unterschiedliche Matching-Komponenten zum Einsatz kommen, wichtig.

Abschließend sei daher kurz ein prototypisches System vorgestellt, welches in (Doan, et al., 2001) dokumentiert ist und einen solchen kombinierten Ansatz aufweist.

Der Ansatz ist ein lernfähiges System, welches aus bereits (manuell) hergestellten Mapping einiger Schemata auf ein intermediäres Schema Regeln ableitet.

Ein Hinweis, der sich aus bereits hergestellten und verifizierten Abbildungen ableiten lässt ist z.B. die Tatsache, dass der Preis eines Hauses in der Regel eine sechsstellige Zahl ist (abgeleitet aus Dateninstanzen), oder dass Elemente, die in ihrem Namen den Teilstring ‚Phone‘ beinhalten, vermutlich auf die entsprechende Telefonnummer im intermediären Schema gemapped werden (abgeleitet aus dem Schema).

Ferner werden strukturelle Informationen ausgewertet: ein Element mit „Id“ im Namen wird beispielsweise nicht auf die Angabe von Zimmern in einem Haus gemapped, das aus einer gewissen Menge von Instanzen klar wird, dass diese Werte (Anzahl der Zimmer) mehrfach auftreten, es sich bei einem Element mit „Id“ im Namen aber offensichtlich um einen (eindeutigen) Schlüssel handelt. Im Falle von Datenbankschemata können hier explizit definierte Constraints ausgewertet werden, im Falle von XML können hierarchische Strukturen identifiziert werden.

„Userfeedback“ durch manuell durchgeführte Korrekturen sollen ebenfalls in den Lernprozess einfließen.

Bei der konkret vorgeschlagenen Architektur handelt es sich um einen „Multi-Strategy Learning Approach“ mit den folgenden besonderen Eigenschaften:

- Es existieren mehrere lernende Matcher, sogenannte Base Learners, die in der Regel unterschiedliche Strategien und Algorithmen verwenden.
- Darüber sitzt ein Meta Learner, der die Ergebnisse der Base Learners zusammenführt.
- Bei der Zusammenführung gewichtet der Meta Learner die Ergebnisse der einzelnen Base Learners unterschiedlich, Grundlage der Bewertung sind die Matching-Qualitäten der Base Learners, wenn diese die bereits manuell hergestellten Mappings selbst berechnen. Die Granularität der Bewertung kann dabei bis auf Elementebene gehen – ein Base Learner kann beispielsweise bestimmte Teile eines Schema gut matchen, andere schlecht.
- Das System ist so ausgelegt, das jederzeit neue Base Learners integriert werden können.

Die Prototypen zum Matching sind in der Regel nicht frei erhältlich, was eine Evaluation schwierig gestaltete. Noch wichtiger ist die Tatsache, dass in der Literatur einheitlich die Meinung vertreten wird, das ein so genanntes Schema Matching nicht *vollautomatisiert* durchgeführt werden kann.

Viel mehr generieren alle bekannten Ansätze Kandidaten, die dann von einem (menschlichen) Domänen-Experten akzeptiert oder verworfen werden können. Diese Leistung stellt jedoch bereits eine signifikante Verbesserung gegenüber einem komplett manuell durchgeführten Matching dar.

Für die Implementierung ist daher eine Generierung der Abbildungsregeln zur Laufzeit nicht angebracht, da diese wie oben beschrieben nicht zuverlässig automatisch generiert werden können und ein manueller Eingriff in diesen Vorgang während einer Verhandlung nicht erwünscht ist. Menschliche Eingriffe sollen höchstens auf der Verhandlungsebene selbst, nicht jedoch in den darunter liegenden Prozessen erfolgen.

5.4 Entwurf der B2B-Ontologie

Als Ergänzung der Verhandlungsontologie ist eine B2B-Ontologie notwendig, um einen interoperablen Austausch über Verhandlungsobjekte zu ermöglichen.

Ontologien sind die explizite Konzeptualisierung einer Domäne (Gruber, 1993). Ontologien, die grundlegende Konzepte abbilden, welche

unabhängig von dem Einsatzgebiet sind, werden als Oberontologien („upper-level ontologies“) bezeichnet. Sie definieren elementare Begriffe wie Raum und Zeit. Ontologien, welche Konzepte eines bestimmten Fachgebietes formalisieren, werden als Domänenontologien („domain ontologies“) bezeichnet (Geerts, et al., 2000).

Im weiteren Verlauf geht es um die Konstruktion einer solchen Domänenontologie.

Zunächst wird die Entwurfsmethodologie vorgestellt, nach der die B2B-Ontologie entworfen wurde.

5.4.1 Entwurfsmethoden für B2B-Ontologien

Oben wurden schon allgemein Entwurfsmethoden für Ontologien vorgestellt. Die im Folgenden dargestellte Methode von Noy und McGuinness (Noy, et al., 2001) wurde konkret für den Entwurf der B2B-Ontologie verwendet.

Eine Ontologie definiert die statische Struktur einer Begriffswelt – kommen Instanzen hinzu, also hat man beispielsweise eine Wein-Datenbank mit 550 Weinen basierend auf einer Wein-Ontologie, so spricht man von einer Knowledge Base (Wissensdatenbank).

Der Entwurf einer Ontologie ähnelt eher einer Kunst denn einer Wissenschaft. In der Literatur existieren eine Reihe von Ansätzen, den Entwurf von Ontologien methodisch zu untermauern (Jones, et al., 1995). Allerdings besteht Einigkeit darüber, dass es nicht nur einen richtigen Weg gibt Ontologien zu entwerfen.

Der Ansatz von von Noy und McGuinness (Noy, et al., 2001) sieht im wesentlichen die folgende Schritte vor, die hier auf den Kontext des E-Business angepasst wurden:

Die Domäne und den Umfang der Ontologie bestimmen.

Hier geht es um die Frage was Gegenstand der Domäne und damit der zu erstellenden Ontologie ist, beispielsweise Bestellungen im Electronic Commerce. An dieser Stelle kann es sinnvoll sein, Abgrenzungskriterien zu definieren, wie z.B. dass das Einholen von Angeboten nicht Gegenstand der Betrachtung sein soll.

Ferner sollten, so weit möglich, Fragen gestellt werden, die anhand der Ontologie (auf Schema- oder Instanzlevel) beantwortet werden sollen (Competency questions) – wie z.B. die Frage, wie hoch die Einzelsummen oder die Gesamtsumme der bestellten Waren ist, oder ob aus der Adresse ein Ansprechpartner hervorgeht. Dies ist vergleichbar mit einer ‚Testsuite‘ in der Softwareentwicklung.

Existieren Ontologien, die wiederverwendet werden können?

Es gibt mittlerweile einige Bibliotheken von Ontologien. Beispiele dafür sind die ‚Ontolingua Ontology Library‘, <http://www.ksl.stanford.edu/software/ontolingua/> oder die ‚DAML Ontology Library‘, <http://www.daml.org/ontologies/>. Der Aufwand, eine gute und durchdachte Ontologie zu erzeugen, ist in der Regel deutlich höher als eventuell nötige Konvertierungen an der Repräsentation (Dateiformaten) vorzunehmen.

Aufzählen der für die Ontologie wichtigen Konzepte.

In diesem Schritt sollten in recht freier Form die wesentlichen Konzepte dargestellt werden, ohne bereits auf genauere Klassenstrukturen und Zusammenhänge einzugehen. Sie werden in den nächsten Schritten verfeinert und ergänzt.

Definition der Klassen und Klassenhierarchie.

Dieser und die nächsten beiden Schritte müssen nicht streng sequentiell ausgeführt werden. In diesem Schritt werden die im vorherigen Schritt erhaltenen Konzepte in Klassen – oder wenn sinnvoll – in Eigenschaften einer Klasse transferiert.

Bei der Erstellung der Klassenhierarchie sollte die Bedeutung von Unterklassen beachtet werden - sie erben, wie im objekt-orientierten Paradigma, alle Attribute der einer Oberklasse. Eine Unterklasse sollte also konzeptuell immer eine Spezialisierung der Oberklasse sein und nicht zur Gruppierung mehrere Attribute einer Klasse verwendet werden.

Wiederverwendung sollte für sich genommen keinen Ausschlag dafür geben, eine neue Klasse zu formulieren. Auch Attribute können aus einem „Pool“ von bereits definierten wieder verwendet werden. Mehr zu dem Thema weiter unten.

Man unterscheidet ferner zwischen top-down und bottom-up Ansätzen. Beim top-down Ansatz gelangt man über sehr generelle Begriffe zu den Spezialisierungen, beim bottom-up Prozess werden quasi viele kleine Puzzle-Steinchen nach und nach zu einem ganzen zusammengefügt. Keine der beiden Ansätze bietet eindeutige Vorteile gegenüber dem anderen – allerdings erscheint mir ein top-down Ansatz dann sinnvoll, wenn ein sehr gutes und detailliertes Verständnis über die Zusammenhänge der Begriffswelt der betroffenen Domäne existiert, während ein bottom-up Ansatz besser geeignet für ein „verteiltes Erstellen“ scheint, d.h. Teilfelder auf Spezialisten aufzuteilen und diese Fragmente danach zu einem ganzen zusammenzuführen.

Definition der Eigenschaften der Klassen (Slots).

Die Eigenschaften drücken Attribute einer (Instanz einer) Klasse aus. Diese können Eigenschaften wie eine Farbe sein, oder Teile eines Objektes kennzeichnen, so fern dieses strukturiert ist – also beispielsweise der Reifentyp eines Automobils.

Definition der Einschränkungen der Eigenschaften (Facets).

Die Eigenschaften können bestimmten Bedingungen unterliegen. Die wichtigste ist die Kardinalität – sie beschreibt, wie oft die Klasse eine Eigenschaft führt, und ob diese optional ist, oder nicht.

Den Eigenschaften wird üblicherweise ein Datentyp zugeordnet (im einfachsten Falle wird immer mit beliebigen Zeichenketten gearbeitet), wie Zahl, Zeichenkette, Boolescher Wert oder Aufzählungstyp.

Eine weitere interessante Frage ist, wie Beziehungen zwischen Klassen, die nicht Generalisierung bzw. Spezialisierung sind, ausgedrückt werden können. Hierfür wird ein spezieller Wertetyp benutzt: der Instanz-Typ. Auch er ist Eigenschaft einer Klasse, sein Wert zeigt aber auf die Instanz einer anderen Klasse. So kann eine Eigenschaft „Manufacturer“ auf einen Hersteller zeigen, der Instanz einer anderen Klasse ist. Eine solche Eigenschaft funktioniert also ähnlich wie ein Fremdschlüssel in relationalen Datenbanksystemen.

Ferner ist es möglich, Domain und Range eines Slots des Instanz-Typs festzulegen. Die Range definiert eine Menge von Klassen, auf die die Eigenschaft zeigen darf, während die Domain von der anderen Seite greift: Sie bestimmt, von welchen Klassen aus sie mittels eines Instanz-Typs referenziert werden darf. (Auch hier besteht eine gewisse Ähnlichkeit zu relationalen Datenbanksystemen und den dort über eine Verknüpfungstabelle möglichen n-zu-m Beziehungen).

Ontologien erlauben das Prinzip der Mehrfachvererbung. Die Unterklasse erbt dann, wie zu erwarten, die Slots aller Oberklassen.

Erstellen der Instanzen.

Zuletzt werden konkrete Instanzen der vorher definierten Klassen erzeugt und die dazugehörigen Eigenschaften definiert. Dieser Schritt erfolgt in der Regel bei dem Aufbau einer Wissensdatenbank. In dem hier betrachteten Kontext des Electronic Commerce werden die Instanzen in der Regel automatisiert während des Betriebes des Systems als Dokumenten-Instanzen erzeugt.

Die Facets können dabei insbesondere für die Erstellung von Eingabemasken dienen. Beispielsweise können bei der Befüllung eines Aufzählungstyp nur die entsprechend der vorher definierten Ontologie gültigen Auswahlmöglichkeiten angezeigt werden.

Abgesehen von diesem Vorgehen ergeben sich einige, teilweise knifflige Detailfragen:

Bei der Erstellung der Klassenhierarchie stellt sich häufiger die Frage ob etwas als Klasse oder als Eigenschaft einer anderen Klasse formuliert werden sollte. Dabei sollte berücksichtigt werden, das als Klassen in der Regel solche Dinge in Frage kommen, die allein stehend existieren können. Die Beantwortung dieser Frage kann in Grenzfällen jedoch schwierig sein.

Noy und McGuiness (Noy, et al., 2001) versuchen über die Unterscheidung, ob die zu treffenden Aussagen einer unären oder binären Relation entspricht, Hinweise für die Wahl "Klasse" oder "Eigenschaft" abzuleiten:

We can also view classes as unary predicates—questions that have one argument. For example, "Is this object a wine?" Unary predicates (or classes) contrast with binary predicates (or slots)—questions that have two arguments. For example, "Is the flavor of this object strong?" "What is the flavor of this object?"

Leider hängt die Art der Relation recht willkürlich von der Fragestellung ab: "Is this object a redwine?" und „Is the colour of the wine red“ lässt einem auch hier die Wahl zwischen Spezialisierung und Attributierung.

An späterer Stelle wird ein besseres Kriterium definiert: Wie wichtig ist das Konzept in der Ontologie ? Bei einem Automobil ist die Farbe lediglich eine von vielen Eigenschaften - nicht mehr. Bei einem Wein ist die Farbe eine ganz wesentliche Eigenschaft – vermutlich die wichtigste überhaupt. Außerdem können, und das ist ebenso wichtig für die Entscheidung, abhängig von der Farbe des Weines neue Attribute (bei Rotwein z.B. der Tanningehalt) hinzukommen, während die meisten anderen für alle Weine gleich sind.

Genau hier greift dann wieder das Konzept der Spezialisierung, welches das Hauptkriterium für Unterklassen in der Objektorientierung ist.

Sollte ein Konzept eine Klasse oder eine Instanz sein ? Als Beispiel dienen Weinregionen (hier fehlt noch ein plausibles Beispiel aus dem B2B-Bereich). Auf oberster Ebene werden Länder als Klassen definiert (ansonsten können zwar der Klasse „Land“, aber nicht den tatsächlichen Ländern Subklassen zugeordnet werden).

Regionale Gebiete innerhalb dieser Länder könnten nun ebenfalls (Sub-)klassen sein, oder als Instanzen einer Landesregion gesehen werden.

Hier sollte, wenn die Länder selbst auch instanziiert werden können, auch für die Regionen Klassen gewählt werden – ansonsten ist nicht intuitiv klar, welche Regionen Klassen und welche Instanzen sind.

Protégé erlaubt die Definition abstrakter Klassen, die nicht instanziiert werden können.

5.4.2 Exemplarische B2B-Ontologie

Als exemplarisches Beispiel wird der Adressteil der B2B-Ontologie betrachtet. Dazu werden die aus den BMEcat und UBL Standards gebildeten Ontologien, sowie im Anschluss die resultierende EPN-Ontologie dargestellt.

Präsenzen: einfach (0...1), mehrfach (0...n), obligatorisch (1), obligatorisch mehrfach (1...n).

Parent = „thing“ bedeutet, das entweder kein Oberklasse existiert (weil die Ontologie aus einer nicht objektorientierten Struktur gewonnen wurde) oder keine inhaltliche Generalisierung des Konzeptes existiert.

5.4.2.1 BMEcat Ontologie

Frame Parent Ontologie	address thing bmecat_1_2			
Parameter	Beschreibung	Präsenz	Typ	Reserviert
Type	Definiert den genaueren Kontext des Adresselementes. In dieser BMEcat Version sind die Werte „BUYER“ und „SUPPLIER“ gültig.	einfach	enumeration	
Name	Adresszeile	einfach	string	
name2	Adresszeile 2, z.B. für Abteilung	einfach	string	
name3	Adresszeile 3	einfach	string	
Contact	Ansprechpartner	einfach	string	
Street	Straßenname und Hausnummer	einfach	string	
Zip	Postleitzahl	einfach	string	
Boxno	Postfachnummer	einfach	string	
Zipbox	Postleitzahl Postfach	einfach	string	
City	Ort	einfach	string	
State	Bundesland	einfach	string	
Country	Land	einfach	string	
Phone	Telefon	einfach	string	
Fax	Fax	einfach	string	
Email	Emailadresse	einfach	string	
public_key	öffentlicher Schlüssel. Gibt den öffentlichen Schlüssel, z.B. von PGP, der hier adressierten	mehrfach	string	

	Person an			
public_key_type	In der Version 1.2 ist gegenüber Version 1.2 final draft das Attribut "type" hinzugefügt worden, um das Verschlüsselungsverfahren genau spezifizieren zu können.	mehrfach	string	
url	Internetadresse, z.B. http://www.bmecat.org	einfach	string	
address_remarks	Bemerkungen	einfach	string	

Dieses Element dient zur Übertragung von Adressinformationen zum einkaufenden und verkaufenden Unternehmen in BMEcat. Es wird in den Tags BUYER und SUPPLIER verwendet.

5.4.2.2 UBL Ontologie

Frame Parent Ontologie	address thing ubl_0_7			
Parameter	Beschreibung	Präsenz	Typ	Reserviert
Id	a unique identifier given to a specific address within a scheme of registered addresses.	obligatorisch	string	
Postbox	a post office box number or a numbered post box in a post office assigned to a person or organization where letters for them are kept until called for, used as part of an address.	einfach	string	
Building	the identity which uniquely identifies a building within an address.	einfach	string	
Floor	identification by name or number of the floor in a	einfach	string	

	building, as part of an address.			
Room	identifies the room, suite or apartment as part of an address.	einfach	string	
Street	the name of the street, as part of an address.	einfach	string	
additonalstreet	contains additional text information within an address about the street.	einfach	string	
Housename	the name that identifies the location of a house or building: usually within a street as part of an address.	einfach	string	
housetnumber	the house number or description that identifies a house, building or site: usually within a street as part of an address.	einfach	string	
inhousemail	identifies an in-house mail location as part of an address.	einfach	string	
department	the identity of the department of an organisation, used as part of an address.	einfach	single	
Cityname	the name of the city, town, village, or a built up area and used as part of an address.	einfach	single	
Postalzone	the identifier for one or more	einfach	single	

	properties according to the postal service of that country; a group of letters and/or numbers added to the postal address to assist in the sorting of mail.			
Countrysub-entity	the name that identifies the territorial division (a chief unit of local administration) of a country, as part of an address.	einfach	single	
Countrysub-entitycode	the code that identifies the territorial division (a chief unit of local administration) of a country, as part of an address.	einfach	single	
Region	the name that identifies a non-administrative division of a country, or a commonly used name for a grouping of countries, as part of an address.	einfach	single	
District	the name that identifies a non-administrative division of a city, country sub-entity, or country, as part of an address.	einfach	single	
Timezone offsetmeasure	the measure of time offset from UTC (Universal	einfach	single	

	Coordinated Time) for the time zone in which the address is situated. A synonym for UTC is GMT (Greenwich Mean Time).			
ref_country	references an instance of the country class as part of an address.	einfach	instance	classes= country(Rebstock, 2001)

Die UBL ‚address‘ gehört zu den so genannten „reusable“ Komponenten. Sie wird in der UBL von BuyerParty, DestinationParty, SellerParty und FreightForwarderParty benutzt. Die Beschreibung aus der UBL-Dokumentation lautet: „the particulars that identify and locate the place where someone lives or is situated, or where an organisation is situated.“

Frame Parent Ontologie	country thing ubl_0_7			
Parameter	Beschreibung	Präsenz	Typ	Reserviert
Id	provides the country part of an address using a code. ISO3166 alpha codes are recommended.	einfach	string	

Steht für ein Land. Dabei wird nicht der Name, sondern ein Code nach ISO3166 verwendet. Schreibweise(n) des Landesnamens in Landes- oder anderen Sprachen sind hier also nicht ableitbar.

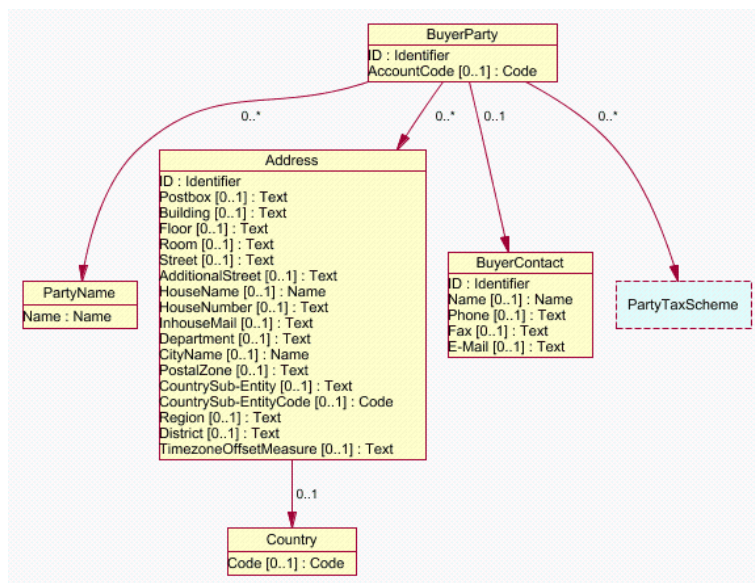


Abbildung 5.2: UBL-Adresselement mit Kontext

5.4.2.3 EPN Business Ontologie

Frame Parent Ontologie	address thing epn- business			
Parameter	Beschreibung	Präsenz	Typ	Reserviert
first_name	Vorname des Adressaten	einfach	string	
middle_name	weitere Vornamen des Adressaten	einfach	string	
last_name	Nachnahme des Adressaten	einfach	string	
Title	Anrede und/oder akademischer Titel des Adressaten	einfach	string	
Company_name	Name der Firma des Adressaten	einfach	string	
Street	Straße des Adressaten	obligatorisch	string	
Zipcode	Postleitzahl oder ähnlich des Adressaten	obligatorisch	integer	

City	Stadt des Adressaten	obligatorisch	string	
State	Bundesland des Adressaten oder ähnliches Konzept. Größer als Stadt aber feiner als Land	einfach	string	
Country	Land des Adressaten	einfach	string	
address_addons	weitere Angaben, wie Gebäude o.ä.	einfach	string	

Die ‚address‘ ist eine Adresse, wie beispielsweise eine Liefer- oder Rechnungsadresse. Sie soll die eindeutige Lokalisierung einer Person oder Firma ermöglichen. Es ist daher inhaltlich erforderlich, dass mindestens Nachname oder Firmenname belegt sind.

Sie ist eine abstrakte Klasse. Ihre Subklassen beschreiben den Kontext, in dem die Adresse verwendet wird:

Frame	billing-address			
Parent	address			
Ontologie	epn-business			
Parameter	Beschreibung	Präsenz	Typ	Reserviert

Die ‚billing-address‘ ist eine Subklasse von ‚address‘. Sie beschreibt eine Rechnungsadresse.

Frame	shipping-address			
Parent	address			
Ontologie	epn-business			
Parameter	Beschreibung	Präsenz	Typ	Reserviert

Die ‚shipping-address‘ ist eine Subklasse von ‚address‘. Sie beschreibt eine Lieferadresse.

5.5 Integration von Verhandlungskommunikation und B2B-Messaging

Da die Verhandlungskommunikation im Rahmen der B2B-Kommunikation stattfindet, sollte sie entsprechend integriert werden. Dabei sind jedoch die Verhandlungsnachrichten nicht nur eine Art der B2B-Interaktion, sondern die B2B-Interaktionen enthalten auch Informa-

tionen die für die Verhandlungsnachrichten relevant sind; im Wesentlichen sind dies Informationen über die Verhandlungsgegenstände.

Bei der Integration treten die folgenden relevanten Elemente auf: die Verhandlungssprache, B2B-Standards für Geschäftsdokumente, sowie Messaging-Frameworks. Die Messaging-Frameworks stammen weithin aus dem Bereich der B2B-Kommunikation, beispielsweise ebXML oder Biztalk und umschließen dann die Geschäftsdokumente, jedoch kann auch die von der FIPA spezifizierte Agentenkommunikation als entsprechendes Messaging-Framework angesehen werden.

Der Lösung dieser Problemstellung sollen generische Konzepte zugrunde liegen, die sich nicht nur auf einzelne Standards oder Techniken beziehen. Für eine beispielhafte konkrete Umsetzung werden jedoch konkrete Standards ausgewählt. Dies sind:

- BMEcat (zusammen mit openTrans, als B2B-Standard für Geschäftsdokumente)
- ebXML (als B2B Messaging Framework)
- xCBL (als B2B-Standard für Geschäftsdokumente)
- EDI (als sowohl historisch als auch vom Umfang der angewandten Lösungen relevanter Standard für Geschäftsdokumente und auch als Messaging Framework)
- FIPA(-ACL) als allgemeines / agentenorientiertes Messaging Framework

Es ergeben sich auch aufgrund der verschiedenen Typen von Elementen mehrere relevante Alternativen für die Integration.

Messaging-Framework wird um Verhandlungsfunktionalität erweitert: Da die Erweiterung innerhalb der Struktur der Frameworks erfolgen würde, müsste jedes einzelne B2B-Framework erweitert werden. Ähnliche Probleme treten auf, wenn einzelne B2B-Standards für Geschäftsdokumente um eine Verhandlungsfunktionalität erweitert würden.

Diese Alternativen werden daher nicht weiter betrachtet.

Verhandlungssprache kapselt Messaging-Frameworks: Die Verhandlungssprache müsste so zum Teil die Funktionalitäten eines Messaging-Frameworks übernehmen.

Diese Alternative wird somit nicht weiter betrachtet.

Messaging Framework enthält Verhandlungsnachrichten: Es wird ein einziges einheitliches Messaging-Framework verwendet; beispielsweise FIPA. Die Kompatibilität zu anderen Messaging-Frameworks muss über entsprechende Adapter gewährleistet werden. Die Nachrichten des Messaging-Frameworks enthalten unter anderem die Sprachprimitive der Verhandlungen. Die Verhandlungsnachrichten können bei Bedarf wieder-

um Geschäftsdokumente als Beschreibung des Verhandlungsgegenstandes oder des sich entwickelnden Vertrages enthalten.

Aus architektureller Sicht ist dieser Ansatz klar und gut strukturiert: er vermeidet Anpassungen von bestehenden Standards, kann aber trotzdem kompatibel zu mehreren existierenden Standards gestaltet werden, da das Messaging Framework Verhandlungsnachrichten wie Geschäftsdokument einfach als Nutzlast versteht.

Trennung von Verhandlungssprache und Standards: Verhandlungs- und B2B-Nachrichten werden grundsätzlich getrennt voneinander behandelt, gegebenenfalls auch über unterschiedliche Transportwege versandt. Dennoch existiert eine Möglichkeit, zugehörige Nachrichten aus beiden Bereichen einander zuzuordnen. Für die Verhandlungssprache muss die Möglichkeit bestehen, einzelne B2B-Nachrichten zu referenzieren. Dazu muss ein für eine Instanz des EPN-Systems zentraler Message Coordinator (MC) die Verwaltung dieser Referenzen übernehmen und die nötigen Mechanismen bereitstellen, um sie bei Bedarf aufzulösen. Auf diese Weise werden einige Probleme anderer Varianten vermieden: Die Kommunikation mit „herkömmlichen“ Systemen ist einfacher möglich und die Modifikation bestehender B2B-Kommunikation wird auf ein Mindestmaß reduziert. Es ergibt sich außerdem ein hohes Maß an Flexibilität in der Handhabung und Gestaltung der einzelnen Nachrichten.

Bei Realisierung dieser Variante kommt es zur in Abbildung 5.3 dargestellten Struktur und Zusammenspiel bei Verhandlungs- und B2B-Nachrichten.

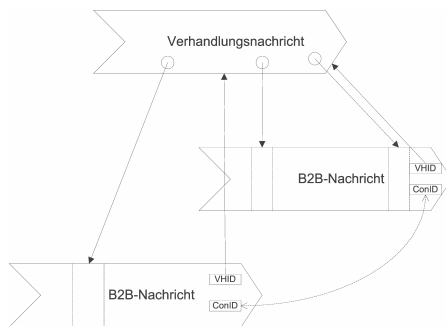


Abbildung 5.3: Zusammenhang zwischen Verhandlungs- und B2B-Nachrichten

Die Verhandlungsnachrichten können dabei Referenzen auf B2B-Nachrichten enthalten. Diese Referenzen werden durch das EPN-System unabhängig von der Syntax auf konzeptueller Ebene (= Ontologie) gehandhabt. Die Referenzen spezifizieren Abschnitte in den B2B-

Nachrichten, durch die bestimmte für die Verhandlung relevante Konzepte in den Geschäftsdokumenten, wie beispielsweise die Farbe oder der Preis eines Produktes beschrieben werden.

Jede B2B-Nachricht enthält eine eigene Identifikation, die ‚Verhandlungs-Identifikation (VHID)‘. Durch diese VHID können die B2B-Nachrichten beim Empfänger wieder der zugehörigen aber getrennt übertragenen Verhandlungsnachricht zugeordnet werden. Geht eine B2B-Nachricht von einem klassischen EC-System ohne VHID ein, so muss der MTM die Identifikation übernehmen, indem er die Antwort anhand der Absender-Adresse der ursprünglichen Anfrage zuordnet.

Manche B2B-Standards enthalten bereits in eigener Form eine ‚Conversation-Identifikation (ConID)‘ im Nachrichtenformat, die es ermöglicht mehrere B2B-Nachrichten zu einer gesamten B2B-Konversation zuzuordnen. So eine Konversation kann beispielsweise ein Bestellvorgang sein, der mit Einzelnachrichten wie einer Produktnachfrage startet und mit einer Bestellung endet. Bei mehreren gleichzeitigen Kommunikationen mit anderen Systemen müssen die Nachrichten exakt zugeordnet werden können. Teils könnte diese Funktionalität von der VHID übernommen werden. Handelt es sich jedoch um eine Interaktion mit einem klassischen EC-System fehlt diese Identifikation über die VHID.

Die Gesamtarchitektur des EPN-Systems wird in Kapitel 6 vorgestellt. Im folgenden werden jedoch schon einige Elemente der Architektur im Zusammenhang mit der Behandlung von aus- und eingehenden Nachrichten im System zum Teil vorgestellt.

Abbildung 5.4 zeigt die Behandlung *ausgehender* Nachrichten im EPN-System.

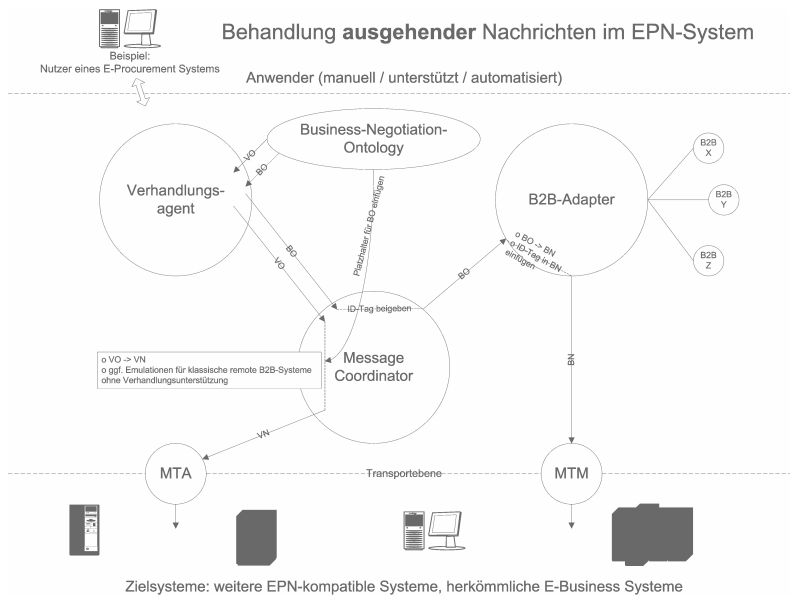


Abbildung 5.4: Ausgehende Nachrichten im EPN-System

Nachdem durch einen Anwender der Verhandlungsprozess initiiert und die konkret durchzuführende Verhandlungsaktion angestoßen wurde setzt der Verhandlungsagent die notwendige Gesamtnachricht mit Hilfe der Business-Negotiation-Ontology zusammen. Der Verhandlungsagent reicht die Nachricht mit den Anteilen ‚Verhandlungsontologie (VO)‘ und Business-Ontologie (BO)‘ an den ‚Message Coordinator‘ weiter. Dort wird die VO in den ‚Content‘ einer FIPA-konformen Nachricht konvertiert. Mit Hilfe der Business-Negotiation-Ontology wird ein Platzhalter für den in der Folge heraus getrennten BO-Anteil eingefügt. Dann wird die Basis-Verhandlungsnachricht (VN) über den MTA ans Zielsystem versandt. Handelt es sich beim Zielsystem nicht um ein dem EPN-System kompatibles verhandlungsfähiges System, sondern um ein klassisches B2B-System so werden, bis zum möglichen Grad, über den B2B die Verhandlungsnachricht emulierende B2B-Nachrichten erzeugt. In jedem Fall fügt der ‚Message Coordinator‘ dem BO-Anteil der Verhandlungsnachricht ein Identifikations-Tag (VHID) bei, das beim Empfang wieder die Zuordnung zur Basis-Verhandlungsnachricht ermöglicht. Der BO-Anteil wird dann zum ‚B2B-Adapter‘ überstellt der den BO-Anteil ergänzt durch die VHID in das Nachrichtenformat des B2B-Standards konvertiert/vermittelt, die das Zielsystem versteht (der B2B-Standard wird dabei in der Regel ein Konzept oder einen Mechanismus für eine solche VHID bereits vorsehen). Diese Nachricht wird durch den MTM über das jeweils durch das Zielsystem gewünschte Kommunikationsframework

versandt. Diese Parameter des Zielsystems können beispielsweise über Anfrage an entsprechende Registries ermittelt werden.

Abbildung 5.5 zeigt die Behandlung eingehender Nachrichten im EPN-System.

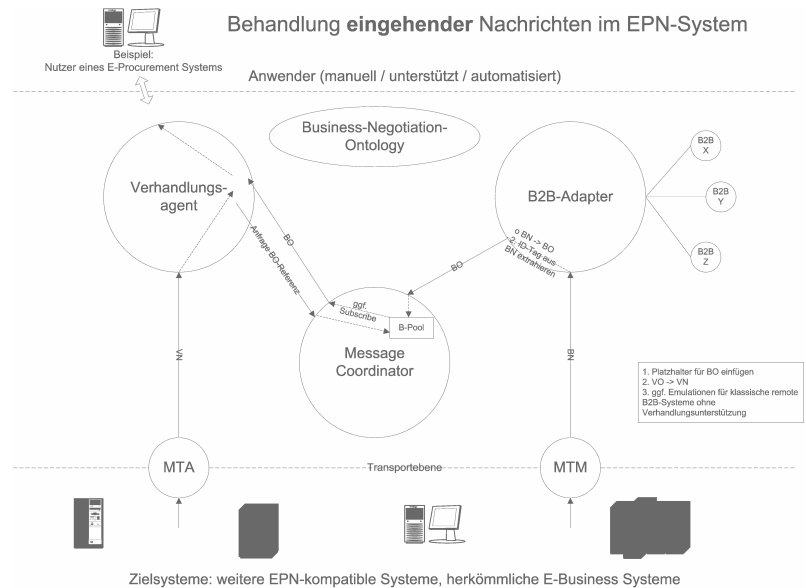


Abbildung 5.5: Eingehende Nachrichten im EPN-System

Geht eine B2B-Nachricht im EPN-System über den MTM ein, so leitet er sie an den B2B-Adapter weiter. Dieser konvertiert die Nachricht in die Ontologiedarstellung BO. Er extrahiert dabei auch explizit die VHID. Diese Informationen werden als BO an den ‚Message Coordinator‘ übertragen, der die eingehenden Nachrichten zunächst in einem Pool speichert (B-Pool).

Wird vom MTM festgestellt, dass es sich bei den Nachrichten um Antworten auf B2B-Nachrichten handelt, die zur Verhandlungssimulation generiert wurden, so wird vom ‚Message Coordinator‘ der ‚Verhandlungsagent‘ benachrichtigt, dem auch alle zugehörigen weiteren Nachrichten weitergeleitet werden.

Geht nun eine Verhandlungsnachricht über den MTA ein, so wird sie an den Verhandlungsagenten weitergeleitet. Sind Referenzen auf B2B-Anteile enthalten, so fragt der Verhandlungsagent diese beim ‚Message Coordinator‘ nach. Sind sie im B-Pool, so reicht der ‚Message Coordinator‘ diese sofort weiter, ansonsten werden sie bei Eintreffen an den Verhandlungsagenten weitergeleitet.

Verfügt der Verhandlungsagent über alle notwendigen Informationen, kann er sie an den Anwender weitergeben und die Verhandlungsaktion wurde erfolgreich abgeschlossen.

5.6 Verhandelbare Verhandlungsprotokolle

Wenn Verhandlungspartner dazu bereit sind, ihre anstehende Verhandlung konkreten und exakten Regeln zu unterwerfen, können sie ein Verhandlungsprotokoll vereinbaren. Ein klassisches Beispiel sind dafür die vielfältigen Auktionsprotokolle (Reck, 1997).

Die Vereinbarung von bestimmten Protokollen kann Auswirkungen auf den Ausgang der Verhandlung haben, was seit je her zu besonderen Interesse am Design von Verhandlungsprotokollen führt (Milgrom, 1989). Diese Auswirkungen sind jedoch komplex und nur schwer oder teils gar nicht bestimmbar.

Wichtig ist jedoch auch, dass das gewählte Protokoll zu den Zielen der Verhandlungspartner in ihrer Verhandlungssituation passt. „There is not a single best negotiation protocol for all negotiation situations“ ((Bichler, 2001), Seite 82). Beispielsweise kann es für die Stadtwerke Tüchtigburg (siehe Fallbeispiel in Abschnitt 4.1.3) wichtig sein, in Sekundenschnelle ein Ergebnis der Verhandlung zu erreichen. Dazu muss das richtige Protokoll gewählt werden.

Die Wahl des gemeinsamen Protokolls kann nicht immer nur durch eine Seite geschehen. Dadurch kann die Bereitschaft von potentiell an einer Verhandlung interessierten Akteuren so weit reduziert werden, dass sie jegliche weitere Verhandlung ablehnen und somit für keinen der Beteiligten Vorteile realisiert werden können.

Um allen potentiellen Verhandlungsakteuren eine Einflussnahme auf das vereinbarte und verwendete Protokoll zu ermöglichen, wurde im Kontext dieser Arbeit das so genannte ‚Bootstrapping-Konzept‘ neu entwickelt und ein Vorschlag einer möglichen Realisierung erarbeitet (Knollmann, 2003).

Das Verfahren wird beispielhaft im folgenden Fallbeispiel dargestellt.

5.6.1 Fallbeispiel für Bootstrapping

Das weiter oben bereits eingeführte Fallbeispiel der Stadtwerke Tüchtigburg soll nun auch als Beispiel für das Bootstrapping dienen. Abbildung 5.6 zeigt den entsprechenden Verlauf eines Fallbeispiels und soll auch

durch die ODER-Verzweigungen den generellen Ablauf noch einmal verdeutlichen.

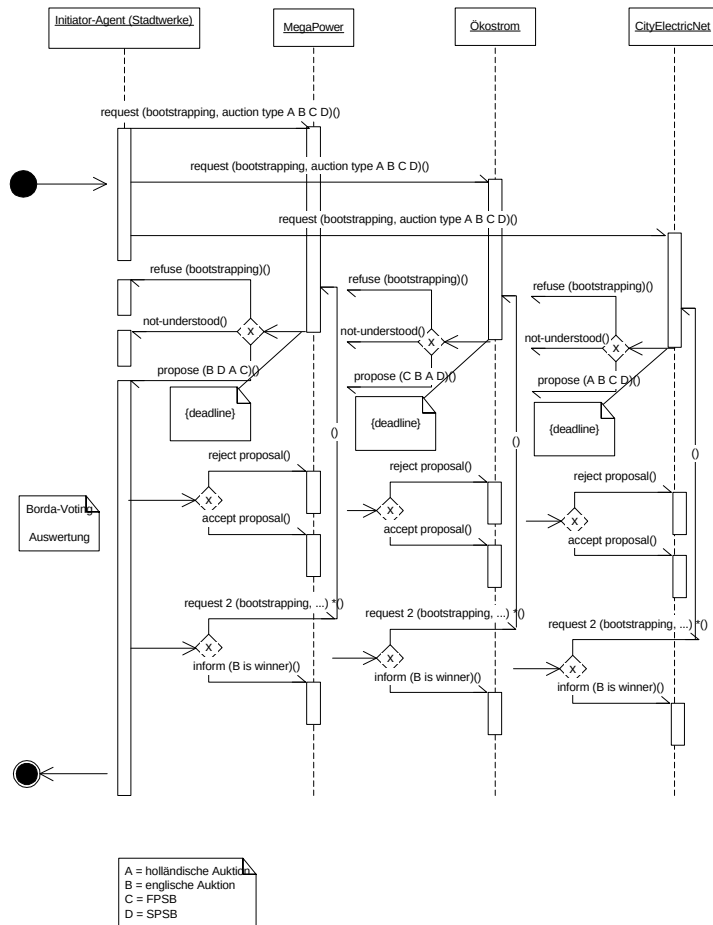


Abbildung 5.6: Beispiel für Bootstrapping am Fallbeispiel Stadtwerke Tübingen

Die vier Bootstrapping-Teilnehmer haben sich vor Beginn über das Lobbymodell erkannt. Dann sendet zunächst der Initiator-Agent, hier die Stadtwerke Tübingen, einen ‚request‘ mit Angabe der möglichen Auktionstypen an alle potentiellen Bieter. Diese geben ihr Voting ab, welches bestätigt wird. Das hier nicht ersichtliche eigene Voting der Stadtwerke Tübingen ist (D B C A). Die Stadtwerke werten das Voting aus und informieren die Teilnehmer über das Ergebnis B des Voting. Es wird nun also eine englische Auktion als explizites Verhandlungsprotokoll verwendet.

Im Falle von ‚rejects‘ seitens der Teilnehmer oder eines nicht eindeutigen Wahlergebnisses kann sich der Initiator-Agent für eine Wiederholung der Borda-Wahl entscheiden und erneute ‚requests‘ senden (siehe * in der Abbildung).

6 Implementation verhandlungs-spezifischer Infrastrukturen und Systeme

Die in den vorigen Kapiteln erarbeiteten Konzepte und Lösungen für Verhandlungen im Electronic Business in Form spezifischer Infrastrukturen und Systeme erfordern adäquate Technologien, die eine effiziente Implementation erlauben.

Die Agententechnologie bietet mit der Autonomie der Agenten und dem sprechaktbasierten Kommunikationsparadigma entsprechende Voraussetzungen. Ebenso wie die Web-Service-Technologie als ein Element zur Verbesserung von Interoperabilität in offenen verteilten Systemen des E-Business.

Diese Technologien werden verwendet, um eine Infrastruktur für Verhandlungen im Electronic Business und wieder verwendbare Systemkomponenten für die Durchführung von Verhandlungen zu implementieren.

Auf dieser Basis können in Systeme des Electronic Business weitreichende Verhandlungsfunktionalitäten in einer offenen verteilten Umgebung integriert werden. Um dies beispielhaft im Bereich der elektronischen Beschaffung zu zeigen, wurde im Kontext dieser Arbeit ein E-Procurement-System prototypisch implementiert (Böge, 2003)..

6.1 Wichtige Technologien für die Realisierung verhandlungsspezifischer Systeme

Die für E-Business-Anwendungen wichtigen Realisierungstechnologien wurden zum Abschluss von Abschnitt 2.1 über die Charakteristika des E-Business vorgestellt. Für die Realisierung verhandlungsspezifischer Systeme sind weitere Technologien spezifisch.

Die *Agententechnologie* ist einerseits für Verhandlungsanwendungen geeignet, da sie die Autonomie der Agenten / der verhandelnden Entitäten beachtet. Andererseits bietet sie mit ihrem sprechaktbasierten Kommunikationsparadigma gute Voraussetzungen für Verhandlungskommunikation.

Elektronische Verhandlungen finden zwischen verschiedensten Personen und Organisationen statt, die meist über eine ebenso verschieden gestaltete IT-Landschaft verfügen. Damit treten vielfältige Heterogenitäten auf und es sind enorme Anforderungen im Bereich der Interoperabilität der Systeme zu bewältigen. Einen wichtigen technischen Beitrag dazu können *Web-Services* im Middleware-Bereich liefern.

6.1.1 Agententechnologie

Zunächst wird der Begriff „Agent“ näher umrissen, was vor allem bedeutet, sich mit den Eigenschaften zu beschäftigen, die gemeinhin mit Agenten verbunden werden. Darauf aufbauend wird das Konzept der Multiagentensysteme zusammengefasst und anschließend die Spezifikationen der FIPA für die Umsetzung solcher Systeme vorgestellt. Darauf basierende konkrete Agentenplattformen sind beispielsweise FIPA-OS oder JADE, die beide intensiv in dieser Arbeit evaluiert wurden.

6.1.1.1 Der Agentenbegriff

Der Begriff „Agent“ erweist sich als äußerst schwer zu fassen. Bis heute gibt es trotz zahlloser Versuche keine klare und allgemein akzeptierte Definition. Eine solche zu finden, wird schon durch den sehr weitläufigen Gebrauch des Begriffes erschwert, der teils unbedacht für sehr verschiedene Dinge verwendet wird, von einfachen Regelsystemen über Benutzerdialoge mit textuellen Hilfestellungen bis hin zu komplexen, wissensbasierten Systemen. Einige der Definitionsversuche erwecken den Eindruck, daß sich ihre Urheber stark an spezifischen Beispielen oder Projekten orientieren, die sie dabei im Hinterkopf haben, was zu recht unterschiedlichen Gewichtungen führt. Einige Beispiele:

„An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.“

(Russel, 1995)

„Let us define an agent as a persistent software entity dedicated to a specific purpose. ‘Persistent’ distinguishes agents from subroutines; agents have their own ideas about how to accomplish tasks, their own agendas. ‘Special purpose’ distinguishes them from entire multifunction

applications; agents are typically much smaller.“
(Smith, et al., 1994)

Auf der anderen Seite gibt es durchaus Versuche, solche Definitionen zu sammeln, sie auf ihr Wesentliches zu reduzieren und Gemeinsamkeiten herauszustellen, um so zu einer übergreifenden Agentendefinition zu gelangen. Aus solchen Sammlungen resultieren teilweise jedoch recht weit gefaßte Beschreibungen und Listen von möglichen Eigenschaften, die Agenten haben können. Die Object Management Group (OMG, 2000) definiert ihre Sicht auf Agenten wie folgt:

„The basic dictionary definition of agent is one who acts. However, for developing IT systems, such a definition is too general: IT-related agents need additional properties.“

Im offensichtlichen Bestreben, es wirklich jedem recht zu machen, folgt eine Liste von insgesamt 18 sehr unterschiedlichen, grob nach ihrer Bedeutung sortierten möglichen Eigenschaften, die beschrieben werden und die in unterschiedlicher Kombination einen Agenten ausmachen können. Immerhin werden dabei drei Eigenschaften für unverzichtbar erachtet: Autonomie, Interaktivität und Adaptivität. Viele andere – darunter Mobilität und Intelligenz – sind eher spezifisch für bestimmte Domänen, und in Abhängigkeiten von diesen wird ihnen entsprechende Bedeutung beigemessen.

Nahezu der einzige Punkt, in dem sich alle Quellen einig sind, ist der, dass Agenten die Fähigkeit besitzen, autonom zu handeln. Diesen Umstand nutzt Michael Wooldridge, um Agenten von anderen Systemen abzugrenzen (Wooldridge, 1999). Er betrachtet Agenten als Computer-Systeme, die in einer bestimmten Umgebung platziert sind und die in der Lage sind, in dieser Umgebung autonom zu handeln, um die Ziele zu erreichen, für die sie entworfen wurden. Autonomie bedeutet in diesem Zusammenhang, dass Agenten ohne Einwirkung von Menschen oder anderen Systemen handeln können, dass sie sowohl ihren inneren Zustand als auch ihr Verhalten selbst kontrollieren.

In fast allen Fällen, bis auf die einfachsten, ist die Umgebung des Agenten zu komplex, als daß er sie vollständig kontrollieren könnte. Sie reagiert im wesentlichen nicht-deterministisch auf die Versuche des Agenten, sie zu beeinflussen, und so muss er immer damit rechnen, dass seine Handlungen fehlschlagen können.

Im Gegensatz zu „einfachen“ Agenten – zu denen man auch Systeme zählen kann, die man im alltäglichen Gebrauch nicht als Agenten bezeichnen würde, wie etwa Steuerungssysteme oder Software-Daemons – zeichnen sich „intelligente“ Agenten durch flexibles autonomes Handeln aus. Die Flexibilität äußert sich für Wooldridge in der angemessenen Kombination dreier Eigenschaften des Agenten: Reaktivität, Proaktivität

und soziale Fähigkeit. Reaktivität heißt für den Agenten, seine Umgebung wahrzunehmen und auf Veränderungen in angemessener Weise zu reagieren. Proaktiv ist er, wenn er selbst die Initiative zu zielgerichteten Handlungen ergreift. Diese beiden Komponenten stehen in einem gewissen Widerspruch zueinander, und die Herausforderung für den Agenten besteht darin, sie in Balance zu halten. Die soziale Fähigkeit bezeichnet über den simplen Datenaustausch „normaler“ Systeme hinausgehende Kommunikationsfähigkeit, also z.B. die Möglichkeit, mit anderen Agenten zu verhandeln oder zu kooperieren.

Insgesamt scheinen es nicht bestimmte Eigenschaften zu sein, die durch ihr Vorhandensein oder durch ihre Kombination das Wesen eines Agenten ausmachen, sondern vielmehr die Qualität einzelner Eigenschaften. Alle Merkmale, die üblicher Weise zitiert werden, um Agenten zu charakterisieren, finden sich in der einen oder anderen Form auch in Systemen, die man intuitiv nicht unbedingt zu dieser Gruppe zählen würde. Auch autonomes Verhalten ist erst ein hinreichendes Kriterium, wenn die Anforderungen an dies Verhalten merklich über die hinausgehen, die man an herkömmliche Systeme stellen würde. Das führt zwar auch dazu, dass die Abgrenzung von Agenten wiederum etwas unscharf wirkt, aber das lässt sich anscheinend nicht völlig vermeiden: Wie Franklin et al. anmerken (Franklin, et al., 1996), „leben“ Agenten eher in einer realen Welt, als in einer theoretischen, und in der realen Welt ergeben sich nur unscharfe Kategorien.

6.1.1.2 Multiagentensysteme

Auch wenn es Situationen gibt, in denen ein einzelner Agent auf sich allein gestellt sinnvolle Arbeit verrichten kann, so sind die Möglichkeiten eines solch isolierten Systems doch recht eingeschränkt. Zudem werden solche Situationen in anbetracht der schnell voranschreitenden Verbindung und Vernetzung von Computer-Systemen zunehmend seltener. Besonders in Fällen, in denen zahlreiche unterschiedliche Agenten interagieren sollen, ist es zweckmäßig, diese nicht ausschließlich individuell zu betrachten, sondern auch kollektiv, gewissermaßen als eine Gemeinschaft von Agenten (Huhns, et al., 1999). Für Systeme, die solchen Agentengemeinschaften zu diesem Zweck zumindest teilweise eine gemeinsame Umgebung bereitstellen, hat sich der Begriff der Multiagentensysteme (MAS) etabliert.

Die Wurzeln der MAS liegen im Forschungsbereich der verteilten künstlichen Intelligenz, parallel zur ebenfalls aus dieser Richtung stammenden verteilten Problemlösung (Distributed Problem Solving). Letzteres befasst sich mit der Frage, wie ein bestimmtes Problem durch mehrere kooperierende Module gelöst werden kann, die das zur Lösung nötige Wissen

untereinander aufteilen und sich gegenseitig bereitstellen. Die hierfür nötigen Interaktionsstrategien sind integraler Bestandteil des Systems. Im Gegensatz dazu ist die Forschung an MAS eher am gemeinschaftlichen Verhalten möglicherweise schon bereits existierender, autonomer Agenten interessiert, die das Ziel haben, ein gegebenes Problem zu lösen. Oft wird der Begriff MAS allerdings auch weiter gefaßt, und bezeichnet dann alle Formen von Systemen, die aus mehreren (semi-) autonomen Komponenten bestehen. Zu den typischen Merkmalen von MAS zählen folgende (Jennings, et al., 1998) :

- Jedem Agenten steht nur ein Teil der für die Problemlösung notwendigen Informationen oder Fähigkeiten zur Verfügung, d.h. er hat eine beschränkte Sicht auf das Gesamtsystem
- Es gibt keine globale Kontrolle im System
- Daten werden dezentral gehalten
- Die Berechnung erfolgt asynchron

In Kombination mit den oben beschriebenen Eigenschaften individueller Agenten – vor allem ihrer Fähigkeit, autonom zu handeln – stellen diese Merkmale besondere Anforderungen an die Kommunikation zwischen den Komponenten des Systems. Die Agenten müssen beispielsweise die Möglichkeit haben zu kooperieren, sich zu koordinieren und miteinander zu verhandeln, um zielgerichtet die ihnen übertragenen Aufgaben erfüllen zu können. Der Entwurf von Lösungen, die MAS mit den dafür nötigen Mitteln ausstatten, wird auch als *society design* bezeichnet, ist dabei aber nicht unabhängig vom *agent design*, dem Entwurf einzelner Agenten; beide Bereiche beeinflussen sich gegenseitig (Wooldridge, 2002).

Obwohl Entwurf und Implementation von MAS noch mit weiteren Schwierigkeiten verbunden sind (vgl. (Jennings, et al., 1998)), erhofft man sich, auf ihnen aufbauend robustere und effizientere Systeme zu schaffen, die sich besonders zur Lösung von Problemen eignen, bei denen Daten, Spezialfähigkeiten oder Steuerung in verteilter Form vorliegen und die darüber hinaus in der Lage sind, mit bestehenden „herkömmlichen“ Systemen zu interoperieren.

6.1.1.2.1 Der FIPA-Standard

Zu den frühen Ansätzen, Multiagentensysteme zu untersuchen und umzusetzen, zählen beispielsweise die Arbeiten von Davis und Smith, aus denen das Contract Net Protocol hervorging (Davis, et al., 1983), und auch das „Actors“ Modell (Agha, 1986). Erst deutlich später folgten Versuche, notwendige Eigenschaften von Agenten und ihre Interaktion allgemeiner zu spezifizieren, um so zu einem Standard für Multiagentensysteme zu gelangen. Hierzu gehören die im Rahmen des DARPA Knowledge Sharing Effort (Finin, et al., 1993; Patil, et al., 1992)

entwickelten Spezifikationen für KIF (Knowledge Interchange Format) und KQML (Knowledge Query and Manipulation Language). Etwas später folgte der von der Object Management Group (OMG) herausgegebene MASIF-Standard (Mobile Agent System Interoperability Facility) (Milojicic, et al., 1999; OMG, 1997), der allerdings nicht weitreichend genug war und derzeit in weiterführende Bemühungen der OMG einfließt (OMG, 2000).

Der bisher umfassendste und meistversprechende Standardisierungsversuch, in den auch viele Komponenten aus oben genannten Spezifikationen eingeflossen sind, ist der der FIPA, auf den im folgenden Abschnitt weiter eingegangen wird.

6.1.1.2.2 Die FIPA und ihre Aufgaben

Die „Foundation for Intelligent Physical Agents“ (<http://www.fipa.org/>) (FIPA) ist eine non-profit Organisation, deren Aufgabe nach eigener Auffassung darin besteht, Entwicklung und Erfolg agentenbasierter Anwendungen, Dienstleistungen und Einrichtungen zu fördern. Dieses Ziel verfolgt sie, in dem sie Standards für die Interoperation zwischen heterogenen Software-Agenten entwickelt. Zu den über 60 Mitgliedern der FIPA gehören Unternehmen aus der Computer- und aus der Telekommunikationsbranche sowie Universitäten aus verschiedenen Ländern, die sich aktiv an der Entwicklung beteiligen.

Seit ihrer Gründung im Jahr 1996 hat die FIPA zahlreiche Standards für alle wichtigen Bereiche der Agenten und der Multiagentensysteme entworfen und veröffentlicht. Trotzdem hat die Arbeit der FIPA bereits großen Anklang gefunden. Es wurden bereits verschiedene Systeme implementiert, die die FIPA-konforme Umsetzung von Agenten ermöglichen, und viele aktuelle Forschungsprojekte aus dem Bereich der MAS bedienen sich dieser.

6.1.1.2.3 Die Spezifikationen im Überblick

Für praktisch alle Bereiche der Multiagentensysteme existieren inzwischen, zum Teil in der Entwicklung befindliche, Standards, die von der FIPA diesen Bereichen entsprechend gegliedert sind. Eine Übersicht hierzu zeigt Abbildung 6.1. Die oberste, die Applikationsebene, enthält Spezifikationen für prototypische Anwendungen auf Basis der Standards und allgemeine Richtlinien für die Implementation von Anwendungen in bestimmten Domänen. Es folgt die Ebene der abstrakten Architektur, die eine gemeinsame Basis in Form verschiedener Dienste für unter dem FIPA-Standard zu implementierende Agenten bereitstellt.

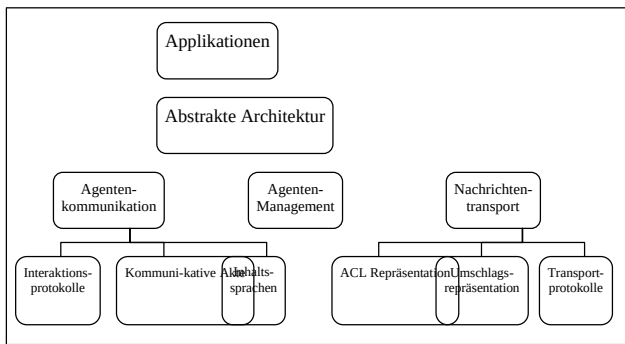


Abbildung 6.1: Thematische Struktur der FIPA-Standards

Darunter liegen Spezifikationen aus drei weiteren Bereichen. Die Agentenkommunikation umfasst unter anderem die kommunikativen Akte, die wie schon angesprochen für die Verhandlungskommunikation besondere Bedeutung haben. Das Agenten-Management definiert Dienste, Ontologien und Rahmenbedingungen, die für den Betrieb eines Agenten nötig sind. Der Bereich Nachrichtentransport schließlich enthält Standards, die die Repräsentation, Kodierung und die Transportmöglichkeiten der von den Agenten generierten Nachrichten betreffen.

6.1.1.2.4 Die abstrakte Architektur

Die FIPA ist bemüht, durch ihre Standards die Zusammenarbeit heterogener Agentensysteme zu fördern und auch ihren Einsatz im kommerziellen Umfeld zu erleichtern. Dazu muss die Architektur der Systeme in der Lage sein, bestehende Software-Umgebungen zu nutzen, und ein möglichst großes Spektrum an etablierten Mechanismen unterstützen. Um dies zu erreichen, wurden zunächst die Elemente zusammengestellt, die spezifiziert werden müssen, um eine funktionsfähige Agentenplattform zu erhalten. Dann wurden ggf. bereits bestehende Systeme untersucht, die einige dieser Elemente bereitstellen und die in der Regel unterschiedliche Techniken einsetzen, um diese zu realisieren. Anschließend galt es, die gemeinsamen Merkmale der verschiedenen Herangehensweisen zu identifizieren und so zu einer Abstraktion der Gesamtarchitektur zu gelangen.

Die FIPA Abstract Architecture (FIPA, 2002g) bildet den konzeptionellen Kern der FIPA-Spezifikationen und umfasst folgende Bereiche:

- Ein Dienst- und Dienstauffindungsmodell für die Agenten
- Interoperabilität des Nachrichtentransports
- Unterstützung für verschiedene Formen der ACL-Repräsentationen
- Unterstützung für verschiedene Inhaltssprachen
- Unterstützung für mehrere Verzeichnisdienstrepräsentationen

Über einige andere wichtige Bereiche – darunter die Lebenszyklen der Agenten und ihre Handhabung, Mobilität und Konversationsrichtlinien, die eigentlich Teil dieses Spezifikationskerns sein sollten – besteht bisher keine hinreichende Einigkeit. Sie bleiben daher vorerst Teil der peripheren Spezifikationen.

Die FIPA bezeichnet ihre Architektur auch deshalb als abstrakt, weil sie sich nicht direkt implementieren lässt. Sie bildet vielmehr die Grundlage für die Entwicklung konkreter Architekturen, in denen dann erst detailliert festgelegt ist, wie genau die einzelnen Bestandteile eines Agentensystems umzusetzen sind, welche Programmiersprachen, Schnittstellen, Netzwerkprotokolle, Betriebssysteme usw. zum Einsatz kommen. Der obligatorische Teil der FIPA-Spezifikationen garantiert eine minimale gemeinsame Basis für alle konformen Architekturen. Obwohl dem Entwickler in allen Bereichen große Freiheit gelassen wird, soll dies die Interoperation verschiedener, konkreter, FIPA-konformer Agentenarchitekturen ermöglichen.

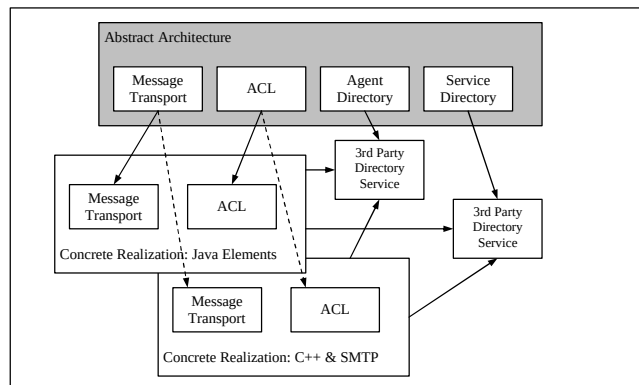


Abbildung 6.2: Übergänge von der abstrakten zu konkreten Architekturen

Abbildung 6.2 verdeutlicht mögliche Übergänge von der abstrakten zu konkreten Architekturen. Nicht alle obligatorischen Elemente müssen in jeder konkreten Architektur direkt vorgesehen werden. Es ist durchaus möglich, einzelne Bestandteile losgelöst vom Gesamtsystem zu beschreiben, um ihre gemeinsame Nutzung durch unterschiedliche konkrete Architekturen zu ermöglichen.

6.1.2 Systemtechnische Interoperabilität mit Web-Services

Der Bereich der Verhandlungssysteme im E-Business stellt besonders intensive Anforderungen an die Interoperabilität der Systeme, da es sich um ein ausgesprochen heterogenes und dynamisch veränderliches Anwendungsfeld handelt.

Die Beziehungen zwischen den Geschäfts- und Verhandlungspartnern verändern sich oft dynamisch und können ad hoc zustande kommen. So müssen unter diesen Umständen möglichst interoperable Kommunikationsmöglichkeiten zwischen den Verhandlungspartnern geschaffen werden.

Da die Verhandlungsphase ein Teil der Kauf- und Beschaffungsprozesse ist, sind meist Verbindungen zu vielen anderen internen und externen Systemkomponenten notwendig. Intern kann der Zugriff auf Produktions- und Lagerhaltungssysteme, das ERP-System mit Rechnungswesen, Datenbanken und Datawarehouses oder Marketingsysteme nötig sein. Externe Informationen können für Verhandlungen sehr wichtig sein. Zunächst müssen die richtigen Verhandlungspartner gefunden werden, Zahlungen werden extern abgewickelt, Informationen über Kreditwürdigkeit können für ein Unternehmen lebenswichtig sein. Auch externe Partner können für Outsourcing, wie beispielsweise im Bereich der Logistik und der Auslieferung des Öfteren üblich, einbezogen werden.

Um diesen Anforderungen zu entsprechen, bedarf es dem Einsatz innovativer Interoperabilitätstechnologien, wie beispielsweise dem Konzept der Web-Services, die in der Folge vorgestellt werden.

6.1.2.1 Web-Services als Interoperabilitätstechnologie

Die Veränderung des Internets von statischen Inhalten hin zu einem dynamischen Medium für verteilte Dienste und Anwendungen führt zu neuen Anforderungen an die Interoperabilität von Systemen. Klassische Middleware-Technologien, können diese Anforderungen, insbesondere an die nicht vorher explizit geplante und automatische Kopplung von Systemen, nicht mehr in ausreichender Form erfüllen.

Die Entwicklung solch klassischer Lösungen ist sowohl in Form von großen Middleware-Systemen als Standardlösung als auch in der Individualentwicklung sehr kostenintensiv. Die Notwendigkeit einer klar definierten Schnittstellentechnik für verteilte Dienste und Anwendungen ist klar ersichtlich und wird durch das Konzept der Web-Services adressiert.

Das besondere an Web-Services ist, dass sie sich und ihre Schnittstellen selbst beschreiben. Demgemäß bedarf es nicht mehr proprietären Middleware-Lösungen. Die großen Softwarehersteller unterstützen die Spezifi-

kationen der Webservices und garantieren so ein größtmögliches Maß an Kompatibilität der einzelnen Dienste im Internet.

Dies ist von besonderer Wichtigkeit wenn autonome Akteure im E-Business die Möglichkeit gegeben werden soll, ad hoc zu interagieren und zu verhandeln.

Web-Services lassen sich durch die folgenden Eigenschaften charakterisieren (Bettag, 2001):

Web-Services sind ...

- *programmierbar*: Web-Services sind über programmierbare Schnittstellen erreichbar. Sie dienen zur Kommunikation zwischen Anwendungen und haben keine eigene graphische Oberfläche.
- *selbstbeschreibend*: Die Beschreibung eines Dienstes und seiner Schnittstellen wird über WSDL (Web Service Description Language) realisiert. Darin sind Informationen bezüglich der Dienststart, Name, Beschreibung etc. enthalten.
- *gekapselt*: Nur die Schnittstellen eines Dienstes werden veröffentlicht. Eine genaue Beschreibung über die Implementation gibt es nicht. Dadurch ist eine Kompatibilität zu neueren Versionen gegeben.
- *lose gekoppelt*: Die Kommunikation zwischen den Dienstpartnern erfolgt über den Austausch von Nachrichten und nicht über eine direkte Verbindung.
- *ortstransparent*: Die Dienste sind nicht ortsgebunden und von allen Teilen des Internets aus abrufbar, Zugriffsrechte vorausgesetzt.
- *transparent bezüglich des Transportprotokolls*: Die Kommunikation kann über verschiedene Transportprotokolle erfolgen. Mögliche Varianten sind http und SMTP.
- *komponierbar*: Web-Services können aus mehreren einzelnen Diensten zusammengesetzt werden, um so ein breiteres Angebot bereit zu stellen. Andererseits können sich mehrere Web-Services dem Angebot eines einzelnen bedienen.
- Web-Services ermöglichen technologisch eine vereinfachte Entwicklung der Kommunikation zwischen Diensten. Sie wird im Idealfall sogar automatisiert. Die Suche nach bestimmten Diensten und ihren Anbietern wird durch WSDL und UDDI (Universal Description, Discovery and Integration) weitestgehend selbstständig erledigt. Jedoch kann die Entscheidung und die Gewichtung der Kriterien, die einzelne Anbieter charakterisieren, nicht automatisiert werden. Sie muss vom Entwickler getätigt werden. Lediglich durch den Einsatz von Algorithmen der

KI (Künstlichen Intelligenz) kann eine Entscheidung übernommen werden.

Im Folgenden werden die wichtigsten technologischen Elemente von Web-Services dargestellt (vgl. (Ayala, et al., 2002; Schmelzer, et al., 2002; Snell, et al., 2002)). Dies sind SOAP, WSDL und UDDI. Weiterhin wird der praktische Einsatz von Web-Services im E-Business erörtert.

6.1.2.1.1 SOAP

Das Protokoll SOAP („Simple Object Access Protocol“) (siehe <http://www.w3.org/TR/SOAP/>) unterstützt die Kommunikation zwischen einzelnen Diensten. Da es auf der XML-Technologie beruht und multiprotokollfähig ist, kann es durch unterschiedliche Transportprotokolle übertragen werden. Das am meisten verwendete Übertragungsprotokoll ist http. Eine weitere Möglichkeit wäre beispielsweise SMTP. Diese Standardprotokolle bieten bei der Übertragung Vorteile, da sie bewährte Technologien verwenden. Bei der Verwendung von http kann beispielsweise auf die Verlässlichkeit und Erfahrung mit http und die Möglichkeit der Verschlüsselung (SSL) zurückgegriffen werden.

Dass SOAP als Standard einen guten Ausgangspunkt hat, liegt zum einen in der parallelen Entwicklungsunterstützung durch IBM und Microsoft und zum anderen in der Akzeptanz durch viele der großen Softwareunternehmen. SOAP liegt derzeit als Standard in der Version 1.2 vor.

Ein einzelnes SOAP-Paket besteht aus einem ‚Envelope‘, einem optionalen ‚Header‘ und einem ‚Body‘. Der ‚Envelope‘ enthält Angaben über das Kodierung („Encoding“) eines SOAP Paketes sowie über die Namensräume („Namespaces“). Der ‚Envelope‘ schließt den ‚Header‘ und den ‚Body‘ wie ein Umschlag ein. Der ‚Header‘ ist optional und kann Informationen zur Authentifizierung, zu Transaktionen, Bezahlungen usw. enthalten. Der ‚Body‘ stellt die eigentlichen Nutzdaten („Payload“) eines SOAP-Paketes dar.

Die folgende Abbildung 6.3 verdeutlicht den Aufbau eines SOAP-Paketes:

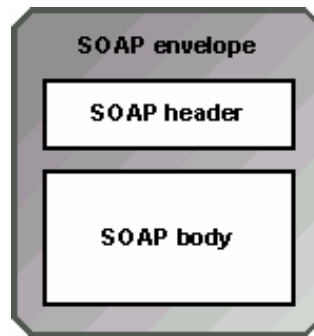


Abbildung 6.3: Aufbau einer SOAP-Nachricht (Oracle, 2002)

Ein wichtiger Aspekt in SOAP sind die Namensräume („Namespaces“). Sie definieren einen Bereich, in dem Bezeichnungen gültig sind. Namensräume sind mit lokalen Variablen von Programmiersprachen wie C++ vergleichbar. Durch die Verwendung von Namensräume stellt man sicher, dass bestimmte Werte auch richtig interpretiert werden, da sie nur im Kontext mit diesem gültig sind.

Im Body der SOAP Nachricht findet man, wie in XML üblich, die Daten mit ihrer dazugehörigen Beschreibung und ihres Typs. Die einzelnen Datentypen können einfach sein, wie z.B. String, Integer. Aber auch komplexe und neu definierte (zusammengesetzte) Typen, wie z.B. „Arrays“ oder „Structs“, finden im „Body“ eines SOAP-Paketes Platz. Ein besonderes Element in einer SOAP-Nachricht ist das Fehlerelement, das sich aus dem Fehlercode, der Beschreibung sowie einer detaillierten Angabe über den Fehler zusammensetzt.

6.1.2.1.2 WSDL

Die Beschreibung der Funktion sowie der Schnittstellen eines Web-Services wird durch die Sprache WSDL („Web Service Description Language“) (siehe <http://www.w3c.org/TR/wsdl>) realisiert. Auch WSDL basiert auf XML und ist so für den Menschen lesbar. Alle Methoden, die ein Web-Service bereitstellt und vor allem wie er dies realisiert, werden hier definiert.

Diese Beschreibung der Dienste und ihrer Schnittstellen führt dazu, dass eigene Anwendungen leichter für die Verbindung zu einem bestimmten Web-Service angepasst werden können. Der nächstmögliche Schritt ist die automatisierte Auswertung des WSDL-Dokumentes und die selbstständige und dynamische Anpassung der Anwendung zur Laufzeit.

Die Darstellung der Eigenschaften in WSDL ist zwar für den Menschen lesbar und theoretisch auch selbst generierbar, doch gerade bei komplizierteren Diensten mit hohem Aufwand verbunden. So gibt es von verschiedenen Herstellern Programme, die automatisch aus vorhandenen

Programmen das dazugehörige WSDL Dokument erzeugen. Zu nennen wäre hier das ‚Web Services Toolkit‘ von IBM sowie der ‚JDeveloper‘ von Oracle. Die automatisch generierten WSDL Dokumente enthalten dann die Beschreibungen der öffentlichen Methoden der zugrunde liegenden Programme.

Ein WSDL Dokument besteht aus folgenden Elementen:

- *definitions*: Die Definitionen geben die Namespaces für den Web-service an.
- *types*: Hier werden die Typen deklariert, normalerweise im „XML schema definition“ – Format (XSD)
- *message*: Eine abstrakte Beschreibung der Inhalte der Nachrichten
- *operation*: Die Beschreibung des Dienstes, den der Webservice anbietet.
- *portType*: Angaben über die Operationen eines oder mehrerer Endpunkte.
- *binding*: Spezifiziert das Protokoll und das Datenformat eines portTypes.
- *port*: Adresse eines Kommunikationsendpunktes (URL).
- *service*: Zusammenfassung von mehreren in Beziehung stehenden Kommunikationsendpunkten.

Die einzelnen Elemente bauen hierarchisch aufeinander auf. Abbildung 6.4 verdeutlicht den Aufbau eines WSDL-Dokumentes:

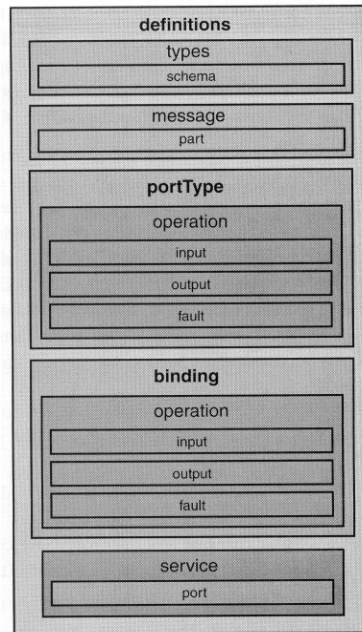


Abbildung 6.4: Struktur eines WSDL-Dokumentes (Schmelzer, et al., 2002)

Ein Beispiel für ein WSDL-Dokument findet sich in Anhang A.3 zu den Implementationen. Es beschreibt einen verhandlungsfähigen Web-Service der im Kontext dieser Arbeit entwickelt wurde (Waldau, 2002).

6.1.2.1.3 UDDI

Nach der Implementierung eines Web-Services sowie der Generierung des zugehörigen WSDL-Dokumentes fehlt noch die Bekanntmachung im Internet. Diesen Vorgang übernimmt die Datenbank UDDI („Universal Description, Discovery and Integration“) (siehe <http://www.uddi.org/>). Hier werden die Einträge über die angebotenen Web-Services gespeichert und sind so für jeden Dienstsuchenden abrufbar. UDDI kann als Verzeichnisdienst bezeichnet werden.

Die Spezifikation von UDDI ist so allgemein gehalten, dass jeder nur erdenkliche Dienst in das Schema von UDDI passt. So ist die Verwaltung der Beschreibungen von Web-Services nur ein Teil der Aufgaben von UDDI, für die Zukunft aber ein bedeutender. Theoretisch kann ein Dienst, der in der UDDI Datenbank beschrieben ist, auch eine telefonische Auskunft (Callcenter) sein, hier sind den Möglichkeiten keine Grenzen gesetzt.

Die UDDI setzt sich aus drei verschiedenen Verzeichnissen zusammen (vgl. (Kuschke, et al., 2001)):

- ‚white pages‘: Hier findet man die allgemeinen Informationen über den Dienstanbieter (Firmenname, Kontaktperson, Adresse, Beschreibung sowie Informationen wie z.B. Handelsregisternummer)
- ‚yellow pages‘: Dies ist eine kategorische Einteilung des Unternehmens (Art der Produkte, geographische Lage)
- ‚green pages‘: In den ‚green pages‘ findet man die technische Beschreibung zu den einzelnen Diensten sowie des Zugriffs auf diese Dienste.

UDDI Datenbanken sollen weltweit Informationen über die angebotenen Dienste kostenlos zur Verfügung stellen. Zurzeit sind nur einige Server verfügbar, die eine Verwaltung von UDDI Einträgen übernehmen. Viele befinden sich in einer Testphase. Eine Implementierung von UDDI gibt es unter dem Namen UDDI4J (UDDI für Java) von IBM, die auf den Webseiten von IBM zu finden ist.

6.1.2.1.4 Web-Services im praktischen Einsatz

Um die Zusammenarbeit von Web-Services und E-Business Anwendungen darzustellen sei folgendes Szenario gegeben:

Eine Anwendung stellt unter Zuhilfenahme verschiedener Anbieter, die durch Web-Services repräsentiert werden, eine Komplettreise zusammen. Der Benutzer gibt dem Programm die gewünschten Vorgaben wie Reisezeit, Ziel, gewünschte Unterbringung oder eventueller Mietwagen. Über die UDDI-Datenbank sucht das Programm nun Anbieter verschiedener Hotels, Fluglinien, Autovermietungen und nimmt mit den Servern Kontakt auf. Das Besondere ist, dass man viel mehr Anbieter erreicht als über proprietäre Verfahren, da vorher keine Absprachen über die Art der Kommunikation zwischen dem Programm und den Servern getroffen werden mussten. Alle nötigen Informationen erhält die Anwendung von der UDDI Datenbank. Nach dem Finden der besten Anbieter nach den vorgegebenen Kriterien, die mit den Einträgen in der UDDI-Datenbank verglichen werden, kann das Programm dann je nach Konfiguration eine Empfehlung an den Benutzer zurückgeben (vgl. (Kuschke, et al., 2001)). Die UDDI-Datenbank kann ebenfalls Informationen zur Verhandlungsfähigkeit einzelner Anbieter enthalten. So kann die Anwendung bereits automatische Teilverhandlungen führen oder eine manuelle Verhandlung vorbereiten.

Allgemein lässt sich der Ablauf von Dienstangebot und –nutzung wie folgt erklären (siehe Abbildung 6.5):



Abbildung 6.5: Ablauf Web-Services

In der Skizze hat ein Web-Service Provider einen Dienst implementiert und will diesen nun öffentlich anbieten. Dazu nimmt er mit dem Web-Service-Directory Verbindung auf (1) und trägt seinen Dienst mit Namen, Beschreibung und den technischen Details per WSDL in die Datenbank ein. Damit ist der Dienst registriert und kann abgerufen werden.

Ein Interessent (Web-Service-Client) benötigt einen gewissen Dienst und befragt dazu die UDDI-Datenbank (2), die ihm eine Liste von möglichen Diensten anbietet. Wenn der richtige nach den gegebenen Kriterien ausgesucht wurde, wird die Schnittstelle des Clients mit Hilfe des WSDL-Dokumentes angepasst. Gleichzeitig werden SOAP-Nachrichten an den Web-Service-Provider geschickt (3). Damit ist die Verbindung zwischen Dienstanbieter und –nutzer hergestellt und kann über den weiteren Austausch von SOAP-Nachrichten fortgesetzt werden.

Dieses Vorgehen verwirklicht das in verteilten Systemen bekannte Trader-/„Broker“-Prinzip.

Für die Web-Service basierten Realisierungen in dieser Arbeit wurden verschiedene Werkzeuge evaluiert. Werkzeug-Unterstützung für Web-Services wird sowohl von kommerzieller als auch von Open-Source Seite geboten. Beispielsweise hat der Hersteller Oracle in seinem Application Server und in „Oracle Portal“ bereits verschiedene Ansätze zur Unterstützung der Entwicklung und Ausführung von Web-Services integriert. So sind die Integration des SOAP-Protokolls sowie das Angebot von UDDI-Datenbanken zum Einstellen von Webservices im Application Server verfügbar. Ebenso wird eine automatische Generierung von Benutzungsschnittstellen („Frontends“) für Webservices mit Hilfe des WSDL-Dokuments durch das Portal in Aussicht gestellt (Oracle, 2002).

Neben der serverseitigen Unterstützung durch den Application Server und das Portal bietet Oracle auch die Integration von Webservices in ihre Tools an. Der „JDeveloper“, Oracles Werkzeug zur Erstellung von Java-

Programmen gibt dem Entwickler in seiner neusten Version einige Hilfen zur Integration von Webservices.

Neben der Bereitstellung von verschiedenen Koderahmen für die unterschiedlichsten Javaprogramme stellt der ‚JDeveloper‘ auf Wunsch eine Verbindung zu einem Applicationserver oder SOAP-Server her, um die erstellten Webservices einzubinden.

Die SOAP-Funktionalität wird mit verschiedenen Oracle-SOAP-Klassen realisiert, die dem Entwickler zur Verfügung stehen. Eine große Erleichterung für den Programmierer bietet die Funktion, WSDL-Dokumente automatisch und assistentgesteuert anzufertigen. Dies wird vom ‚JDeveloper‘ ab der Version 9.0.2. unterstützt. Die aufwendige und fehlerträchtige Erstellung per Hand entfällt.

Neben den kommerziellen Angeboten gibt es weiterhin gut dokumentierte „open source“ Lösungen zur Unterstützung von Web-Services. Ein Beispiel ist die SOAP Implementation von Apache, die sehr gut dokumentiert ist und auch ein großes Maß an Unterstützung und Stabilität liefert und daraufhin letztendlich im Rahmen dieser Arbeit praktische Verwendung fand (vgl. (Waldau, 2002))

6.2 Architektur einer Systeminfrastruktur zur Unterstützung von Verhandlungen

Die Systeminfrastruktur setzt sich aus den schon in Kapitel 5 vorgestellten Konzepten und teils vorgestellten Realisierung und den in der Folge präsentierten Komponenten zusammen.

6.2.1 Die Grundarchitektur

In diesem Abschnitt wird beispielhaft der Aufbau eines prototypischen Frameworks für die Unterstützung von Verhandlungen erläutert.

Vorausgesetzt werden Kenntnisse der zum Teil oben schon vorgestellten Agentenplattform JADE, insbesondere bezüglich der Implementation von Agenten und der Funktionsweise der Agent-Behaviours.

Es folgt zunächst ein Überblick über den generellen Aufbau des EPN-Systems und über die Paketstruktur des Frameworks. Dieser Überblick wird abgeschlossen von Hinweisen zum Start des Systems und einer Beschreibung der verfügbaren Kommandozeilenparameter.

Daraufhin werden Protokolle und Richtlinien für die Kommunikation zwischen den Systemkomponenten behandelt. Diese basieren auf der

Systemontologie, deren Aufbau und Weiterentwicklung mit Hilfe von Protégé-2000 und des Plugins JOCEP vorgenommen wird.

Die Beschreibung des Rahmens der Grundarchitektur schließt mit der Vorstellung zentraler Klassen und Schnittstellen des Frameworks sowie deren Funktionsweise.

6.2.1.1 Struktur des EPN-Systems

Das prototypische Framework orientiert sich am konzeptionellen Entwurf des EPN-Systems. Abbildung 6.6 zeigt einen Ausschnitt der schematischen Darstellung des Kernsystems an zwei Systeminstanzen und verdeutlicht die Beziehungen zwischen den zentralen Komponenten.

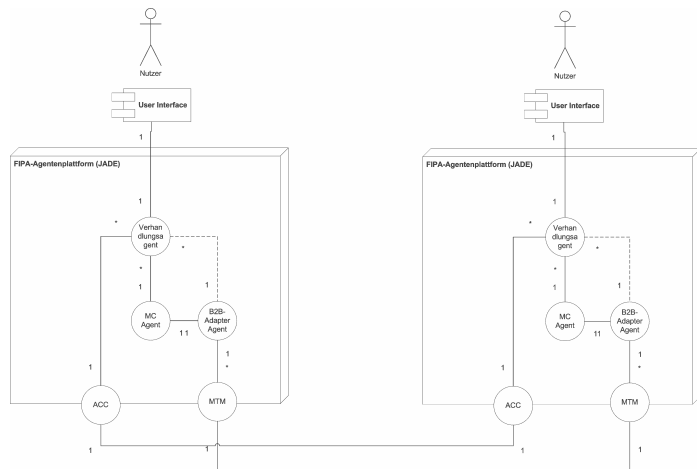


Abbildung 6.6: Komponenten des EPN-Systems

Die Komponenten arbeiten innerhalb einer FIPA-konformen JADE Agentenplattform, d.h. sie sind als autonome Software-Agenten realisiert. Die Plattform ermöglicht die sprechaktbasierte Kommunikation zwischen den Agenten, was sowohl die systeminterne Kommunikation zwischen den Komponenten erleichtert, als auch die Modellierung und Implementation sprechaktbasierter Verhandlungen sowie der zur Kommunikation benötigten Ontologien (vgl. Abschnitt 6.2.1.2).

Die Verhandlungsagenten beinhalten die eigentliche Funktionalität zur Unterstützung und Automatisierung von Verhandlungen und bedienen sich dabei der anderen Komponenten des EPN-Systems. Ihnen stehen zwei unterschiedliche Kommunikationskanäle zur Verfügung: Einer für den Austausch von Verhandlungsnachrichten und ein weiterer für den Versand von Nachrichten, die etablierten E-Business-Standards genügen (z.B. Geschäftsdokumente in ebXML).

Der Agent Communication Channel (ACC) ist Bestandteil der Plattform und ermöglicht den Austausch von Nachrichten zwischen Agenten, auch wenn sich die Kommunikationspartner auf verschiedenen Plattforminstanzen befinden. Die Verhandlungsnachrichten zwischen den Verhandlungsteilnehmern werden auf diesem Wege übertragen.

Nachrichten anderer E-Business-Standards werden zwischen den Teilnehmern durch die B2B-Adapter ihres jeweiligen EPN-Systems vermittelt. Die Adapter haben dabei die Aufgabe, Nachrichten verschiedener Standards zwischen ihrer standardisierten und einer allgemeinen EPN-internen Darstellungsform zu wandeln, deren Struktur auf der gemeinsamen Verhandlungsentologie beruht.

Der eigentliche Versand dieser Nachrichten, der auf die vom jeweiligen Standard festgelegte Weise erfolgen soll, wird vom Message Transport Mediator (MTM) des EPN-Systems übernommen. Dieser stellt dem B2B-Adapter eine einheitliche Schnittstelle für den Versand bereit.

Um die Strukturen der etablierten E-Business-Standards auch innerhalb der EPN-Verhandlungen nutzbar zu machen, ermöglicht das EPN-System darüber hinaus den koordinierten Versand von Verhandlungs- und E-Business-Nachrichten. Hierzu werden letztere zunächst an den Message Coordinator (MC) übergeben, der die noch im internen Format befindlichen Nachrichten mit einer eindeutigen Kennzeichnung versieht, bevor er sie an den B2B-Adapter weiterreicht. Diese Kennzeichnung kann verwendet werden, um die E-Business-Nachrichten innerhalb von Verhandlungsnachrichten zu referenzieren. Beim jeweiligen Verhandlungspartner hat wiederum der MC die Aufgabe, solche Referenzen entsprechend aufzulösen.

In Vorgriff auf die detaillierte Beschreibung einzelner Klassen und Schnittstellen in Abschnitt B.1 sei hier eine kurze Übersicht über die Paketstruktur des Frameworks gegeben (siehe auch Abbildung 6.7).

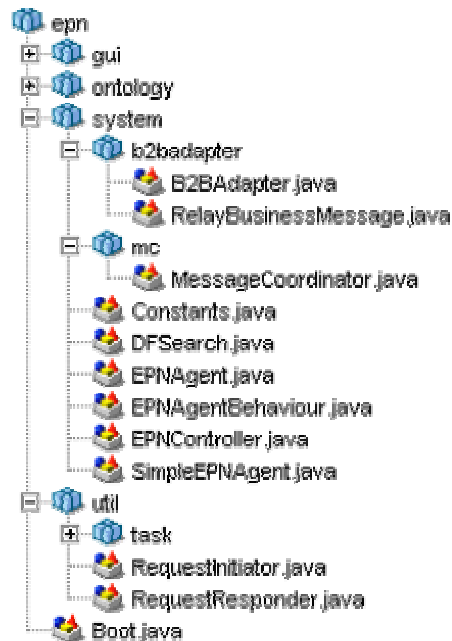


Abbildung 6.7: Paketstruktur des Frameworks (Übersicht)

epn

Auf oberster Ebene ist das EPN-System möglichst übersichtlich konzipiert; d.h. dieses Paket enthält möglichst wenig Klassen. Derzeit befindet sich hier lediglich die Klasse Boot (siehe Abschnitt B.1).

epn.gui

Dieses Paket enthält die graphischen Benutzungsschnittstellen der Systemagenten und die von diesen direkt benötigten Klassen.

epn.ontology

Dieses Paket und seine Unterpakete enthalten die Klassen, die die Ontologien des EPN-Systems definieren und die im Wesentlichen mit Hilfe von Protégé entworfen und mit JOCEP (JADE Ontology Class export Plugin, siehe Abschnitt 6.2.1.3) generiert wurden. Jeder einzelnen der in Abschnitt 6.2.1.2 beschriebenen Ontologien entspricht eines der Unterpakete.

epn.system

Dieses Paket enthält die Klassen, die für den grundlegenden Betrieb des EPNSystems benötigt werden. Dazu gehören auch die Systemagenten, deren Code sich in jeweils eigenen Paketen unterhalb von epn.system befindet.

epn.util

Hier befinden sich Klassen, die keinem der anderen Bereiche eindeutig zugeordnet werden können. Sie stellen Dienste und Methoden bereit, die von Komponenten aus unterschiedlichen Bereichen genutzt werden.

Erläuterungen zu den einzelnen Klassen des Systems finden sich im Anhang zu den Implementationen.

Zum Starten des EPN-Systems dient die Klasse `epn.Boot`. Sie steuert den kompletten Startvorgang des Systems, so dass derzeit keine der erforderlichen Komponenten zusätzlich von Hand gestartet werden müssen. Während der Initialisierung des Systems wird zunächst eine JADE Agentenplattform vorbereitet und gestartet. Anschließend wird ein EPN-Controller-Agent auf der Plattform erzeugt, dessen Aufgabe darin besteht, die benötigten Systemagenten zu starten.

Bevor das EPN-System gestartet wird, ist sicherzustellen, daß sich die benötigten Programmbibliotheken im Klassenpfad befinden. Hierzu gehören neben dem eigentlichen EPN-System die Archive der Agentenplattform JADE, das Modul für den Nachrichtentransport per HTTP sowie der von diesem benötigte XML-Parser Crimson. Der eigentliche Aufruf variiert geringfügig in Abhängigkeit vom verwendeten Betriebssystem, entspricht aber grundsätzlich dem folgenden Schema:

```
,CLASSPATH=epn.jar;jade.jar;jadeTools.jar;http.jar;crimson.jar
java epn.Boot [optionale Parameter]'
```

Einige Einstellungen des EPN-Systems können beim Start konfiguriert werden. So ist es für den Betrieb mehrerer EPN-Systeme auf dem selben Rechner beispielsweise notwendig, beiden Instanzen jeweils unterschiedliche Port-Nummern zuzuweisen.

Die Parameter, die die `epn.Boot`-Klasse derzeit unterstützt, sind in Tabelle 6.1 beschrieben.

Tabelle 6.1: Startparameter des EPN-Systems

-p <port>	TCP Port, der für den Nachrichtentransport via http verwendet werden soll.
-P <port>	TCP Port, der für RMI-Kommunikation verwendet werden soll.
-h <host>	Auflösbarer Name oder IP-Adresse, an die der Nachrichtentransport via HTTP gebunden werden soll.

-rma	Wird dieser Parameter übergeben, so öffnet das EPNSystem das Remote Agent Management GUI von JADE, das die Überwachung und Steuerung der Plattform ermöglicht.
-nogui	Dieser Parameter verhindert, daß das GUI des EPNSystems geöffnet wird. Das System läuft dann vollständig im Hintergrund.

6.2.1.2 Kommunikation innerhalb des Frameworks

Die Kommunikation zwischen den verschiedenen Systemagenten des EPNFrameworks erfolgt durch die von der JADE-Plattform bereitgestellten Dienste für den Nachrichtentransport. Die auf diese Weise ausgetauschten Nachrichten basieren auf der von der FIPA standardisierten ACL (FIPA, 2002b) und ihren kommunikativen Akten (FIPA, 2002d). Die eigentlichen Inhalte der Nachrichten werden in der Sprache SL (FIPA, 2002f) ausgedrückt und beziehen sich auf die durch die EPN-Systemontologie definierte Begriffswelt.

Im Folgenden wird auf die allgemeinen und grundlegenden Fähigkeiten der EPN-Systemagenten eingegangen und noch nicht implementierten Dienste bestehen aus Handlungen der jeweiligen Agenten, und potentielle Nutzer können diese Dienste in Anspruch nehmen, indem sie den Agenten auffordern, die entsprechende Handlung auszuführen. Eine solche Aufforderung erfolgt mittels des FIPA Request Interaction Protocols (FIPA, 2002e).

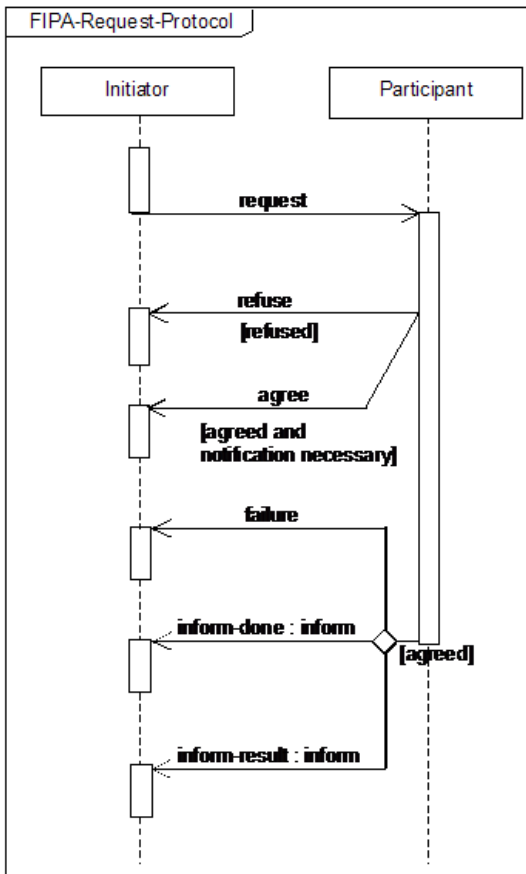


Abbildung 6.8: FIPA Request Interaction Protocol

Abbildung 6.8 zeigt das AUML-Sequenzdiagramm des Protokolls: Der Nutzer sendet einen request-Akt, dessen Proposition aus einer vom Gegenpart auszuführenden Handlung besteht, an den Anbieter. Dieser kann die Ausführung — ggf. unter Angabe eines Grundes — ablehnen, indem er mit einem ‚refuse‘ antwortet, oder durch ein ‚agree‘ zustimmen. Durch seine Zustimmung verpflichtet er sich zur Durchführung der Handlung und dazu, den Initiator im Anschluss über das Ergebnis zu informieren. Dies geschieht durch ein ‚failure‘, falls die Handlung fehlgeschlagen ist, oder durch ein ‚inform‘, das den Nutzer über eine erfolgreiche Ausführung informiert und ihm ggf. gewünschte Resultate liefert. In Anhang A finden sich hierzu konkrete Beispiele aus der internen Kommunikation des EPN-Systems.

6.2.1.2.1 Ontologien im EPN-System

Die Systemontologie definiert die Begriffswelt, auf die sich die interne Kommunikation des Frameworks bezieht. Bei diesen Begriffen handelt es sich um Konzepte und Prädikate, sowie um Agenten und Handlungen von Agenten als spezielle Untergruppen der Konzepte. Als Werkzeug für die Modellierung von Ontologien dienen hier Protégé-2000 und die Erweiterung JOCEP, ein Plugin zur Erzeugung von JADE-konformen Java-Klassen aus entworfenen Ontologien.

Diese Generierung und JOCEP werden weiter unten ausführlicher beschrieben.

Zum EPN-System gehören neben der EPN-Systemontologie weitere Ontologien, auf die später noch eingegangen wird.

Die Ontologien bilden eine hierarchische Struktur, in der grundlegende Ontologien ihre Konzepte an weiterführende vererben.

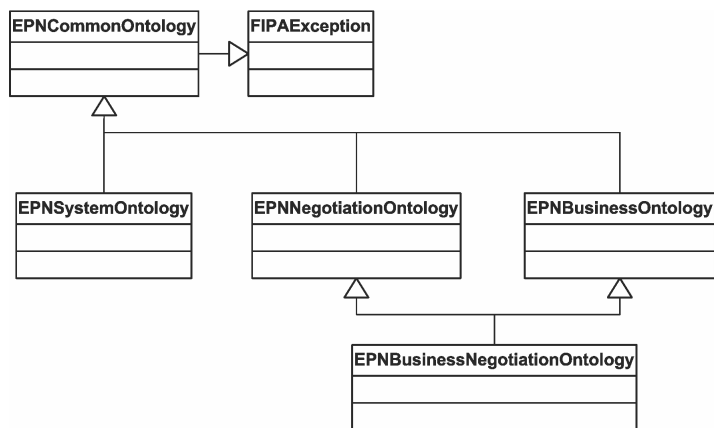


Abbildung 6.9: Hierarchie der Ontologien

Abbildung 6.9 verdeutlicht dies als UML-Klassendiagramm. Zu jeder hier als Klasse dargestellten Ontologie existiert ein entsprechendes Projekt unter Protégé. Im Einzelnen handelt es sich dabei um folgende:

- **FIPAException:** Die Prädikate, die hier definiert werden, entsprechen den durch die FIPA ((FIPA, 2002c)), Abschnitt 6.3) spezifizierten. Sie beschreiben allgemeine Fehlersituationen (exceptions), die beim Nachrichtenaustausch auftreten können.
- **EPNCommonOntology:** Dies ist die gemeinsame Oberontologie für alle Ontologien des EPN-Systems. Sie erbt unter anderem Prädikate von FIPAException.
- **EPNSystemOntology:** Diese Ontologie definiert Prädikate und Konzepte, die die Funktionen und Komponenten des EPN-Systems beschreiben. Die interne Kommunikation des

Systems basiert auf dieser Ontologie. Sie wird im folgenden Abschnitt detailliert beschrieben.

- EPNNegotiationOntology: Diese Ontologie ermöglicht es auf Grundlage des formalen Basismodells aus Kapitel 3, Aufbau und Ablauf von Verhandlungen allgemein zu beschreiben.
- EPNBusinessOntology: Diese Ontologie ermöglicht es, Inhalte verschiedener E-Business-Standards einheitlich zu erfassen.
- EPNBusinessNegotiationOntology: Diese Ontologie kombiniert Wissen über Verhandlungen und den E-Business-Bereich, und ermöglicht so Verhandlungen über Inhalte von E-Business-Dokumenten.

6.2.1.2.2 Die EPN-Systemontologie

Die EPN-Systemontologie definiert Prädikate und Konzepte, die die Funktionen und Komponenten des EPN-Systems beschreiben. Auf ihr basiert die interne Kommunikation des EPN-Systems. Sie beschreibt u.a. eine Reihe von Handlungen, die Agenten ausführen können und die die Systemagenten als Dienste bereitstellen. Agenten, die diese Dienste nutzen wollen, können die Ausführung solcher Handlungen mit Hilfe des Request-Protokolls veranlassen. In Anlehnung an die Spezifikationen der FIPA (FIPA, 2002c) kann man diese Handlungen daher als Funktionen betrachten und als solche werden sie im Folgenden beschrieben. Dabei werden einige Bezeichnungen verwendet, die näher erklärt werden müssen:

Funktion: Bezeichnet das Symbol, das die Funktion bzw. Handlung innerhalb der Ontologie identifiziert.

Ontologie: Der Name der Ontologie, zu deren Definitionen die Funktion gehört.

Unterstützt: Der oder die Agenten, die die Funktion ausführen können.

Beschreibung: Eine natürlichsprachliche Beschreibung der Semantik der Funktion.

Def.-Bereich: Der Definitionsbereich der Funktion. Die Argumente, die der Funktion übergeben werden, müssen aus dieser Menge stammen.

Wertebereich: Der Wertebereich der Funktion. Das explizite Ergebnis der Funktion ist Element dieser Menge. Funktionen, die zwar den Zustand des ausführenden Agenten verändern, aber kein explizites Ergebnis liefern, haben keinen Wertebereich.

Stelligkeit: Die Anzahl der Argumente, die die Funktion entgegennimmt.

Öffnen eines GUI

Funktion	show-gui
Ontologie	epn-system
Unterstützt	MC und B2BAdapter
Beschreibung	Ein Agent, der diese Funktion ausführt, öffnet sein GUI oder zeigt es dem Benutzer an. Derzeit ist ausschließlich der EPNController-Agent autorisiert, diese Funktion aufzurufen.
Def.-Bereich	Keiner
Wertebereich	Keiner
Stelligkeit	0

Schließen eines GUI

Funktion	hide-gui
Ontologie	epn-system
Unterstützt	MC und B2BAdapter
Beschreibung	Ein Agent, der diese Funktion ausführt, schließt bzw. versteckt sein GUI. Derzeit ist ausschließlich der EPNController-Agent autorisiert, diese Funktion aufzurufen.
Def.-Bereich	Keiner
Wertebereich	Keiner
Stelligkeit	0

Beenden eines Agenten

Funktion	shutdown
Ontologie	epn-system
Unterstützt	MC und B2BAdapter
Beschreibung	Ein Agent, der diese Funktion ausführt, meldet sich bei der Agentenplattform ab und beendet seine Aktivitäten. Derzeit ist ausschließlich der EPNController-Agent autorisiert, diese Funktion aufzurufen.
Def.-Bereich	Keiner
Wertebereich	Keiner
Stelligkeit	0

Einholen von Statusinformationen

Funktion	report-status
Ontologie	epn-system
Unterstützt	MC und B2BAdapter

Beschreibung	Ein Agent, der diese Funktion ausführt, liefert menschenlesbare Informationen über seinen Status.
Def.-Bereich	Keiner
Wertebereich	string
Stelligkeit	0

Versenden eines E-Business Dokuments

Funktion	send-business-document
Ontologie	epn-system
Unterstützt	B2BAdapter
Beschreibung	Ein Agent, der diese Funktion ausführt, versendet das übergebene EBusiness- Dokument auf geeignetem Wege an die durch den zweiten Parameter spezifizierten Empfänger. Die Definition dieser Funktion hat vorläufigen Charakter und dient eher als Beispiel für eine mögliche Umsetzung.
Def.-Bereich	business-document × Set of string
Wertebereich	Keiner
Stelligkeit	2

6.2.1.3 JOCEP - JADE Ontology Class Export Plugin

Das JADE Ontology Class Export Plugin (Gräfe, 2003), kurz JOCEP, ist eine Erweiterung für Protégé-2000, die speziell für die Entwicklung des EPN-Systems erstellt wurde. Protégé-2000 ist ein freier und im Quellcode vorliegenden Ontology- und Knowledgebase-Editor, der seit 1997 an der Stanford University School of Medicine entwickelt wird. Protégé-2000 bietet dem Benutzer unter anderem eine graphische Entwicklungsumgebung für den Entwurf von Ontologien, in der die zugehörigen Klassen, ihre Hierarchie, ihre Slots und deren Einschränkungen sowie die Beziehungen zwischen diesen Eigenschaften modelliert werden

Darüber hinaus gestattet es den Im- und Export von Ontologien in den Standardformaten RDF Schema und DAML+OIL und lässt so nötigenfalls den Austausch von Daten mit anderen Werkzeugen zu. Mit Hilfe von JOCEP können aus in Protégé entworfenen oder bearbeiteten Ontologien Java-Klassen generiert werden, die den direkten Einsatz der Ontologien in Agenten der JADE-Plattform ermöglichen. Die Anwendung solcher Ontologien in JADE wird von Caire (Caire, 2002) näher beschrieben. Allerdings sind beim Entwurf einer Ontologie in Protégé hierzu einige Einschränkungen zu beachten, denn nicht alles, was dort modellieren werden kann, lässt sich auf das Ontologiemodell von JADE abbilden.

Bereits zuvor existierte ein anderes Plugin für Protégé, der Bean Generator, mit dem sich ebenfalls JADE-kompatible Ontologieklassen erzeugen

lassen. Eine Evaluation dieses Werkzeugs zu Beginn der Arbeit ließ jedoch einige Schwächen erkennen bzw. fehlende Fähigkeiten, auf die im Verlauf der Arbeit nicht verzichtet werden konnte. Da der Bean Generator nicht im Quellcode vorlag, wurde mit JOCEP ein neuer Klassengenerator entwickelt, der den Anforderungen des EPN-Projektes genügt. Insbesondere sind es folgende Merkmale, die ihn vom Bean Generator unterscheiden:

Kombinierte Ontologien: JADE ermöglicht die Kombination von Ontologien, d.h. für eine Ontologie kann definiert werden, daß sie eine oder mehrere andere Ontologien erweitert. JOCEP unterstützt dies mit Hilfe der Protégé- Funktion zum inkludieren von Ontologien. Auf Wunsch werden die Klassen aus inkludierten Ontologien nicht direkt erzeugt, sondern stattdessen deren Definitionen an die neu zu erzeugende Ontologie vererbt.

Vocabulary Interfaces: Auf Wunsch werden die i.d.R. zahlreichen Definitionen von Konstanten, die das eigentliche Vokabular der Ontologie umfassen, in ein eigenes Java-Interface ausgegliedert. Wird darüber hinaus von der oben erwähnten Möglichkeit zur Kombination von Ontologien gebrauch gemacht, so wird auch zwischen den „Vocabulary Interfaces“ eine entsprechende Vererbungshierarchie aufgebaut, wie dies von Caire (Caire, 2002) empfohlen wird. Das Auffinden und Zuordnen bestimmter „Vokabeln“ wird auf diese Weise stark erleichtert.

Multiple Vererbung: Protégé gestattet den Einsatz multipler Vererbung beim Entwurf von Ontologien. Da dies in Java nicht direkt möglich ist, verwendet JADE hierfür sogenannte „Abstract Descriptors“. Diese besitzen die hierfür notwendige Flexibilität, auch wenn sie die Verwendung der Ontologie komplizierter gestalten (Caire, 2002). JOCEP bildet die multiple Vererbung von Konzepten mit Hilfe der „Abstract Descriptors“ ab.

Anonymous Slots: Speziell die von der FIPA standardisierte Inhaltssprache SL gestattet die Verwendung von anonymen Slots, also solchen, die in einer Instanz nicht durch ihre Bezeichner, sondern durch ihre Reihenfolge bestimmt werden. Der Bean Generator sah zwar bereits die Möglichkeit vor, solche Slots als anonym zu kennzeichnen, schien dies aber bei der Java-Klassen-Generierung nicht entsprechend umzusetzen.

Explizites Vokabular: Eine Funktion, die ebenfalls bereits im Bean Generator vorgesehen war, von diesem aber nicht vollständig korrekt umgesetzt wurde, ist die Vergabe eines zusätzlichen Bezeichners, des so genannten JADE-Names, an Klassen und Slots zur Verwendung im Vokabular der JADE-Ontologie.

Korrekte Nomenklatur der get- und set-Methoden: In allen Fällen wurden vom Bean Generator die get- und set-Methoden der zur Ontologie gehörenden Java Beans so erzeugt, dass sie vom Content Manager unter JADE korrekt verarbeitet wurden.

Abbildung 6.10 zeigt einen Screenshot der Umsetzung von JOCEP als JADE Plugin.

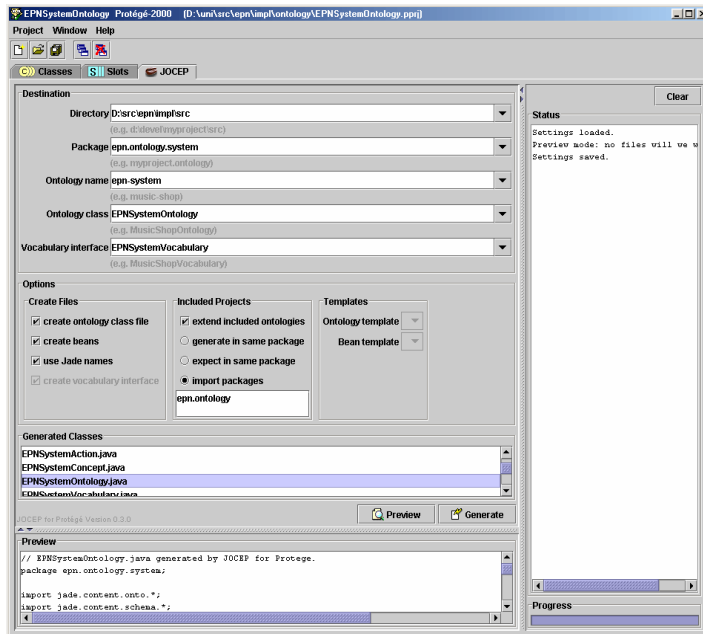


Abbildung 6.10: JOCEP

6.2.1.3.1 Entwurf einer Ontologie für JADE

Beim Entwurf einer Ontologie zur Verwendung unter JADE mit Hilfe von Protégé sind einige Besonderheiten zu beachten. Nicht alles, was sich mit Protégé entwerfen lässt, ließe sich direkt auf Java-Klassen für die JADE-Plattform abbilden.

Zunächst muss jede solche Ontologie die im Projekt BasicJADEOntology enthaltenen Klassen und Slots umfassen². Hierzu muß sie dieses Projekt entweder direkt oder indirekt inkludieren, was unter Protégé wiederum erfordert, dass sich die drei zum BasicJADEOntology-Projekt gehörenden Dateien im selben Verzeichnis wie das zu entwickelnde Ontologie-Projekt befinden. Durch die BasicJADEOntology werden im Wesentlichen die Klassen Predicate und Concept mit den Unterklassen AID und AgentAction bereitgestellt. Deren Gegenstücke unter JADE finden sich in der Java-Klasse jade.content.onto.BasicOntology und alle

Klassen der zu entwerfenden Ontologie müssen von diesen vieren abgeleitet werden. Dabei sind einige Punkte zu beachten, die sich entweder direkt aus den Spezifikationen der FIPA oder aus deren Umsetzung unter JADE ergeben:

Alle Klassen müssen direkt oder indirekt von einer der vier durch die BasicJADEOntology bereitgestellten Klassen erben.

- Von Prädikaten kann nicht geerbt werden, d.h. alle Prädikate müssen direkte Unterklassen von Predicate sein.
- Konzepte können keine Prädikate enthalten, d.h. Klassen, die von Predicate erben, sind keine zulässigen Typen für Slots.
- Multiple Vererbung wird mit Hilfe von abstrakten Descriptoren realisiert (s.o.), d.h. für Klassen mit mehreren direkten Oberklassen wird kein eigenes Java Bean erzeugt. Dies gestaltet ihre Verwendung unter JADE komplizierter. Auch wenn multiple Vererbung zulässig ist, mag es also ratsam sein, sie so weit wie möglich zu vermeiden.
- Wird ein Slot einer Klasse als anonym deklariert, so müssen alle Slots dieser Klasse anonym sein.

The image shows a graphical user interface for JOCEP. It contains three main input areas: a checkbox labeled 'ignore' on the left, a text field labeled 'JADE Name' in the center, and a text field labeled 'Java super-class' on the right. Below these fields is a larger text area labeled 'Additional Java Code'.

Abbildung 6.11: Zusätzliche Klasseneigenschaften unter JOCEP

Durch die Verwendung von JOCEP erhalten Klassen und Slots einige zusätzliche Eigenschaften, über die bestimmte Aspekte der Code-Generierung gesteuert werden können. Abbildung 6.11 zeigt die Eingabefelder, die für die zusätzlichen Klasseneigenschaften vorgesehen sind. Wird die Schaltfläche ‚ignore‘ ausgewählt, so wird die betreffende Klasse bei der Generierung vollständig ignoriert. Sie ist anschließend nicht Teil der Ontologie und für sie wird kein Java Bean erzeugt. Unterklassen einer solchen Klasse erben direkt von der Oberklasse der ignorierten. Dies kann nützlich sein, um Klassen zunächst vorläufig aus der Ontologie zu entfernen, oder auch, um beispielsweise die Prädikat-Klassen mit einer strukturierenden Hierarchie zu versehen, da diese nach der Generierung alle direkt von Predicate erben müssen (s.o.). Durch das Feld JADE Name kann ein zusätzlicher Bezeichner für die Klasse eingeführt werden, der im Vokabular der für JADE generierten Ontologie Verwendung findet. Wird dieses Feld nicht benutzt, so wird der hierfür verwendete Bezeichner in

kanonischer Weise aus dem Protégé-Klassennamen erzeugt. Letzteres ist nicht immer wünschenswert, zumal der Name des Java Beans in jedem Fall aus dem Protégé-Namen abgeleitet wird. So sollte beispielsweise nach Java-Konvention das entsprechende Konzept aus der FIPA Agent Management Ontology durch ein Java Bean mit dem Namen DFAgent-Description repräsentiert werden, während die Spezifikation die Verwendung des Bezeichners df-agent-description im eigentlichen Nachrichteninhalt verlangt.

Das Feld Java Super Class ermöglicht es, dem Java Bean, das zu der Klasse erzeugt wird, explizit eine bestimmte Superklasse zuzuweisen. Diese Zuweisung ist allerdings noch abhängig von Einstellungen bei der Generierung. Außerdem muss der Entwickler dafür Sorge tragen, dass die Anforderungen an die Java-Klassenhierarchie, die sich aus der Ontologie ergeben, berücksichtigt werden. Ein möglicher Anwendungsfall ist beispielsweise, Prädikate, die Fehlersituationen beschreiben, von der Klasse java.lang.Exception erben zu lassen. Im Feld Additional Java Code schließlich kann zusätzlicher Java-Code eingegeben werden, der direkt in das generierte Java Bean eingefügt wird.

Für Slots existiert ebenfalls ein Feld JADE Name mit entsprechender Bedeutung. Darüber hinaus kann ein Slot durch die Schaltfläche Anonymous als anonym deklariert werden, so dass er durch seine Position innerhalb der Klasse bestimmt wird.

In Abhängigkeit von den hier vorgenommenen Einstellungen kann JO-CEP bis zu drei verschiedene Arten von Java-Klassen erzeugen:

- Zu jedem Ontologienprojekt wird eine Ontologie-Klasse erzeugt, die von jade.content.onto.Ontology erbt, und die Strukturen der Ontologie den Vorgaben von JADE entsprechend abbildet.
- Zu jedem Ontologienprojekt wird ein Vocabulary Interface erzeugt, daß das Vokabular der Ontologie in Form von Konstanten definiert. Es erbt ggf. von den entsprechenden Schnittstellen inkludierter Projekte.
- Für jedes Konzept und Prädikat der Ontologie, das nicht mit Hilfe von abstrakten Deskriptoren umgesetzt wird (s.o.), wird eine Java Bean Klasse erzeugt.

6.2.2 Nachrichtenfluß im EPN-System

In diesem Abschnitt soll der Nachrichtenfluß innerhalb eines und zwischen mehreren EPN-Systemen anhand eines Beispiels verdeutlicht werden. Es orientiert sich am Fallbeispiel der Stadtwerke Tüchtigburg. Die Stadtwerke sind aufgrund des Ausfalls ihres Kleinkraftwerkes gezwun-

gen, kurzfristig zusätzliche Strompakete einzukaufen. Nachdem mögliche externe Anbieter über einen geeigneten Verzeichnisdienst ermittelt wurden, werden sie von den Stadtwerken aufgefordert, Gebote für die gewünschte Strompaketgröße abzugeben. Dies geschieht in Form des von der FIPA spezifizierten call-for-proposal Sprechaktes ‚cfp‘. Der Verhandlungsagent der Firma MegaPower wird hier aufgefordert, ein Angebot für den Verkauf von 1000MWh abzugeben, bei dem der Preis pro MWh unter dem Wert von 100 Euro liegen soll:

```
(cfp
  :sender (agent-identifier :name vagent@tuechtigburg)
  :receiver (set (agent-identifier :name
vagent@megapower))
  :content
    ((action
      (agent-identifier :name vagent@megapower)
      (sell (product-description :name power-packet :unit
MWh) 1000)
      (any ?x
        (satisfies
          (unary-constraint
            :attribute (numerical-attribute :name price
:value 100 :unit EUR)
            :operator '<' )
            ?x))))
      :language fipa-sl
      :ontology epn-business-negotiation
    )
)
```

Da jedoch aus dem Verzeichnisdiensteintrag der Firma MegaPower hervorgeht, dass diese in Anfragen enthaltene Produktbeschreibungen in einem bestimmten E-Business- Format benötigt, wird die Aufforderung nicht direkt an den Verhandlungsagenten der Firma geschickt, sondern zunächst an den Message Coordinator des EPN-Systems:

```
(request
  :sender (agent-identifier :name vagent@tuechtigburg)
  :receiver (set (agent-identifier :name epn-
mc@tuechtigburg))
  :content "((action
    (agent-identifier :name epn-mc@tuechtigburg)
    (forward-translated
      (set (agent-identifier :name vagent@megapower))
      (cfp
        :sender (agent-identifier :name
vagent@tuechtigburg)
        :receiver (set (agent-identifier :name
vagent@megapower))
        :content
          ((action
            (agent-identifier :name vagent@megapower)
            (sell (product-description :name power-packet
:unit MWh) 1000)
            (any ?x
```

```

        (satisfies
          (unary-constraint
            :attribute (numerical-attribute :name
price :value 100 :unit EUR)
            :operator '<' )
          ?x)))
      :language fipa-sl
      :ontology epn-business-negotiation
    )
  )))"
:language FIPA-SL
:ontology epn-system
:protocol fipa-request
)

```

Der Verhandlungsagent der Stadtwerke fordert den Message Coordinator durch diesen request-Akt auf, die enthaltene Verhandlungsnachricht an den Verhandlungsagenten der Firma MegaPower weiterzuleiten (forward-translated). In der Nachricht vorkommende Begriffe aus der Business-Ontologie sollen ggf. durch Platzhalter ersetzt und getrennt als Dokumente im Format klassischer E-Business-Standards auf entsprechenden Transportwegen versandt werden. Tatsächlich wird in der Nachricht eine Instanz des Konzeptes product-description gebraucht, das aus der Business-Ontologie stammt. Für diese Instanz wird ein Platzhalter mit einer eindeutigen Kennung erzeugt und an Stelle der Instanz in die Nachricht eingefügt. Die so angepaßte Nachricht wird an den Verhandlungsagenten der Firma MegaPower gesandt, mit dem zusätzlichen Hinweis, Antworten auf diese Nachricht direkt an den Verhandlungsagenten der Stadtwerke zu richten:

```

(cfp
 :sender (agent-identifier :name epn-mc@tuechtigburg)
 :receiver (set (agent-identifier :name
vagent@megapower))
 :reply-to (agent-identifier :name vagent@tuechtigburg)
 :content "(action
            (agent-identifier :name vagent@megapower)
            (sell (document-identifier ac49b718f8e937)
1000)
            (any ?x
              (satisfies
                (unary-constraint
                  :attribute (numerical-attribute :name
price :value 100 :unit EUR)
                  :operator '<' )
                ?x))
              ))"
            :language FIPA-SL
            :ontology epn-business-negotiation
          )
)

```

Die Übersetzung in klassische Standards und der Versand solcher Dokumente ist Aufgabe des B2B-Adapters. Die Instanz des Konzeptes product-

description wird daher zusammen mit dem für sie generierten Platzhalter an den B2B-Adapter-Agenten weitergereicht. An diesen ergeht die Aufforderung, die Instanz in ein geeignetes EBusiness-Format zu übersetzen und anschließend auf einem entsprechenden Transportwege zu versenden:

```
(request
  :sender (agent-identifier :name epn-mc@tuechtigburg)
  :receiver (set (agent-identifier :name b2b-
adapter@tuechtigburg))
  :content "(action
    (agent-identifier :name b2b-
adapter@tuechtigburg)
    (send-business-document
      (set (agent-identifier :name
vagent@megapower)))
    (document-identifier ac49b718f8e937)
    (product-description :name power-packet
:unit MWh)
  )))"
  :language FIPA-SL
  :ontology epn-system
  :protocol fipa-request
)
```

Der eindeutige Bezeichner wird bei diesem Vorgang vom B2B-Adapter an passender Stelle in das Business-Dokument eingefügt, so daß er vom B2B-Adapter der Gegenseite ausgelesen werden kann, und deren MC die Zuordnung des Dokumentes zur entsprechenden Verhandlungsnachricht ermöglicht. Der B2B-Adapter der Gegenseite übersetzt empfangene Dokumente nach Möglichkeit zurück in Konzepte der Business-Ontologie und übergibt sie dem MC zusammen mit dem extrahierten Bezeichner:

```
(inform
  :sender (agent-identifier :name b2b-adapter@megapower)
  :receiver (set (agent-identifier :name epn-
mc@megapower))
  :content "(document-received
    (document-identifier ac49b718f8e937)
    (product-description :name power-packet
:unit MWh)
  )"
  :language FIPA-SL
  :ontology epn-system
)
```

Der Verhandlungsagent der Firma MegaPower erhält die an ihn weitergeleitete Nachricht des Agenten der Stadtwerke. An Stelle der Beschreibung des Produktes, für das er ein Angebot abgeben soll, findet er zunächst nur einen Platzhalter. Mit diesem Platzhalter kann er sich an seinen lokalen MC wenden, um den Inhalt der Produktbeschreibung zu erhalten.

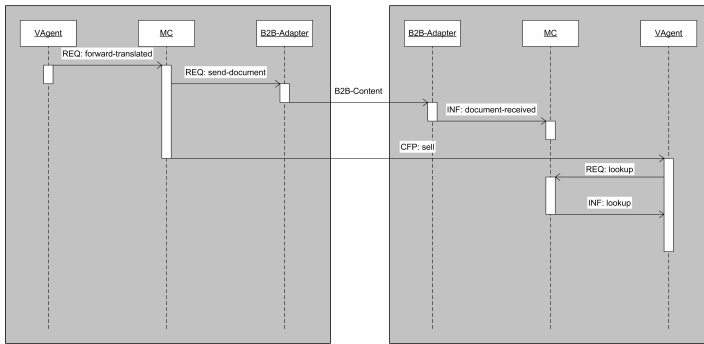


Abbildung 6.12: Sequenzdiagramm: Business-Dokument trifft vor der Verhandlungsnachricht ein

Da die eigentliche Verhandlungsnachricht aber unabhängig vom Business-Dokument versandt wird, kann nicht garantiert werden, daß letzteres auf der Gegenseite bereits verfügbar ist, wenn die Verhandlungsnachricht eintrifft und verarbeitet wird. Das Sequenzdiagramm in Abbildung 6.12 verdeutlicht den Nachrichtenfluß für den Fall, daß der Inhalt des Business-Dokumentes zu dem Zeitpunkt, an dem er benötigt wird, bereits beim MC vorliegt. Der Verhandlungsagent erhält hier vom MC direkt die Auflösung der Referenz.

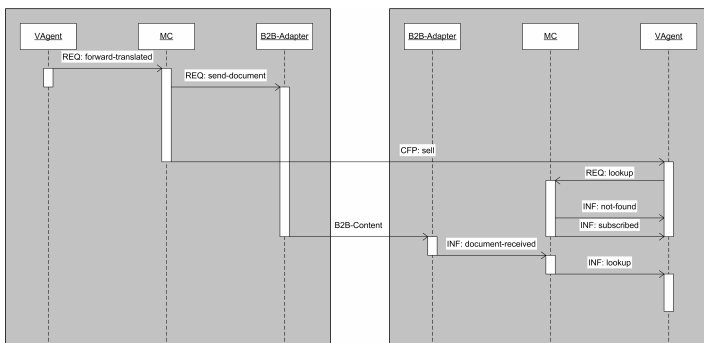


Abbildung 6.13: Sequenzdiagramm: Verhandlungsnachricht trifft vor dem Business-Dokument ein

Für den Fall, daß die Verhandlungsnachricht eintrifft (Abbildung 6.13) und bearbeitet wird, bevor das zugehörige Business-Dokument vorliegt, wird der Verhandlungsagent vom MC zunächst über diesen Umstand informiert. Darüber hinaus erhält er vom MC die Zusage, informiert zu werden, sobald die fehlenden Daten eintreffen.

6.2.3 Architektur der B2B-Brokerkomponente

Die Brokerkomponente hat die Aufgabe zwischen den verschiedenen Standards des B2B-Messaging zu vermitteln und so eine reibungslose Kommunikation in einem heterogenen Umfeld zu garantieren.

Dazu wird hier eine exemplarische technische Umsetzung der in Kapitel 5 vorgestellten Konzepte erläutert.

Abbildung 6.14 zeigt an einem Beispiel den generellen Datenfluss in der Brokerkomponente.

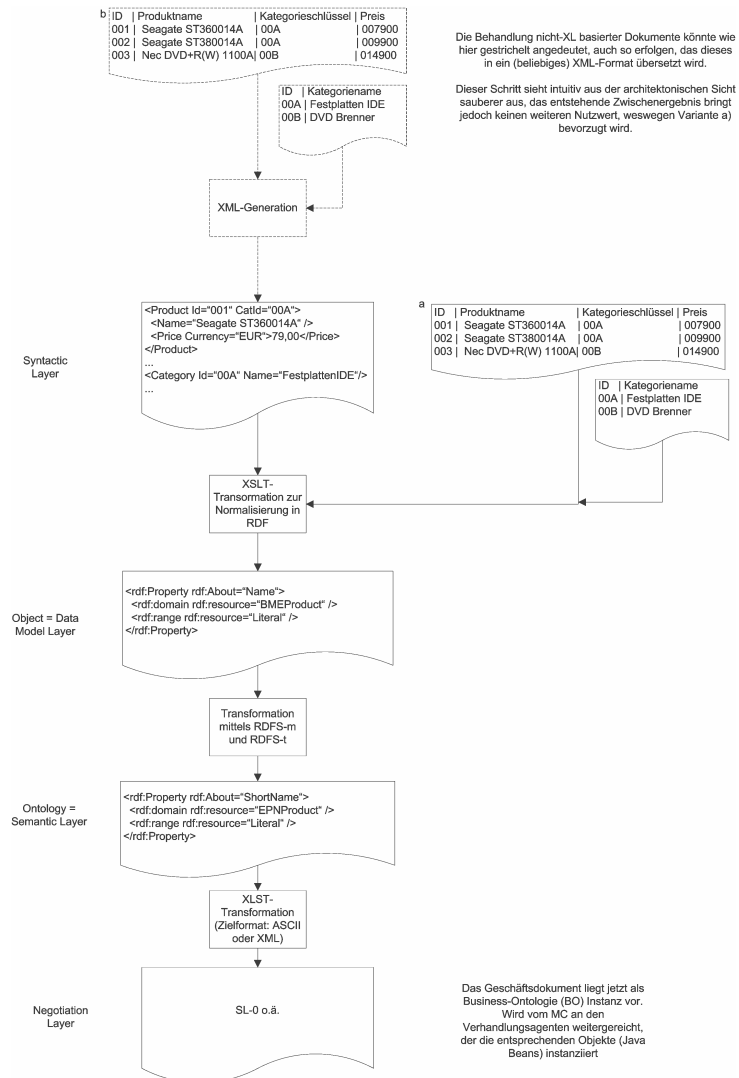


Abbildung 6.14: Datenfluss in der Brokerkomponente

Der erste Layer - *Syntactic Layer* - beschreibt die Serialisierung der Objekte in eine Folge von Zeichenketten; in diesem Beispiel einmal XML und einmal Text-basiert.

In einem XSLT Transformationsschritt wird das XML des konkreten B2B XML-Standards in ein normalisiertes Format gebracht. Bei dem normalisierten Format handelt es sich um RDF.

Der *Object oder Data Model Layer* ermöglicht einen objekt-orientierten Zugriff auf die Informationen. Die Daten liegen jetzt in einem normalisierten Tripel-Format (RDF) vor, um syntaktisch unabhängig von dem Quellformat zu sein.

In einem weiteren Schritt wird mittels RDFS-m und RDFS-t die Daten aus dem Data Model Layer in das Format des EPN Ontology Layers abgebildet. RDFS-t ist eine Mapping-Sprache, die auf RDF basiert und eine Transformationssprache, welche auf RDF-Ebene ähnliches leistet, wie XSLT auf XML-Ebene, allerdings auf einer RDF entsprechenden höheren Abstraktionsebene.

Im *Ontology oder Semantic Layer* wird eine generische Ontologie für jedes mögliche Geschäftsdokument (Katalog, Bestellung) definiert, welche als potentielle Obermenge mehr Informationen enthält, als ein Geschäftsdokument eines bestimmten Formates.

Schließlich wird aus dieser Ontologie-Schicht eine syntaktische Abbildung auf die Inhaltssprache der Verhandlungs-Ontologie vorgenommen.

Die Transformation der Dokumentinstanzen erfolgt im Betrieb automatisch. Die Erstellung der dafür nötigen Transformationsregeln kann jedoch (im Einklang mit der Literatur) nicht automatisch geschehen. Sie werden manuell erstellt. Hier ist Unterstützung durch Tools auf zwei Arten möglich:

- Ein grafischer Editor kann die Abbildung eines Quellformates auf RDF erleichtern, da diese Arbeitsweise übersichtlicher ist. Ferner können die benötigten XSL Templates automatisiert erzeugt werden – beide Punkte beschleunigen die Arbeitsweise und verringern die Fehlerwahrscheinlichkeit.
- Sogenannte ‚semi-automatic schema matcher‘ können anhand verschiedener Kriterien, wie beispielsweise struktureller oder linguistischer Analysen, Kandidaten für die Transformation vorschlagen, die dann vom Benutzer bestätigt oder verworfen werden. Dieser Ansatz wurde nicht weiter verwendet, weil die entsprechenden in der Literatur vorgestellten Prototypen schwer bis gar nicht erhältlich sind und sich dieses Vorgehen erst bei einer größeren Menge von Dialekten und Dokumenten auszahlt,

was für die vorliegende Prototypentwicklung nicht gegeben ist. Der Ansatz wird jedoch noch ausführlicher erläutert.

Für die Implementierung der Brokerkomponente im Rahmen des EPN-Prototypen wurden die nötigen Abbildungsschritte manuell durchgeführt, mit einem grafischen Editor als Hilfe.

Die Verwendung halb-automatischer Matcher wurde verworfen, da entsprechende Software / Prototypen nicht zugänglich sind und auch einen nachteiligen Eingriff zur Laufzeit erfordern würden.

Im Folgenden wird der Transformationsprozess noch einmal detaillierter betrachtet.

Schritt 1a: Überführung von Legacy-Formaten in normalisiertes RDF-Tripelformat

Wie in der Architekturskizze oben angedeutet sollen Legacy-Daten, die in der Regel ein Text-Format benutzen (bspw. EDI), in das normalisierte RDF-Tripelformat überführt werden.

Dabei ist zu beachten, dass mittels XSLT zwar XML in nicht XML-konforme Textformate transformiert werden kann, aber nicht umgekehrt. Das bedeutet, dass an dieser Stelle eine andere Technik zum Einsatz kommen muss:

- entweder kann ein entsprechender Parser entwickelt werden, welcher allgemeine, aber strukturierte Textformate direkt in das normalisierte RDF-Tripelformat umsetzen kann (Variante a im Architekturdiagramm)
- oder das Legacyformat wird recht einfach strukturgleich in XML umgesetzt und durchläuft danach den „normalen“ Prozess der B2B XML-Standards (Variante b im Architekturdiagramm)

Schritt 1b: Überführung des B2B XML-Standard in normalisiertes RDF-Tripelformat

Hierfür sind eine Reihe von Programmen aus dem OpenSource Bereich verfügbar.

Schritt 2: Überführung des normalisierten RDF-Tripelformat in die (Business)-Ontologie

Der Ontology Layer beinhaltet eine generische Ontologie für jeden Dokumententyp wie Produktkatalog, Preisanfrage oder Bestellung.

Das Geschäftsdokument, welches nun im normalisierten RDF-Tripelformat vorliegt, wird auf diese generische Ontologie abgebildet. Diese generische Ontologie sollte idealerweise eine Obermenge der in den Standards implizit vorhandenen Konzepte und Attribute sein.

Allerdings kann durch die offene Umgebung jederzeit die Anforderung entstehen neue B2B-Standards in das System zu integrieren. Daher muss man davon ausgehen, da es immer Konzepte in einem B2B-Standard geben kann, die nicht in der generischen Ontologie vorhanden sind.

Als Workaround dafür muss die generische Ontologie daher das Konzept einer frei benennbaren, optionalen Entität beinhalten, um beliebige frei definierbare Konzepte und Attribute zu transportieren, ähnlich den „tagged values“ in UML.

Der Mapping-Prozess vom normalisierten RDF-Tripelformat auf die generische Ontologie wird aufgrund nicht zu gewährleistender Zuverlässigkeit manuell auf Schema-Level durchgeführt. Die tatsächlichen Instanzen werden dann automatisiert anhand der auf dem Schema definierten Mappings durchgeführt.

Für die Erstellung des Mappings können Hilfsmittel wie graphische Editoren verwendet werden, um den Prozess übersichtlicher und möglichst wenig fehleranfällig zu gestalten.

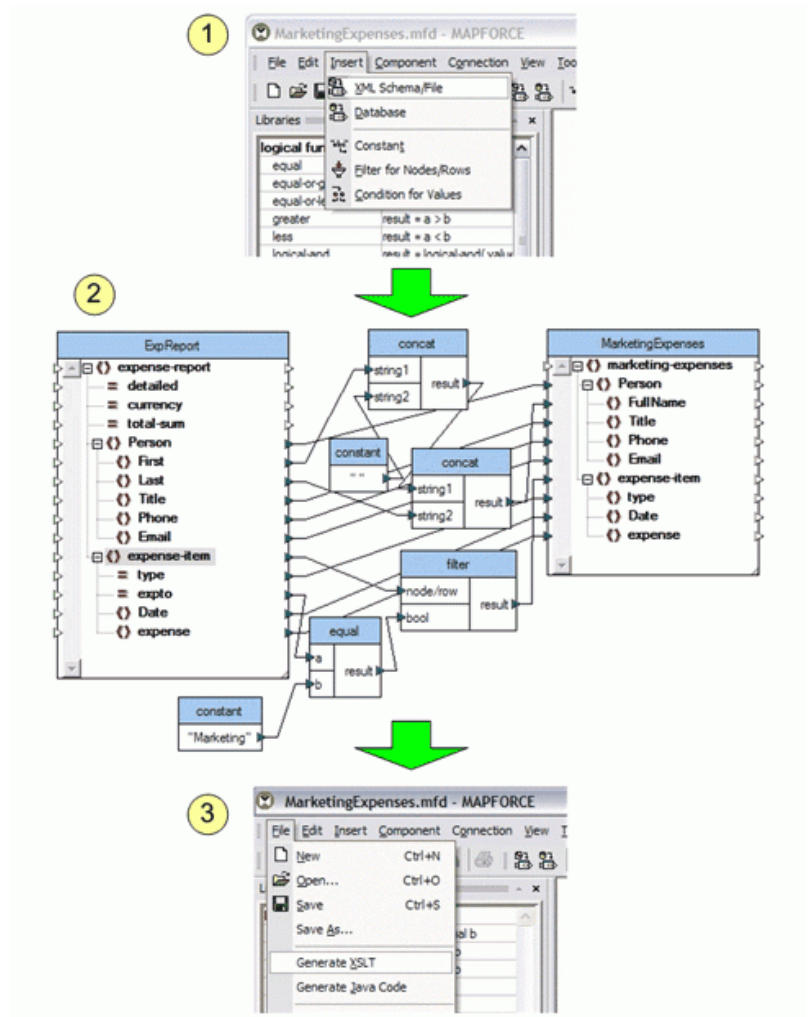
Omelayenko und Fensel (Omelayenko, et al., 2001) proklamieren eine Mappingsprache RDFS-m mit einer dazugehörenden Transformationsprache RDFS-t. Die Transformation kann automatisiert aus den Mappings abgeleitet werden. Die Mappingsprache ermöglicht somit eine deskriptive Beschreibung der durchzuführenden Abbildungen.

Diese beiden Sprachen sind jedoch de facto nicht vorhanden. Zur Zeit gibt es zu dem keine Editoren, welche bei der Erstellung von Mappings direkt auf der RDF-Ebene arbeiten – sie verwenden allesamt XML.

Zwei Beispiele für solche Editoren sind der Biztalk Mapper (Microsoft), welcher im Biztalk Server für die grafische Erstellung von Übersetzungen zwischen zwei XML-Formaten zum Einsatz kommt, und MapForce von Altova (Hersteller des bekannten XML-Spy).

Beide erlauben die grafische Bearbeitung der Mappings und die Definition von sogenannten Functions, die bestimmte Berechnungen (wie Umrechnung zwischen Einheiten, Verarbeitung von Strings) innerhalb der Mappings erlauben.

Anbei ein Beispiel aus MapForce:



Zu 1):

Hier können vordefinierte und eigendefinierte Functions abgerufen werden.

Zu 2):

Links ist das Quelldokument, rechts das Zieldokument, beide auf Schemaebene, d.h. es muss eine entsprechende DTD oder XML-Schema vorliegen.

Zu 3):

Beim Speichern können die benötigten XSL-Templates oder Java-Code, der diese Templates über eine definierte Schnittstelle kapselt, automatisch erzeugt werden

Es existiert ein Plug-In für den Ontologie-Editor Protégé, welches das Mergen zweier Ontologien unterstützt, in dem es eine grafische Unterstützung anbietet und anhand relativ simpler Algorithmen, wie beispielsweise durch die Beachtung von Ähnlichkeiten in Klassen-Namen, Kandidaten für das Mapping vorschlägt (siehe Abbildung 6.15).

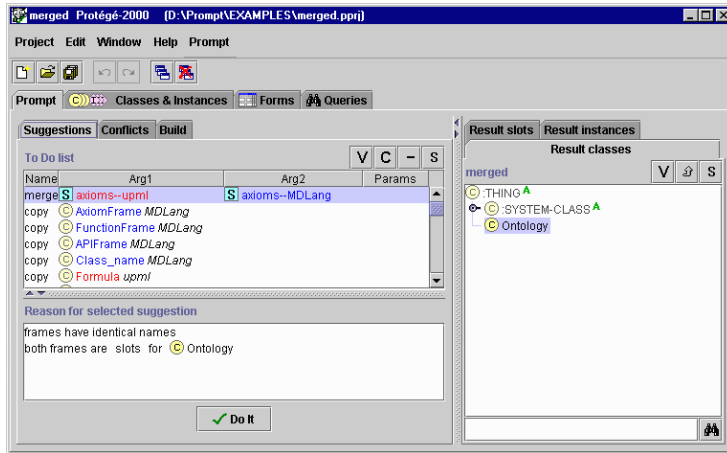


Abbildung 6.15: Merging-PlugIn für Protege

Interessant ist dieser Ansatz für eine explorative Erarbeitung der Problematik.

Schritt 3: Serialisierung der Instanz-Ontologie in ein Repräsentationsformat der Inhaltssprache

Die Ontologie liegt zu diesem Zeitpunkt im RDF-Format vor. Sie muss vor der Weiterleitung über den ‚Message Coordinator‘ an den Verhandlungsagenten entsprechend serialisiert werden.

Zwei Ansätze sind denkbar: entweder wird der JADE-Plattform die Fähigkeit gegeben, die Inhaltssprache in einer RDFS/RDF Repräsentation zu verarbeiten oder die Ontologie wird aus dem RDF-Format in eine der drei „out-of-the-box“ verstandenen Repräsentationen von JADE umgewandelt (ASCII, bit-efficient, XML).

Der erste Ansatz wird dadurch unterstützt, dass es bereits RDF-Codecs auf der Projektseite von JADE gibt, welche die Fähigkeiten der Agentenplattform entsprechend erweitern.

Somit kann die Brokerkomponente zwischen den verschiedenen Standards des B2B-Messaging vermitteln.

6.2.4 Implementation des Bootstrapping-Konzeptes

In Kapitel 4 wurde das Bootstrapping-Konzept zur Verhandlung von Verhandlungsprotokollen vorgestellt. Eine Beispielarchitektur zur Realisierung dieses Konzeptes wurde auf der Basis eines FIPA-konformen sprechaktbasierten Agentenkommunikationssystems (JAS – Java Agent Services) vorgenommen und fügt sich somit in die Entwurfskonzepte der Gesamtarchitektur ein.

Entwickelt wurde die Lösung unter Java 2 SDK, Standard Edition Version 1.4.1. Sie besteht im Kern aus den drei Klassen ASW, ASWException und Voting, wobei die erste die MessageListener-Methoden festlegt und ASWException für das Nutzen der Exceptions innerhalb von ASW sorgt.

Die Version der zugrunde liegenden JAS-Klassen ist vom 30. Januar 2002. Im Rahmen der prototypischen Entwicklung der Systemkomponenten für das Bootstrapping-Konzept wurden Vereinfachungen bei der Nutzung der JAS-Plattform vorgenommen. Die SL-Syntax wurde für diese Systemkomponente durch eine eigene einfache Syntax ersetzt (z.B. "C/B/request/voting/(auctiontype/borda)"). Statt "propose" wird auch für die Votingübermittlung ein "inform" verwendet. Und für die Hauptklasse "Voting" wurde ein Layer gebaut, das einfache Textnachrichten zwischen den initiierten Agenten austauscht, wobei die Funktionalitäten von JAS intern weiter genutzt werden.

Die grundsätzlichen Kommunikationsabläufe werden durch einen ‚MessageListener‘ asynchron realisiert, dessen Interface von der jeweiligen Klasse realisiert wird. Der ‚MessageListener‘ ruft eine Callback-Funktion auf (‚receiveMessage‘), wenn eine Nachricht eintrifft.

Das Demonstrationsprogramm sieht drei Agenten vor, deren Namen und Rollen als Szenario festgelegt sind ("A", "B", "C"=Initiator). Jede Instanziierung von Voting.class erzeugt eine Abfrage des Buchstabens, wurde einer der Dreien eingegeben, wird der Agent registriert und gestartet. Agent C als Initiatoragent schickt an A und B eine Aufforderung zur Abgabe des Borda-Votings via Message und dem Performativ ‚request‘. Die Parameter des Message-Inhalts bestehen aus Absender, Empfänger, Aktion (‚Senden‘ oder ‚Empfangen‘) und dem eigentlichen Inhalt, im Falle von C ist das ‚C‘, ‚B‘, ‚send‘, ‚startbootstrapping‘. Durch die Instanziierung der empfangenden Agentenklasse werden wiederum ‚sender‘, ‚receiver‘, ‚performativ‘ und ‚content‘ eingelesen und ausgewertet. Im Falle des Performativs ‚request‘ wird die manuelle Eingabe einer Bordawahl für vier Auktionstypen erfordert, im Format beispielsweise 2-3-1-4. Der Gewinner der Wahl wird den Agenten über das Attribut "Voting.Endergebnis" als Content des "inform"-Performativs mitgeteilt.

Eine typische Bildschirmausgabe für einen Agenten, der nicht die Initiator-Rolle hat, ist in Abbildung 6.16 zu sehen; in einigen Zeilen werden die Inhalte der Nachrichten protokolliert.

Abbildung 6.17 zeigt anschließend wie sich die erste in Abbildung 6.16 übertragende Nachricht im Detail zusammensetzt. Dazu wurde mit der Methode `toString()` der innere Teil der dem `MessageListener` übergebenen `Transport-Message` herausgefiltert. Ein vollständiges Parsen des Strings wäre möglich, jedoch bietet JAS prinzipiell selbst Methoden die hier gekapselten Eigenschaften der `Message` auszulesen. Sehr gut lässt sich in diesem String der FIPA-Standard wieder erkennen.

```

Bitte Agenten-Name eingeben:
B
---
B receive
---
Bisher kein Initiatoragent
Lebensdauer 180 Sekunden

'receiveMessage' aktiviert:
Content = C/B/request/voting/(auctiontype/borda)
-----
s: C r: B p: request c: voting/(auctiontype/borda)
-----
*****
Bitte Borda-Voting für Auktionstypen durchführen
1: dutch auction
2: englisch auction
3: FPSB
4: Vickrey

*****
Nach abfallendem Rang im Format a-b-c-d

1-2-3-4

---
B C send B/C/inform./(1-2-3-4)
---
Ziel-Agent: C
'receiveMessage' aktiviert:
Content = A/B/(Borda-Result: 1)
-----
s: A r: B p: (Borda- Result: 1)

```

Abbildung 6.16: Bildschirmausgabe für einen Agenten-Task beim Borda-Voting

```

waiting...

received message:

[transportmessage:
  [envelope:
    [receiver:
      [locator:
        [properties:
          [property:
            [fqdn: javax.agent.locator.ip.rmi.id]
            [string: f81a8fcf838445b3:1f9dc36:f19d420f92:-7fed]
          ]
          [property:
            [fqdn: javax.agent.locator.host]
            [string: 217.227.41.33]
          ]
          [property:
            [fqdn: javax.agent.locator.port]
            [number: 55555]
          ]
          [property:
            [fqdn: javax.agent.locator.protocol]
            [string: rmi]
          ]
        ]
      ]
      [type: rmi]
      [address: rmi://217.227.41.33:55555/f81a8fcf838445b3:1f9dc36:f19d420f92:-7fed]
    ]
    [sender:
      [locator:
        [properties:
          [property:
            [fqdn: javax.agent.locator.host]
            [string: 217.227.41.33]
          ]
          [property:
            [fqdn: javax.agent.locator.ip.rmi.id]
            [string: f81a8fcf838445b3:1f9dc36:f19d427c5c:-7ff6]
          ]
          [property:
            [fqdn: javax.agent.locator.protocol]
            [string: rmi]
          ]
          [property:
            [fqdn: javax.agent.locator.port]
            [number: 55555]
          ]
        ]
      ]
      [type: rmi]
      [address: rmi://217.227.41.33:55555/f81a8fcf838445b3:1f9dc36:f19d427c5c:-7ff6]
    ]
  ]
  [payload:
    [message:

```

Abbildung 6.17: Inhalt einer JAS-Transport-Message

Weitere Details der technischen Implementation sind in Knollmann (Knollmann, 2003) zu finden.

6.2.5 Realisierung von Verhandlungsunterstützungsfunktionen für den E-Business

Um zu innovativer Unterstützung von Verhandlungen im Electronic Business zu kommen, sollen

- die im Abschnitt 3.4 „Elektronische unterstützte Verhandlungen“ vorgestellten Unterstützungsfunktionen und -arten in sich sinnvoll ergänzender Weise in einem Systemmodul kombiniert werden
- die spezifischen Gegebenheiten der Anwendungsdomäne Electronic Business direkt im Systemmodul beachtet werden

Im Kontext dieser Arbeit muss dafür ein prototypisches Modul entwickelt werden, wie beispielsweise in (Adwiraah, 2003), das dem ersten Punkt entsprechend vielfältige Unterstützungsfunktionen sinnvoll integriert. Dieses Modul mit dem Namen SPNSS (Solution driven and Process support Negotiation Support System) wird im Folgenden vorgestellt.

Entsprechend dem zweiten Punkt geht das Modul auch auf die Anwendungsdomäne Electronic Business ein. Eine noch tiefer gehende Integration stellt die Einbeziehung des Messaging im Electronic Business dar.

6.2.5.1 Funktionalitäten des SPNSS

Für SPNSS werden für den lösungsorientierten Teil insbesondere Techniken der ‘Negotiation Analysis’ und für den prozessorientierten Teil sprechaktbasierte Vorgehensweisen verwandt.

Der lösungsorientierte Teil orientiert sich größtenteils an dem solution driven System INSPIRE. So werden unterstützt:

- das Festlegen der Präferenzen des Benutzers und das Erstellen der utility-function:
 - Der Benutzer bewertet die Attribute des Verhandlungsgegenstandes nach Nützlichkeit (utility) auf einer Skala von 1 bis 100.
 - Ebenso werden die Optionen der Attribute durch den Benutzer nach Nützlichkeit bewertet. Hierbei richtet sich die Breite der Skala nach der Bewertung für das jeweilige Attribut.
 - Aus diesen Angaben wird mit Hilfe der Conjoint Analyse die utility-function des Benutzers erstellt.
- das Erstellen und Verschicken von Angebotspaketen und Nachrichten:
 - Die eigenen Angebote und die Gegenangebote des Verhandlungspartners werden mit Hilfe der zuvor ermittelten utility-

function für den Benutzer automatisch nach Nützlichkeit bewertet.

- o Die Kommunikation zwischen den Verhandlungspartnern wird durch den Austausch von unstrukturierten Textnachrichten unterstützt. Die Übermittlung der Nachrichten geschieht nahezu in Echtzeit und bietet daher einen relativ kurzen communication cycle und folglich einen reichen Kommunikationskanal.
- die Optimierung des Verhandlungsergebnisses durch das Generieren von Pareto-optimalen Lösungsvorschlägen während der Verhandlung.

Im prozessorientierten Teil kommt insbesondere die Sprechaktttheorie zum Tragen. Sie bietet die Möglichkeit den Kommunikationsprozess einer Verhandlung besser zu strukturieren.

Dazu werden nur Äußerungen bestimmten Typs und nur in einer bestimmten Abfolge zugelassen. Beispielsweise könnte festgelegt werden, dass eine Verhandlung nur durch Direktiva der Form ‚request‘ oder ‚offer‘ eröffnet werden kann, und dass auf ein ‚request‘ nur eine ‚counteroffer‘, ein ‚accept‘ oder ein ‚reject‘ der Gegenseite folgen darf. Eine solche Festlegung ergibt eine Ablaufsequenz des Verhandlungsprozesses nach bestimmten Regeln: ein Verhandlungsprotokoll (Gramer, 2001; Schoop, 2000; Searle, 1971).

Der prozessorientierte Teil bietet:

- grundlegend den Austausch von unstrukturierten Textnachrichten
- eine Strukturierung des Nachrichtenaustausches durch die Typisierung der Nachrichten nach Sprechakten.
- Verwendung von ‚Turn Taking‘-Regeln der Konversationsanalyse für zwei Verhandlungspartner zur Strukturierung des Verhandlungsprozesses.
- Berücksichtigung von ‚adjacency pairs‘ zur teilweisen Strukturierung des Ablaufs

Abbildung 6.18 zeigt beispielhaft wie durch SPNSS Verhandlungsformen konfiguriert werden können, wie Sprechaktnachrichten versandt und empfangen werden und wie Angebotspakete bewertet werden.

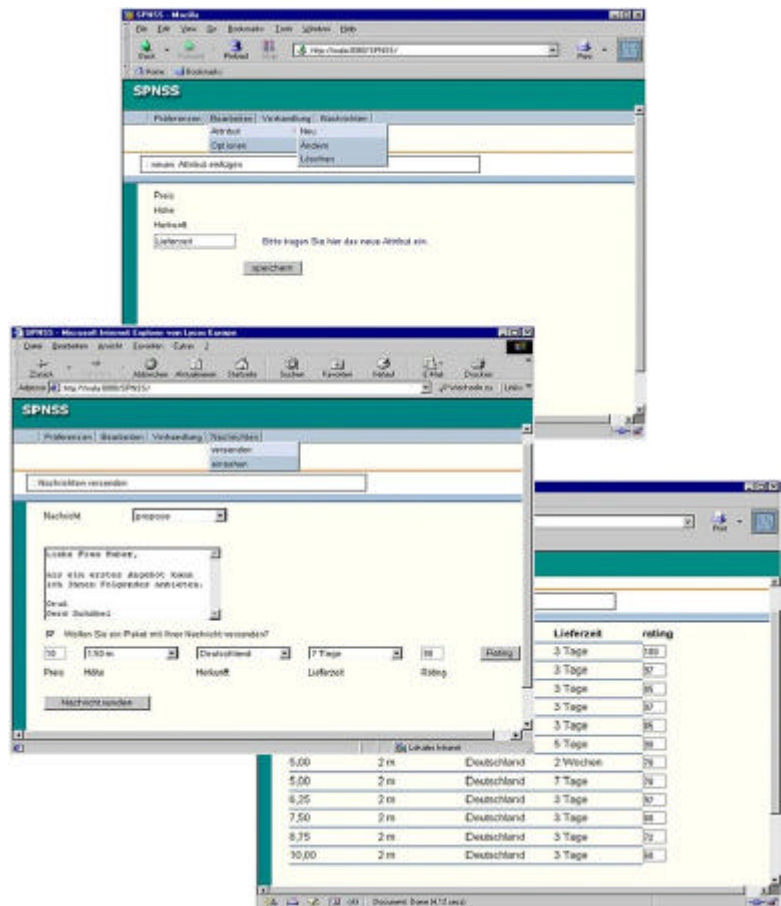


Abbildung 6.18: SPNSS – Beispiele

6.2.5.2 Technische Architektur des SPNSS

Abbildung 6.19 zeigt die grundlegende webbasierte Architektur des SPNSS Moduls.

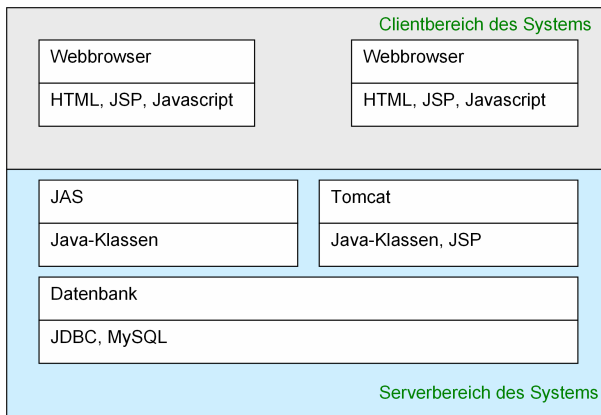


Abbildung 6.19: SPNSS - Architektur

Da SPNSS ein webbasiertes System ist handelt es sich bei der Architektur um eine Client-Server-Architektur, bestehend aus Webserver und Webbrowser. Die Applikation verwendet serverseitig Java Server Pages (JSP) und clientseitig HTML und JavaScript. Als JSP-Server wurde Tomcat 4.1. verwendet. Für die Nachrichtentransport zwischen den Clients wird während der Verhandlung die sprechaktbasierte FIPA ACL, hier im Prototypen mit Hilfe der JAS (Java Agent Services) Implementation, verwendet.

6.2.5.3 Einordnung des SPNSS

Mit SPNSS kann somit eine Kombination aus lösungsorientierten und prozessorientierten Ansätzen zur Verhandlungsunterstützung angeboten werden. Damit werden die Vorteile beider Herangehensweisen vereint.

Durch die für lösungsorientierte Systeme typische Ermittlung der Präferenzen der Benutzer am Anfang der Verhandlung werden sich die Verhandelnden ihrer Verhandlungsziele besser bewusst und können diese während der Verhandlung besser verfolgen. Gleichzeitig wird dadurch die Erstellung einer Nutzenfunktion ermöglicht, die wiederum den Ausgangspunkt für die Automatisierungsansätze im Rahmen der 'Negotiation Analysis' bietet.

SPNSS bietet jedoch auch die für prozessorientierte Systeme typischen Freiheiten für intuitive Handlungen der Benutzer. Durch die verbesserte Kommunikation wird eine Komplexitätsreduktion für den Benutzer erreicht, was auch zu mehr Übersicht während einer prozessorientierten Verhandlung führt und das Finden einer gemeinsamen Lösung erleichtert. Die Parteien können besser auf einander eingehen, leichter und schneller neue Informationen austauschen und in den Verhandlungsprozess mit einbeziehen. Dies kann zu einer integrativen Verhandlungs-

weise beitragen und wöhlmöglich eine weniger erfolgreiche und weniger nützliche distributive Verhandlung verhindern.

6.2.6 Elemente systemtechnischer Kommunikation und Interoperabilität

Die Nutzung der Web-Service-Technologie kann die Interoperabilität zwischen verhandelnden E-Business-Anwendungen deutlich erhöhen. Dies wird in der Folge anhand der Realisierung entsprechender prototypischer Systemkomponenten gezeigt.

6.2.6.1 Web-Services für die Verhandlungskommunikation

Um für die offenen Systeme mit Verhandlungsfunktionalitäten eine maximale Anzahl an Verhandlungspartnern zu ermöglichen, sollte ein hohes Maß an systemtechnischer Interoperabilität erreicht werden. Dazu trägt eine Ausstattung dieser Systeme mit interoperablen Web-Services (siehe Abschnitt 6.1.2) als Kommunikationsschnittstellen bei. Dieses Konzept wurde im Kontext dieser Arbeit als Prototyp umgesetzt (Waldau, 2002) und wird im folgenden beschrieben.

Die grundlegenden Anforderungen an eine Web-Service basierte Kommunikationsschnittstelle bestehen im Austausch von Nachrichten mit anderen Web-Service konformen Diensten (siehe Schritte 2 + 5 in Abbildung 6.20) und in der Integration / Kommunikation mit dem eigenen Anwendungs-Backend, hier aus Sicht des Web-Service als ‚Client‘ bezeichnet (siehe Schritte 1 + 6 in Abbildung 6.20 für WebService01).

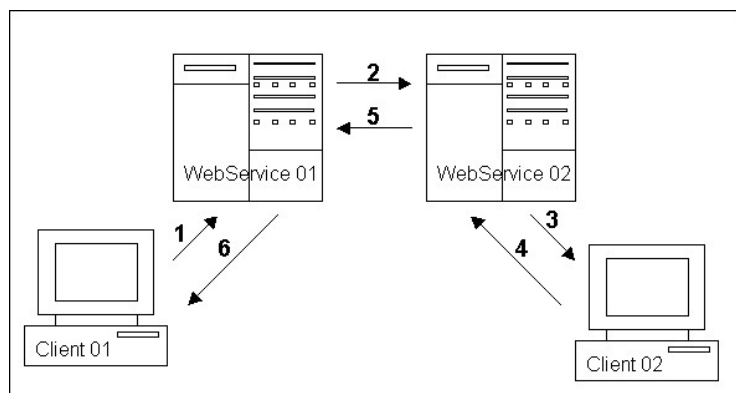


Abbildung 6.20: Ablauf der Kommunikation

In Abbildung 6.20 ist der Ablauf der Kommunikation zwischen zwei Web-Services, wie sie prototypisch zur interoperablen Verhandlungskommunikation implementiert wurden, schematisch dargestellt.

Die Nachrichten werden zur Übertragung in SOAP-Pakete verpackt und sind damit für alle SOAP unterstützenden Dienste lesbar. Zur Implementation fand das Apache SOAP Projekt (<http://xml.apache.org/soap/>) Verwendung. Es bietet neben einem SOAP-Server Unterstützung für den Aufbau und das Versenden von SOAP-Nachrichten mit Hilfe von Java-Klassen. Durch diese Unterstützung wird der Aufwand auf der Serverseite minimiert.

Grundsätzlich sieht der Ablauf des Nachrichtenaustausches (vgl. Abbildung 6.20) wie folgt aus:

- 1.) Der Client verschickt seine Nachricht über die Methode *sendAct* an den gewählten Webservice.
- 2.) Der Webservice verarbeitet den Aufruf und gibt die Nachricht an den korrespondierenden Webservice (Methode *storeAct*) weiter.
- 3.) Nach Anfrage des zweiten Clients (Methode *recAct*) überträgt der Webservice als Antwort auf den Aufruf die für ihn bestimmte Nachricht oder bei Misserfolg (keine Nachricht vorhanden) eine Fehlermeldung.
- 4.-6.) Die Antwort des zweiten Clients verläuft analog zu den Schritten 1 –3

Um die entwickelten Komponenten reibungslos in andere Systeme integrieren zu können, ist insbesondere eine lose Kopplung erforderlich.

Diese Unabhängigkeit von Client und zugehörigem Webservice wurde erreicht indem Client und Webservice mittels SOAP kommunizieren. Somit muss der Webservice zwei SOAP-Schnittstellen bereitstellen und darüber hinaus beim Versenden der Nachrichten einen SOAP-Client implementieren.

In der zentralen Klasse des Serverprogramms des Web-Service wurden für die Kernfunktionalitäten die folgenden vier Methoden implementiert:

- *testConnection*: Diese Methode dient als „Echo“-Funktion der Funktionskontrolle der SOAP-Verbindung. Ein übergebener String wird als Antwort zurückgegeben.
- *sendAct*: Diese Methode nimmt Kontakt zu einem anderen Webservice auf und übergibt diesem eine Sprechaktnachricht. *recAct*: Über diese Schnittstelle kann der Client die für ihn bestimmten Nachrichten abholen. Anhand einer ID wird die Authentifizierung durchgeführt. Sollte keine Nachricht für den Client vorliegen, wird ihm das mitgeteilt.

- ‚storeAct‘: Hier können Nachrichten für die Clients hinterlassen werden. Dazu wird eine Client ID und die eigentliche Nachricht übergeben.

Die einzelnen Sprechakte werden dem Webservice transparent übergeben. Der Aufbau des Sprechaktes erfolgt im Client.

Es wurde ein graphischer Testclient entwickelt in dem der Benutzer die Möglichkeit hat, einen Sprechakt auszuwählen und mit eigenen Parametern zu versehen (siehe Abbildung 6.21). Der Rahmen wird dem Benutzer dabei komfortabel vorgegeben.

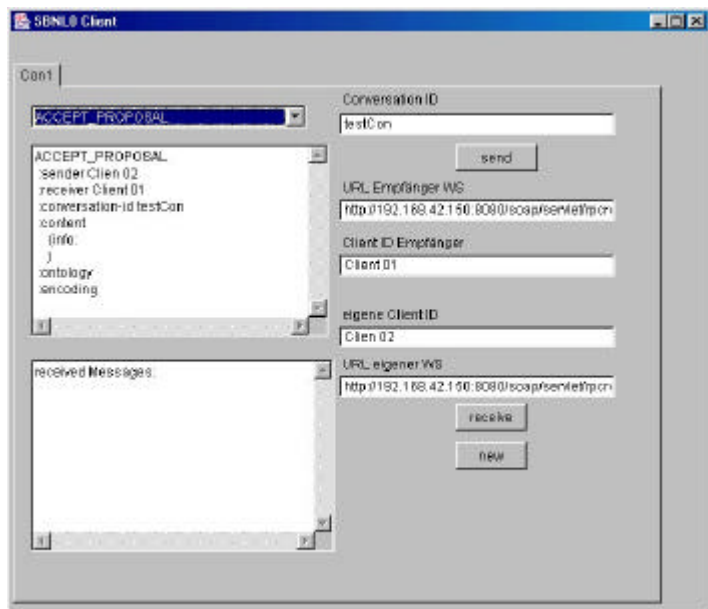


Abbildung 6.21: Testclient des Web-Service für Verhandlungskommunikation

Der prototypische Testclient ist in zwei Bereiche unterteilt: Einem oberen Gebiet zum Senden von Sprechakten und einem unteren zum Empfang der Antwort. Die Eingabe der eigenen und fremden Client ID sowie der „Conversation ID“ werden in die Sprechakte übernommen, um einen gewissen „Komfort“ zu bieten. Ein einfacher Austausch der Sprechakte und damit eine unkomplizierte Verhandlung werden schon mit diesem Client ermöglicht.

Wie in Abschnitt 6.1.2 erläutert wird für jeden Web-Service ein WSDL-Dokument benötigt, mit dem er selbst und seine Schnittstellen beschrieben werden. Für die vorliegenden Prototypen findet sich das WSDL-Dokument im Anhang A.3 zu den Implementationen.

Anhand dieses WSDL-Dokuments kann ein beliebiger Verhandlungspartner für die Kommunikation mit dem Web-Service automatisch angepasst werden.

7 Resümee und Ausblick

Die thematische Kernanforderung dieser Arbeit war Lösungen für eine systemtechnische Infrastruktur für die *Verhandlungskommunikation* im Electronic Business zu schaffen. Dabei sollten elektronische Verhandlungen in den Handelstransaktionen des Electronic Business umfassend betrachtet werden.

Als Ausgangspunkt wurde dazu der Bereich des *Electronic Business* und der hierin verwendbaren Modellierungsmethoden erschlossen.

Als Ergebnis wurde eine Analyse der *Modellierungsmethoden* im Electronic Business erarbeitet. Diese erlaubt eine Einschätzung der Möglichkeiten zur Modellierung von Verhandlungsaspekten im E-Business. Die *e³-Value-Methode* bietet sich als Grundlage für entsprechende Erweiterungen zur Identifikation und Detaillierung von Verhandlungssituationen an.

Weiterhin konnte als Ergebnis der Untersuchung des Electronic Business der Bereich des *E-Procurement* und der *elektronischen Beschaffung* im Allgemeinen aufgrund der darin konzeptionell häufig auftretenden Verhandlungssituationen als geeigneter Kandidat für die weiteren Untersuchungen identifiziert werden.

Elektronische Verhandlungen sind ein derzeit mit innovativen Methoden technisch unterstütztes Gebiet der Informatik (vgl. (Ströbel, 2003)), das dabei aber noch immer großflächigen Forschungsbedarf aufweist (Rebstock, 2001).

Als zentralen Ausgangspunkt für die vorliegende Arbeit wurde zunächst ein *Basismodell für Verhandlungen* geschaffen und formalisiert, das es ermöglicht, die eigenen Analysen und die bestehende Forschung zusammen zu führen. Damit soll es eine exakte und verständliche Grundlage für den wissenschaftlichen Diskurs im Bereich elektronischer Verhandlungen bieten.

Weiterhin ist das Basismodell Grundlage der später entworfenen systemtechnischen *Infrastrukturkomponenten* und ermöglicht durch Formalisierung die korrekte Ausführung von Verhandlungsprozessen und -kommunikation.

Weitere Elemente der grundlegenden Forschung zu elektronischen Verhandlungen sind *elektronische Unterstützungssysteme*. Zunächst wurde im Rahmen der vorliegenden Arbeit eine *Klassifikation* dieser Systeme vorgenommen, die weitere Entwicklungsmöglichkeiten insbesondere bei prozessorientierten Systemen aufzeigt.

Auf dieser Basis wurde dann ein eigenes System zur *Verhandlungsunterstützung* mit innovativen Funktionalitäten im praktischen Teil der Arbeit umgesetzt.

Ein integraler Bestandteil von Verhandlungen ist die Durchführung der *Kommunikation*. Die betrachteten Verhandlungsakteure sind z.B. Unternehmen, die im Electronic Business tätig sind. Dabei gibt es im B2B-Bereich des Electronic Business vielschichtige Ansätze zur Realisierung des Datenaustauschs für Geschäftsdokumente.

Diese Ansätze wurden analysiert und auf ihre Beziehung zur Verhandlungskommunikation untersucht. Dabei zeigte sich, dass es unter den bisherigen Standards nur wenig Ausgangspunkte für die Unterstützung einer derartigen Verhandlungskommunikation gibt. Dieses Defizit wird im späteren Verlauf der Arbeit noch weiter adressiert.

Als effiziente und funktional umfassende Grundlage für die Unterstützung der Verhandlungskommunikation konnte dann die *Sprechakttheorie* identifiziert werden. Die Sprechakttheorie führt insbesondere zu flexibler Kommunikation, die weiterhin formalisierbar bleibt und damit Grundlage für Implementationen bietet. Die Implementationen von Interpretationssystemen für die Sprechakttheorie sind gut zu modularisieren und damit leicht anpassbar.

Da im Electronic Business im Allgemeinen von einer heterogenen, offenen verteilten Umgebung auszugehen ist, wurden *Ontologien* zur Gewährleistung von *Interoperabilität* bei der Verhandlungskommunikation herangezogen.

Es wurde eine *formale Verhandlungsontologie* auf Basis des vorgehend entwickelten Basismodells vorgeschlagen. Diese wird ergänzt durch ein Konzept zur Generierung einer *Business-Ontologie*, die die relevanten Daten aus dem Bereich des Datenaustauschs im Electronic Business einbringt. Diese Herangehensweise auf Ontologieebene soll auch eine leichtere Integration neuer Standards ermöglichen.

Ebenfalls auf dieser Ebene ist damit ein Konzept zur *Vermittlung* zwischen den verschiedenen heterogenen Standards im Business Messaging verbunden.

Eine weitere Anforderung für die praktische Durchführung von Verhandlungskommunikation im Electronic Business ist die technische *Integration mit dem ‚Business Messaging‘*. Dies ist notwendig, da in den B2B-

Standards die Verhandlungsobjekte während des Nachrichtentransfers repräsentiert werden und die Verhandlungsinteraktionen selbst Interaktionen im Sinne der B2B-Standards sind.

Dafür wurde ein Konzept der entkoppelten Übertragung für offene, verteilte Umgebungen entwickelt, das nur wenige Modifikationen an den bestehenden Systemen erfordert.

Zur weiteren Flexibilisierung von Verhandlungen wurde als Infrastrukturelement ein Konzept für die Verhandlung von Verhandlungsprotokollen auf Basis eines initialen Votings vorgeschlagen. Dieses soll auf individuelle Anforderungssituationen angepasste Verhandlungsprotokolle ermöglichen.

Zur *technischen Umsetzung* der so vorgeschlagenen Konzepte konnten insbesondere die *Agententechnologie* aufgrund des autonomen Charakters der agierenden Agenten und der Unterstützung von *Sprechaktkommunikation* sowie *Web-Services* zur Erhöhung von technischer Interoperabilität als besonders geeignet identifiziert werden.

Schließlich wurde eine *Architektur* für die Verhandlungsinfrastruktur vorgeschlagen und die Rahmenarchitektur sowie die einzelnen Elemente wie Verhandlungsagenten, die B2B-Brokerkomponente, das Bootstrapping/Voting-Protokoll und verschiedene Verhandlungsunterstützungsfunktionen prototypisch umgesetzt. Dabei fanden, um die entwickelte Verhandlungsinfrastruktur in offenen verteilten Umgebungen einsetzen zu können, die Kriterien der *Interoperabilität* und der *Flexibilität* besondere Beachtung.

Die *Interoperabilität* der Systeme wurde durch Techniken wie den Einsatz von Web-Services oder den standardisierten Agentenplattformen gefördert. Auf semantischer Ebene wurde Interoperabilität in flexibler dynamischer Weise durch die Verwendung von *Verhandlungsontologien* gefördert.

Durch die Erhöhung von Interoperabilität können so z.B. leichter mehr Verhandlungspartner gefunden werden, mit der Folge, dass Verhandlungsbedingungen und Ergebnisse verbessert werden.

Erhöhte Interoperabilität führt ebenfalls zur stärkeren Integration der eigenen Systeme im Sinne von ‚Enterprise Application Integration‘ (EAI) und somit zu verbesserten Abläufen und inhaltsreicheren Systemen, da alle Informationen über die Verhandlungsobjekte direkt vorliegen.

Die Erhöhung der *Flexibilität* in Verhandlungssystemen wurde unter Wahrung der Formalisierbarkeit und damit leichteren Implementierbarkeit durch den Einsatz der Sprechaktkommunikation und Darstellung der Inhalte über Ontologien erreicht. Dabei können Verhandlungen frei ablaufen oder sich an den in Ontologien festgelegten Protokollen orientieren.

Das initiale Votingprotokoll ermöglicht die dynamische Einigung über zu verwendende Protokolle.

Diese Flexibilität soll zu einer weitergehenden Akzeptanz der Verhandlungssysteme im Kommunikationsprozess führen, so dass erst im späteren Verlauf auf herkömmlichen Methoden zurückgegriffen wird.

Trotz der Flexibilisierung können weite Teile des Verhandlungsprozesses unterstützt und automatisiert werden und somit effizienter ablaufen.

Über den Fokus dieser Arbeit hinausgehen die Ausarbeitung konkreter Verhandlungsprotokolle und die Gestaltung einer umfassenden Ontologie für B2B-Standards.

Um den Ablauf von elektronischen Verhandlungen zu vereinfachen ist es denkbar, eine breite Sammlung von Vorlagen und Interaktionsmustern für Protokolle zu erstellen und in Form der Verhandlungsentologie in das System zu integrieren.

Für die Vermittlung zwischen den verschiedenen B2B-Standards wurden eine umfassende Ontologie und eine Methodik zu ihrer Erstellung vorgeschlagen. Die beispielhaft eingeführte umfassende B2B-Ontologie konnte exemplarisch im Rahmen der Arbeit erstellt werden.

Anhang

A Kommunikation

A.1 Beispiele zur B2B-Kommunikation

A.1.1 EDIFACT-Beispiel

Als inhaltliches Beispiel dient die folgende Rechnung (EDIFACTORY, 1999):

Fahrradhandel Pedal, Wagerstr. 5, 81549 München					
Huber GmbH Obstgasse 2 81549 München					
München, 02.08.99					
Rechnung: Ihre Bestellung Nr. 00010001 vom 15.07.99 9908001					
<u>Pos</u>	<u>Artikel</u>	<u>Beschreibung</u>	<u>Anzahl</u>	<u>Einzelpreis</u>	<u>Gesamt</u>
1	4711.001	Fahrrad, Damen-	1	750,00	750,00
2	4711.002	Luftpumpe, Stand-	1	19,90	19,90
3	4711.003	Ersatzventil	3	2,50	7,50
Gesamtsumme netto					----- 777,40

Umsatzsteuer 16%	124,38
	=====
zu zahlender Betrag	<u>901,78</u>
Alle Beträge verstehen sich in DEM	

Daraus kann sich die folgende Darstellung im EDIFACT-Format ergeben:

```
UNA:+,?'UNB+UNOA:2+FHPEDAL+HUBERGMBH+990802:1557+990802155
7'UNH+INVOIC0001+INVOIC:D:93A:UN'BGM+380+9908001+9'DTM+3+1
9990802+102'RFF+ON+00010001'DTM+4+19999715:102'NAD+SE++Fah
rradhandelPedal++Wagingerstr.+München++81549'NAD+BY++Huber
GmbH++Obstgasse+München++81549'LIN+1++4711.001'IMD+F++::F
ahrrad,Damen'QTY+47:1:PCE'MOA+66:750'PRI+AAA:750'LIN+2++47
11.002'IMD+F++::Luftpumpe,StandQTY+47:1:PCE'MOA+66:19,9'P
RI+AAA:19,9'LIN+3++4711.003'IMD+F++::Ersatzventil'QTY+47:
3:PCE'MOA+66:7,5'PRI+AAA:2,5'UNS+S'MOA+79:777,4'MOA+124:12
4,38'MOA+128:901,78'TAX+7+VAT+++::16+S'UNT+28+INVOIC0001'
UNZ+1+9908021557'
```

A.2 Kommunikation im EPN-System - Dialogbeispiele

A.2.1 Erfolgreiche Ausführung des Request-Protokolls

Der EPN-Controller-Agent des Systems verlangt vom Message Coordinator, dass dieser sein GUI öffnet:

```
(REQUEST
  :sender (agent-identifier
    :name epn-controller@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc))
  :receiver (set (agent-identifier
    :name epn-mc@jiriki:1099/JADE))
  :content "((action(agent-identifier
    :name epn-mc@jiriki:1099/JADE)(show-gui)))"
  :reply-with R3268251_0
  :language fipa-sl0
  :ontology epn-system
  :protocol fipa-request
  :conversation-id C3268251_1060105867249
)
```

Der Message Coordinator stimmt zu und informiert den EPN-Controller anschließend über die erfolgreiche Ausführung der Handlung:

```
(AGREE
  :sender (agent-identifier :name epn-mc@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc))
  :receiver (set(agent-identifier
    :name epn-controller@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc)))
  :content "((action(agent-identifier
    :name epn-mc@jiriki:1099/JADE)(show-gui)))"
  :reply-with epn-
controller@jiriki:1099/JADE1060105867479
  :in-reply-to R3268251_0
  :language fipa-sl0
  :ontology epn-system
  :protocol fipa-request
  :conversation-id C3268251_1060105867249
)
(INFORM
  :sender (agent-identifier :name epn-mc@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc))
  :receiver (set(agent-identifier
    :name epn-controller@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc)))
  :content "((done(action(agent-identifier
```

```

        :name epn-mc@jiriki:1099/JADE)(show-gui))))"
      :reply-with epn-
controller@jiriki:1099/JADE1060105867519
      :in-reply-to R3268251_0
      :language fipa-sl0
      :ontology epn-system
      :protocol fipa-request
      :conversation-id C3268251_1060105867249
    )

```

A.2.2 Request-Protokoll endet durch Refuse

Der EPN-Controller-Agent des Systems verlangt vom B2B-Adapter, daß dieser sein GUI öffnet:

```

(REQUEST
  :sender (agent-identifier
    :name epn-controller@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc))
  :receiver (set (agent-identifier
    :name epn-b2b-adapter@jiriki:1099/JADE))
  :content "((action (agent-identifier
    :name epn-b2b-adapter@jiriki:1099/JADE)(show-gui))))"
  :reply-with R3483343_0
  :language fipa-sl0
  :ontology epn-system
  :protocol fipa-request
  :conversation-id C3483343_1060106195291
)

```

Der B2B-Adapter weigert sich, die Handlung auszuführen:

```

(REFUSE
  :sender (agent-identifier
    :name epn-b2b-adapter@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc))
  :receiver (set (agent-identifier
    :name epn-controller@jiriki:1099/JADE
    :addresses (sequence http://jiriki:2174/acc)))
  :content "((action (agent-identifier
    :name epn-b2b-adapter@jiriki:1099/JADE)(show-gui))
    (unsupported-function show-gui))"
  :reply-with epn-
controller@jiriki:1099/JADE1060106195331
  :in-reply-to R3483343_0
  :language fipa-sl0
  :ontology epn-system
  :protocol fipa-request
  :conversation-id C3483343_1060106195291
)

```


A.3 WSDL-Dokument eines verhandlungsfähigen Webservices

Im Folgenden ist einerseits als Beispiel für ein WSDL-Dokument, andererseits als Schnittstellen-Spezifikation eines verhandlungsfähigen Webservices die WSDL des im Kontext dieser Arbeit in (Waldau, 2002) entwickelten SBNL0-Prozessors dargestellt.

```
<?xml version = '1.0' encoding = 'windows-1252'?>
<!--Generated by the Oracle9i JDeveloper Web Services WSDL
Generator-->
<!--Date Created: Fri Nov 15 13:24:07 CET 2002-->
<definitions
  name="SBNL0Processor"

  targetNamespace="http://192.168.42.150/sbnl0/SBNL0Processor
.wSDL"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

  xmlns:tns="http://192.168.42.150/sbnl0/SBNL0Processor.wSDL"
  xmlns:ns1="http://ISBNL0Processor.xsd">
  <types>
    <schema
      targetNamespace="http://ISBNL0Processor.xsd"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
    </types>
    <message name="testConnection0Request">
      <part name="Content" type="xsd:string"/>
    </message>
    <message name="testConnection0Response">
      <part name="return" type="xsd:string"/>
    </message>
    <message name="sendAct1Request">
      <part name="urlS" type="xsd:string"/>
      <part name="clID" type="xsd:string"/>
      <part name="urn" type="xsd:string"/>
      <part name="Content" type="xsd:string"/>
    </message>
    <message name="sendAct1Response">
      <part name="return" type="xsd:string"/>
    </message>
    <message name="recAct2Request">
      <part name="clID" type="xsd:string"/>
    </message>
    <message name="recAct2Response">
      <part name="return" type="xsd:string"/>
    </message>
    <message name="storeAct3Request">
      <part name="Content" type="xsd:string"/>
      <part name="clID" type="xsd:string"/>
```

```

</message>
<message name="storeAct3Response">
  <part name="return" type="xsd:string"/>
</message>
<portType name="SBNL0ProcessorPortType">
  <operation name="testConnection">
    <input name="testConnection0Request"
message="tns:testConnection0Request"/>
    <output name="testConnection0Response"
message="tns:testConnection0Response"/>
  </operation>
  <operation name="sendAct">
    <input name="sendAct1Request"
message="tns:sendAct1Request"/>
    <output name="sendAct1Response"
message="tns:sendAct1Response"/>
  </operation>
  <operation name="recAct">
    <input name="recAct2Request"
message="tns:recAct2Request"/>
    <output name="recAct2Response"
message="tns:recAct2Response"/>
  </operation>
  <operation name="storeAct">
    <input name="storeAct3Request"
message="tns:storeAct3Request"/>
    <output name="storeAct3Response"
message="tns:storeAct3Response"/>
  </operation>
</portType>
<binding name="SBNL0ProcessorBinding"
type="tns:SBNL0ProcessorPortType">
  <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="testConnection">
    <soap:operation soapAction="" style="rpc"/>
    <input name="testConnection0Request">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
      </input>
    <output name="testConnection0Response">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </output>
  </operation>
  <operation name="sendAct">
    <soap:operation soapAction="" style="rpc"/>
    <input name="sendAct1Request">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </input>
  </operation>
  <operation name="recAct">
    <soap:operation soapAction="" style="rpc"/>
    <input name="recAct2Request">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </input>
    <output name="recAct2Response">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </output>
  </operation>
  <operation name="storeAct">
    <soap:operation soapAction="" style="rpc"/>
    <input name="storeAct3Request">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </input>
    <output name="storeAct3Response">
      <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    </output>
  </operation>
</binding>
</SBNL0ProcessorBinding>
</SBNL0Processor>

```

```

        </input>
        <output name="sendAct1Response">
          <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
          </output>
        </operation>
        <operation name="recAct">
          <soap:operation soapAction="" style="rpc"/>
          <input name="recAct2Request">
            <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
            </input>
            <output name="recAct2Response">
              <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
              </output>
            </operation>
            <operation name="storeAct">
              <soap:operation soapAction="" style="rpc"/>
              <input name="storeAct3Request">
                <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
                </input>
                <output name="storeAct3Response">
                  <soap:body use="encoded"
namespace="SBNL0Processor"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
                  </output>
                </operation>
              </binding>
              <service name="SBNL0Processor">
                <port name="SBNL0ProcessorPort"
binding="tns:SBNL0ProcessorBinding">
                  <soap:address
location="http://192.168.42.150:8080/soap/servlet/rpcroute
r"/>
                  </port>
                </service>
              </definitions>

```

B Implementationen

B.1 Zentrale Klassen und Schnittstellen des EPN-Frameworks

Im Folgenden werden die zentralen Klassen und Schnittstellen des Frameworks dargestellt. Ihre Aufgaben, Abhängigkeiten und Verwendung werden beschrieben, um ein genaues Verständnis der Funktionsweise des Frameworks zu vermitteln und so die weitere Entwicklung einzelner Systemkomponenten zu ermöglichen.

Die in diesem Abschnitt angeführten Auszüge aus dem Quellcode wurden nicht immer exakt übernommen, sondern meist gekürzt und vereinfacht, um so die grundsätzliche Funktionsweise besser verdeutlichen zu können.

B.1.1 Die Klasse `epn.Boot`

Klasse	<code>epn.Boot</code>
Aufgabe	Start des EPN-Systems
Siehe auch	<code>epn.system.EPNController</code>

Die Klasse `epn.Boot` steuert den Startvorgang des EPN-Systems, der durch den Aufruf der `main(String[] args)` Methode erfolgt. Es wird zunächst die per Kommandozeilenparameter bestimmte Konfiguration des Systems eingelesen, aus der ein entsprechendes Profil erzeugt wird. Auf Basis dieses Profils wird anschließend eine JADE Agentenplattform gestartet, die als Rahmen für das gesamte EPN-System dient, und, falls der Parameter `-rma` übergeben wurde, das Remote Agenten Management GUI geöffnet:

```
Runtime rt = Runtime.instance();
Profile pMain = new EPNProfileImpl(host, rmiPort,
httpPort);
MainContainer mc = rt.createMainContainer(pMain);
if (showRMA) {
```

```

    AgentController rma = mc.createNewAgent(
        "rma", "jade.tools.rma.rma", new Object[0]
    );
    rma.start();
}

```

Nachdem die Agentenplattform bereitsteht, wird der zentrale EPN-Controller-Agent gestartet, dem zunächst die Aufgabe zukommt, alle weiteren Systemagenten zu initialisieren:

```

Agent controller = (Agent)mc.createNewAgent(
    EPN_SYSTEM_AGENT_CONTROLLER,
    EPNController.class.getName(),
    new Object[] { startUpLatch, mc, new Boolean(showGUI)
});
controller.start();

```

B.1.2 Die Klasse `epn.system.EPNController`

Klasse	<code>epn.system.EPNController</code>
Aufgabe	Start des EPN-Systems
Siehe auch	<code>epn.Boot</code> <code>epn.system.SimpleEPNAgent</code> <code>epn.system.mc.MessageCoordinator</code> <code>epn.system.b2badapter.B2BAdapter</code>

Die Aufgabe des `EPNController`-Agenten besteht darin, die anderen `EPNSystemagenten` zu starten und zu überwachen. In jeder Instanz des `EPNSystems` ist ein solcher Agent aktiv. Er wird während des Startvorgangs von der Klasse `epn.Boot` initialisiert und führt zunächst sein `StartupBehaviour` aus, durch das die verbleibenden Systemagenten gestartet werden. Dies sind beispielsweise der `MessageCoordinator` und der `B2BAdapter`:

```

try {
    container.createNewAgent(
        EPN_SYSTEM_AGENT_B2BA,
        "epn.system.b2badapter.B2BAdapter", null
    ).start();
    container.createNewAgent(
        EPN_SYSTEM_AGENT_MC,
        "epn.system.mc.MessageCoordinator", null
    ).start();
} catch (Exception e) {
    System.err.println(
        "Exception while starting EPN System Agents: " + e
    );
    e.printStackTrace();
}

```

Der EPNController verfügt über ein GUI, durch das sich die Systemagenten überwachen und steuern lassen. Diese registrieren sich als solche beim DF und stellen einen gemeinsamen Satz einfacher Dienste bereit. Mit Hilfe des EPNController können die Systemagenten beispielsweise aufgefordert werden, ihr eigenes GUI zu öffnen, Auskunft über ihren Status zu geben oder sich zu beenden.

B.1.3 Die Klasse `epn.system.mc.MessageCoordinator`

Klasse	<code>epn.system.mc.MessageCoordinator</code>
Aufgabe	Koordination des Nachrichtenversands
Siehe auch	<code>epn.system.SimpleEPNAgent</code> <code>epn.system.EPNAgentBehaviour</code>

Der MessageCoordinator ist einer der EPN-Systemagenten. Seine Aufgabe besteht darin, den je nach Kontext getrennten oder gemeinsamen (genauer: koordinierten) Versand von Verhandlungs- und E-Business-Nachrichten zu steuern. Er bildet die Schnittstelle zwischen den Verhandlungsagenten und dem B2BAdapter.

B.1.4 Die Klasse `epn.system.b2badapter.B2BAdapter`

Klasse	<code>epn.system.b2badapter.B2BAdapter</code>
Aufgabe	Transformation von E-Business-Nachrichten verschiedener Standards
Siehe auch	<code>epn.system.SimpleEPNAgent</code> <code>epn.system.EPNAgentBehaviour</code>

Der B2BAdapter ist einer der EPN-Systemagenten. Seine Aufgabe besteht darin, Nachrichten verschiedener E-Business-Standards zwischen ihrer externen und ihrer gemeinsamen internen Darstellung zu transformieren.

B.1.5 Die Schnittstelle `epn.system.EPNAgent`

Klasse	<code>epn.system.EPNAgent</code>
Aufgabe	Gemeinsame Schnittstelle aller EPNSystemagenten
Siehe auch	<code>epn.system.SimpleEPNAgent</code> <code>epn.system.EPNAgentBehaviour</code>

Diese Schnittstelle muss von allen EPN-Systemagenten implementiert werden. Die hier definierten Methoden bilden die allgemeinen Dienste

eines Systemagenten ab. Dies ermöglicht, die Kommunikation zwischen den Systemagenten vollständig im EPNAgentBehaviour zu kapseln. Somit genügt es einem Systemagenten, diese Methoden zu implementieren und das EPNAgentBehaviour zu instanziiieren, um den internen Kommunikationsstandard des EPN-Systems zu erfüllen:

```
public boolean guiAvailable();
public boolean guiIsVisible();
public boolean guiSetVisible(boolean b);
public String getStatus();
public boolean shutdown();
```

Die Methoden guiAvailable() und guiIsVisible() müssen jeweils genau dann true zurückgeben, wenn der Agent über ein GUI verfügt bzw. wenn dieses gerade sichtbar ist. Die Sichtbarkeit des GUI wird über die Methode guiSetVisible(boolean b) gesteuert, deren Rückgabewert Auskunft über den Erfolg der Operation gibt.

Die Methode getStatus() eines Systemagenten sollte in Menschenlesbarer Form Auskunft über den Status des Agenten geben und die Methode shutdown() den Agenten veranlassen, sich selbst zu beenden. Ist letzteres zum gegebenen Zeitpunkt nicht möglich, so gibt die Methode false zurück, sonst true.

B.1.6 Die Klasse epn.system.EPNAgentBehaviour

Klasse	epn.system.EPNAgentBehaviour
Aufgabe	Allgemeines Verhalten aller EPNSystemagenten
Siehe auch	epn.system.SimpleEPNAgent epn.util.RequestResponder epn.system.EPNAgent

Diese Klasse implementiert das gemeinsame Verhalten, dem alle EPN-Systemagenten genügen müssen. Dabei handelt es sich beispielsweise um die Einhaltung des internen Kommunikationsstandards, über den die allgemeinen Dienste der Systemagenten bereitgestellt werden. Bei der EPNAgentBehaviour-Klasse handelt es sich um einen RequestResponder, der einen bestimmten Satz von Aktionen unterstützt, und die Kommunikation erfolgt auf Basis der im Paket epn.ontology.system definierten EPN-Systemontologie (siehe oben). Beides wird im Konstruktor der Klasse festgelegt:

```
super(agent, codec,
```

```
epn.ontology.EPNSystemOntology.getInstance() );
addAction( new epn.ontology.ReportStatus() );
addAction( new epn.ontology.ShowGUI() );
addAction( new epn.ontology.HideGUI() );
```

Soll das interne Kommunikationsprotokoll erweitert werden, so müssen die wiederum in der Systemontologie definierten zusätzlichen Aktionen in dieser Klasse auf die entsprechenden Methoden der EPNAgent-Schnittstelle abgebildet werden. D.h. die Aktionen müssen im Konstruktor hinzugefügt und die Methoden `agreeAction` und `handleAction` entsprechend angepaßt werden.

B.1.7 Die Klasse `e pn.system.SimpleEPNAgent`

Klasse	<code>e pn.system.SimpleEPNAgent</code>
Aufgabe	Generische Oberklasse für EPNSystemagenten
Siehe auch	<code>e pn.system.EPNAgentBehaviour</code> <code>e pn.system.EPNAgent</code>

Die Klasse `SimpleEPNAgent` implementiert die Schnittstelle `EPNAgent` und erledigt in generischer Weise die Aufgaben, die jeder EPN-Systemagent erfüllen muss. Darüber hinaus stellt sie nützliche Methoden für den Zugriff auf den DF und für die Kommunikation mit anderen Agenten bereit. Die eigentlichen Systemagenten sollten in der Regel direkt oder indirekt von dieser Klasse erben.

Wird ein `SimpleEPNAgent` erzeugt, so nimmt er das `EPNAgentBehaviour` an. Die durch dieses Verhalten gesteuerten Methoden der EPNAgent-Schnittstelle werden in minimaler Weise implementiert und müssen von den Systemagenten bei Bedarf entsprechend überschrieben werden. Der `SimpleEPNAgent` ist so in der Lage, den `e pn-system`-Dienst anzubieten. Die Methode `getServiceDescriptions()` des Agenten liefert eine Liste von Dienstbeschreibungen, die zunächst nur die Beschreibung des `e pn-system`-Dienstes enthält. Bietet ein Systemagent weitere Dienste an (was die Regel ist), so muss er diese Methode überschreiben, der Liste die Beschreibungen seiner Dienste hinzufügen und die gesamte Liste zurückgeben. Beim Start des Agenten registriert sich dieser mit der so erzeugten Liste seiner Dienste beim DF:

```

DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription[] desc = getServiceDescriptions();
for (int i=0; i<desc.length; i++) {
    dfd.addServices( desc[i] );
}
try {
    DFService.register(this,dfd);
} catch (FIPAException e) {
    [...]
}

```

Die Klasse `SimpleEPNAgent` stellt darüber hinaus einige Methoden bereit, die die Suche im DF nach anderen Agenten unterstützen. Zusammen

mit der DFSearch Schnittstelle ermöglichen sie in komfortabler Weise asynchrone Anfragen an den DF, deren Ergebnisse per Callback verarbeitet werden. Über die statischen Varianten dieser Methoden steht diese Funktion auch anderen Agenten zur Verfügung. Folgendes Code-Beispiel veranlasst den DF, eine Liste aller Agenten zu liefern, die den epn-system-Dienst anbieten:

```

DFAgentDescription tmp1 = new DFAgentDescription();
tmp1.addServices(
SimpleEPNAgent.getEPNSystemServiceDescription()
);
SimpleEPNAgent.searchDF( myAgent, new DFSearch() {
public void dfSearchResult(DFAgentDescription[] result) {
// Bearbeite das Ergebnis hier...
}
}, tmp1);

```

B.1.8 Die Schnittstelle epn.system.Constants

Klasse	Interface epn.system.EPNAgent
Aufgabe	Aufgabe Konstanten zur Verwendung in den EPN-Systemklassen
Siehe auch	

Diese Schnittstelle enthält Konstanten, die im globalen Kontext des EPN-Systems benötigt werden. Die Konstanten sind statisch deklariert, so dass über die Schnittstelle auf sie zugegriffen werden kann. Für Klassen, in denen diese Konstanten sehr häufig Verwendung finden, kann es sinnvoll sein, diese Schnittstelle zu implementieren, um direkt auf die Werte zugreifen zu können.

B.1.9 Die Klasse epn.util.RequestResponder

Klasse	epn.util.RequestResponder
Aufgabe	Generische Participant-Rolle
Siehe auch	epn.system.EPNAgentBehaviour

Der RequestResponder ist eine Behaviour-Klasse, die die Participant-Rolle des FIPA Request Interaction Protocols (FIPA, 2002e) generisch abbildet. Der Participant einer diesem Protokoll genügenden Interaktion hat auf eine request-Nachricht zu reagieren, deren Proposition eine Handlung eines Agenten ist.

Die Klasse automatisiert die Behandlung solcher Nachrichten in einer bestimmten Inhaltssprache, mit Bezug auf eine bestimmte Ontologie und mit zuvor definierten Handlungen aus dieser Ontologie. Inhaltssprache

und Ontologie werden bei der Instanziierung der Klasse spezifiziert. Anschließend werden mit der Methode `addAction(AgentAction action)` Handlungen definiert, auf die das Behaviour reagieren soll. Durch Überschreiben der Methoden `agreeAction` und `handleAction` wird das Verhalten des Agenten bei Erhalt einer „passenden“ Nachricht bestimmt. Der Rückgabewert der Methode `agreeAction` erlaubt es ihm, der Aufforderung zur Handlung zuzustimmen oder diese abzulehnen. Wird die Methode nicht überschrieben, so gibt sie `true` zurück, d.h. der Agent stimmt zu, die entsprechende Handlung auszuführen. Wird sie überschrieben und gibt `false` zurück, so muss mittels `setReason` eine Begründung für die Weigerung angegeben werden, um die Antwort FIPA-konform zu halten. Bei einer solchen Begründung handelt es sich um ein Prädikat aus der betreffenden Ontologie; im Falle der `epn-system` Ontologie stehen hierfür insbesondere jene aus der Oberontologie `fipa-exception` bereit (siehe oben). Die Methode `handleAction` wird nur aufgerufen, wenn der Agent der Handlung zugestimmt hat, und er sollte sie an dieser Stelle ausführen. Gelingt dies, so muss `true` zurückgegeben werden und `false`, falls die Handlung fehlschlägt. Auch hier muß ggf. mit `setReason` ein Grund für das Scheitern angegeben werden.

Folgendes Code-Beispiel veranlasst den Agenten `myAgent`, auf die Aufforderung zur Handlung `SayHelloWorld` mit der Ausgabe einer Zeichenfolge zu reagieren. Der Inhalt der `request`-Nachrichten muß in `SL0` (FIPA, 2002f) verfasst sein und sich auf die Ontologie `MyOntology` beziehen:

```
RequestResponder resp = new RequestResponder(
    myAgent, new jade.content.lang.sl.SLCodec(0),
    MyOntology.getInstance()
){
    public boolean handleAction(
        Action action, ACLMessage response
    ) {
        if (action.getAction() instanceof SayHelloWorld) {
            System.out.println("Hello world!");
            return true;
        } else {
            return false;
        }
    }
};
resp.addAction( new ontology.SayHelloWorld() );
myAgent.addBehaviour( resp );
```

Wenn der Scheduler des Agenten den `RequestResponder` aufruft, so liest er eine erhaltene Nachricht, falls eine solche vorhanden ist, aus der Message-Queue des Agenten. Entspricht sie in irgendeiner Weise nicht den für ihn definierten Kriterien, ist er also nicht in der Lage, diese Nachricht zu behandeln, so schiebt er sie zunächst zurück in die Message-Queue. Liest er zu einem späteren Zeitpunkt jedoch dieselbe Nachricht erneut ein,

so wird sie von ihm verworfen. Da der Behaviour-Scheduler der JADE Plattform im round-robin Verfahren arbeitet, wird auf diese Weise sichergestellt, dass auch alle anderen Behaviours die Gelegenheit erhalten, die Nachricht zu bearbeiten, bevor sie verworfen wird. Somit ist es einem Agenten möglich, unabhängig voneinander beliebig viele Instanzen des RequestResponders zu betreiben, die jeweils auf bestimmte Handlungen spezialisiert sind.

Literatur

- Adwiraah, A.: "Negotiation Support Systems im B-to-B Electronic Commerce", Fachbereich Informatik, Universität Hamburg, Hamburg, Studienarbeit, 2003.
- Agha, G.: "ACTORS: A Model for Concurrent Computation in Distributed Systems", MIT Press, Cambridge, 1986.
- Alt, R.; Zimmermann, H.-D.: "Introduction to Special Section - Business Models", Electronic Markets, 11 (1), 2001.
- Arms, W. Y.: "Digital libraries", MIT Press, Cambridge, 2000.
- Auer, P.: "Sprachliche Interaktion - Eine Einführung anhand von 22 Klassikern", Tübingen, 1999.
- Austin, J. L.: "Zur Theorie der Sprechakte (How to do things with Words)", Reclam, Stuttgart, 1962.
- Ayala, D.; Browne, C.; Chopra, V.; Sarang, P.; Apshankar, K.; McAllister, T.: "Professional Open Source Web Services", Wrox Press Ltd., Birmingham, 2002.
- Bakos, J. Y.: "Reducing Buyer Search Costs: Implications for Electronic Marketplaces", Management Science, 43 (12), 1997.
- Ballmer, T.; Brennstuhl, W.: "Speech Act Classification: A study in the Lexical Analysis of English Speech Activity Verbs", Springer-Verlag, Berlin, Heidelberg, New York, 1981.
- Banerjee, S.; Kumar, R. L.: "Managing Electronic Interchange of Business Documents", Communications of the ACM, 45 (7), 96-102, 2002.
- Barbu, A.: "Agententechnologie im Electronic Commerce - Ein Überblick", Fachbereich Informatik, Abteilung Informationssysteme, Universität Oldenburg, Oldenburg, Studienarbeit, 1999.
- Bartelt, A.; Faensen, D.; Faulstich, L.; Schallehn, E.; Zirpins, C.: "Building Infrastructures for Digital Libraries", in: DELOS Workshop on Interoperability in Digital Libraries, ERCIM Workshop Proceedings, 2001a.
- Bartelt, A.; Lamersdorf, W.: "Agent-Oriented Concepts to Foster the Automation of e-Business", in: 11th International Workshop on Database and Expert Systems (DEXA 2000), London, IEEE, Los

- Alamitos, California; Washington; Brussels; Tokyo, 775-779, 2000a.
- Bartelt, A.; Lamersdorf, W.: "Geschäftsmodelle des Electronic Commerce: Modellbildung und Klassifikation", in: Verbundtagung Wirtschaftsinformatik 2000, Siegen, Shaker, Aachen, 17-29, 2000b.
- Bartelt, A.; Lamersdorf, W.: "A Multi-Criteria Taxonomy of Business Models in Electronic Commerce", in: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), WS on Electronic Commerce, Heidelberg, Springer-Verlag, 193-205, 2001b.
- Bartelt, A.; Meyer, J.: "A Practical Guideline to the Implementation of Online Shops", in: 18th IEEE Symposium on reliable distributed systems (SRDS'99), WELCOM'01, Lausanne, Switzerland, IEEE Computer Society Press, 348-353, 1999.
- Bartelt, A.; Weinreich, H.; Lamersdorf, W.: "Kundenorientierte Aspekte der Konzeption von Online-Shops", in: GeNeMe2000, Virtuelle Organisation und Neue Medien 2000, Dresden, Josef Eul Verlag, 159-172, 2000c.
- Bartelt, A.; Zirpins, C.; Fahrenholtz, D.: "Geschäftsmodelle der Electronic Information: Modellbildung und Klassifikation", in: Informatik 2001 - Wirtschaft und Wissenschaft in der Network Economy, Wien, Deutsche Gesellschaft für Informatik (GI) und Österreichische Computer Gesellschaft (OCG), 902-908, 2001c.
- Bayer; Kühn; Junginger; Petzmann: "E-BPMS: Ein Modellierungs-Framework für E-Business-Anwendungen", in: Informatik 2001, Wien, 2001.
- Beam, C.; Segev, A.: "Automated negotiations: a survey of the state of the art", CMIT, <http://groups.haas.berkeley.edu/citm/publications/papers/wp-1022.pdf>, Abruf am: 16.01.2003, 1997.
- Beam, C.; Segev, A.; Bichler, M.; Krishnan, R.: "On Negotiations and Deal Making in Electronic Markets", Information Systems Frontier, 1 (3), 241-258, 1999.
- Bettag, U.: "Web-Services", Informatik Spektrum, 302-304, 2001.
- Bichler, M.: "The Future of e-Markets: Multi-Dimensional Market Mechanisms", Cambridge University Press, Cambridge, UK, 2001.
- Bichler, M.; Kalagnanam, J.; Katircioglu, K.; King, A.; Lawrence, D.; Lee, H. S.; Lin, G.; Lu, Y.: "Applications of Flexible Pricing in Business-to-Business Electronic Commerce", IBM Systems Journal, (02), 2002.
- Bichler, M.; Kaukal, M.; Segev, A.: "Multi-attribute auctions for electronic procurement", in: 1st IBM IAC Workshop on Internet Based Negotiation Technologies, Yorktown Heights, 1999.
- BMEcat: "BMEcat - eBusiness Standardization Committee", BMEcat, <http://www.bmecat.org>, Zugriff am 28.06.2003, 2003.

- Böge, S.: "Konzeption und Realisierung eines prototypischen interoperablen E-Procurement Systems", Fachbereich Informatik, VSIS, Universität Hamburg, Hamburg, Diplomarbeit, 2003.
- Borgman, C. L.: "From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World", MIT Press, Cambridge, 2000.
- Borgman, C. L.: "Massaging the Message", www.research.ucla.edu/chal/99/making?sense/article03.htm, Zugriff am January 2002, 2001.
- Brockhaus: "Brockhaus - Die Enzyklopädie in 24 Bänden", F. A. Brockhaus, Leipzig, Mannheim, 1996-99.
- BYU-Marketing-Department: "Conjoint Analysis Tutorial: Designing Your Conjoint Analysis Study", BYU Marketing Department, <http://marketing.byu.edu/htmlpages/tutorials/conjoint.htm>, Zugriff am 10. Oktober 2002, 2002.
- Caire, G.: "JADE Tutorial: Application-Defined Content Languages and Ontologies", TILab, <http://jade.cselt.it/doc/CLOntoSupport.pdf>, 2002.
- Ceri, S.; Fraternali, P.; Bongio, A.; Brambilla, M.; Comai, S.; Materna, M.: "Designing Data-Intensive Web Applications", Morgan Kaufmann, Amsterdam u.a., 2003.
- Chang, M. K.; Woo, C. C.: "A Speech-Act-Based Negotiation Protocol: Design, Implementation, and Test Use", ACM Transactions on Information Systems, 12 (4), 360-382, 1994.
- Conallen, J.: "Building Web Applications with UML", Addison Wesley, Boston u.a., 2002.
- Crantz, C.: "E-Business und Prozessmodellierung am Beispiel von E-Procurement Systemen", Fachbereich Informatik, VSIS, Universität Hamburg, Hamburg, Bakkalaureatsarbeit, 2002.
- cXML: "cXML.org Commerce XML Resources", <http://www.cxml.org>, Zugriff am 25.06.03, 2003.
- Davies, J.; Fensel, D.; van Harmelen, F. (eds.): "Towards the semantic web, ontology-driven knowledge management", Wiley, 2003.
- Davis, R.; Smith, R. G.: "Negotiation as a Metaphor for Distributed Problem Solving", Artificial Intelligence, 20, 63-109, 1983.
- De Paula, G. E.; Ramos, F. S.; Ramalho, G. L.: "Bilateral Negotiation Model for Agent Mediated Electronic Commerce", in: Workshop on Agent Mediated Electronic Commerce III, 4th Intl. Conf. on Autonomous Agents, Barcelona, 2000.
- DiLeo, J.; Jacobs, T.; DeLoach, S.: "Integrating ontologies into multi-agent systems engineering", 2002.
- Dittrich, H.: "Konzeption und Realisierung eines hoch integrativen Online-Auktionssystems", Fachbereich Informatik, Abteilung Informationssysteme, Universität Oldenburg, Oldenburg, Diplomarbeit, 2000.
- Doan, A.; Domingos, P.; Halevy, A.; .., P. o. t.: "Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach", in: ACM SIGMOD Conf. on Management of Data (SIGMOD-2001), 2001.

- dobney.com: "Conjoint Analysis", Internet, <http://dobney.com/>, Zugriff am 14. April 2003, 2003.
- Dolmetsch, R.: "E-Procurement - Einsparungspotentiale im Einkauf", Addison-Wesley, Boston u.a., 2000.
- Doorenbos, R. B.; Etzioni, O.; Weld, D. S.: "A scalable comparison-shopping agent for the world-wide web", in: Agents '97, 1997.
- Dürr, M.; Schlobinski: "Einführung in die deskriptive Linguistik", Westdeutscher Verlag GmbH, Opladen, 1994.
- Eberhart, A.; Fischer, S.: "Java-Bausteine für E-Commerce-Anwendungen, Verteilte Anwendungen mit Servlets, CORBA und XML", Hanser, München, Wien, 2000.
- ebXML: "Electronic Business using eXtensible Markup Language", <http://www.ebxml.org/specs/index.htm>, Zugriff am 25.06.03, 2003.
- ebXML-Initiative: "ebXML - Enabling a Global Electronic Market", Website, <http://www.ebxml.org/>, Zugriff am 15. Mai 2003, 2003.
- EDIFACTORY: "UN/EDIFACT Beispielnachricht", EDIFACTORY, <http://www.edifactory.de/edifact/ediexam1.html>, Zugriff am 20. Juni 2003, 1999.
- Ehtamo, H.; Härmäläinen, R. P.: "Joint Gains - Negotiation Support in the Internet", Computer software, Systems Analysis Laboratory, Helsinki University of Technology, Finland, <http://www.jointgains.hut.fi>, Zugriff am 14. December 2002, 1999.
- Ehtamo, H.; Härmäläinen, R. P.: "Negotiation Analysis eLEARNING SITE", <http://www.negotiation.hut.fi/index.html>, Zugriff am 14. December 2002, 2002.
- EIDX: "EIDX - B2B Content Standards", Electronics Industry Data Exchange Association, <http://www.eidx.org/publications/b2b-standards.html>, Zugriff am 25.06.03, 2003.
- Endres, A.; Fellner, D. W.: "Digitale Bibliotheken: Informatiklösungen für globale Wissensmärkte", dpunkt.verlag, Heidelberg, 2000.
- E-Negotiations-Community: "E-Negotiations Community", enegotiations.wu-wien.ac.at, Zugriff am 14. Februar 2001, 2002.
- Eymann, T.: "Digitale Geschäftsagenten, Softwareagenten im Einsatz", Springer, Berlin, Heidelberg, New York u.a., 2003.
- Finin, T.; Weber, J.: "Specification of the KQML Agent-Communication Language", DARPA Knowledge Sharing Initiative External Interfaces Working Group, Technical Report, 1993.
- FIPA: "FIPA - The Foundation for Intelligent Physical Agents", <http://www.fipa.org>, Zugriff am 2002 2002, 2002a.
- FIPA: "FIPA ACL Message Structure Specification", FIPA - The Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00037/>, 2002b.
- FIPA: "FIPA Agent Management Specification", FIPA - The Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00023/>, 2002c.

- FIPA: "FIPA Communicative Act Library Specification", FIPA - The Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00037/>, 2002d.
- FIPA: "FIPA Request Interaction Protocol Specification", FIPA - The Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00026/>, 2002e.
- FIPA: "FIPA SL Content Language Specification", FIPA - The Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00008/>, 2002f.
- FIPA: "XC00001K - FIPA Abstract Architecture Specification", FIPA - Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00001/XC00001K.pdf>, 2002g.
- Fitzgerald, M.: "Building B2B Applications with XML", John Wiley & Sons, Inc., New York, 2001.
- Fletcher-Forum: "The Peacemaker: An Interview with UN Under-Secretary-General Alvaro de Soto", The Fletcher Forum of World Affairs, 26 (1), 61-67, 2002.
- Fowler, M.: "Patterns of Enterprise Application Architecture", Addison Wesley, Boston u.a., 2003.
- Frank, U.: "Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze", Wirtschaftsinformatik, 43 (3), 283-293, 2001.
- Franklin, S.; Graesser, A.: "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents." in: Intelligent Agents III: Agent Theories, Architectures, and Languages, ECAI Workshop (ATAL), Budapest, Hungary, Springer, 1996.
- Geerts, G. L.; McCarthy, W. E.: "The Ontological Foundation of REA Enterprise Information Systems", The University of Delaware / Michigan State University, Working Paper, <http://www.msu.edu/user/mccarth4/paplist1.html>, 2000.
- Genesereth, M. R.; Fikes., R. E.: "Knowledge Interchange Format, Version 3.0 Reference Manual", Stanford, CA, USA, Technical Report Logic-92-1, 1992.
- Genesereth, M. R.; Nilsson, N. J.: "Logical Foundations of Artificial Intelligence", Morgan Kaufmann, Amsterdam u.a., 1987.
- GigaTS: "GigaTS RSD", GigaTS, <http://www.telin.nl/NetworkedBusiness/GigaTS/ENindex.htm>, Zugriff am 5. Dezember 2002, 2002.
- Gordijn, J.: "Value based requirements engineering: Exploring innovative e-commerce ideas", Faculty of Sciences, Vrije Universiteit, Amsterdam, Netherlands, PhD. thesis, <http://www.gordijn.org/>, 2002.
- Gräfe, C.: "Eine sprechaktbasierte Kommunikationssprache zur Modellierung und Implementation von Verhandlungsprotokollen", Fachbereich Informatik, Universität Hamburg, Hamburg, Diplomarbeit, 2003.
- Gramer, I.: "Einführung in die Sprechakttheorie", Albert-Ludwigs-Universität Freiburg, <http://omnibus2.ruf.uni->

- freiburg.de/~kelle/h1gramer.htm, Referat, Zugriff am 03. Dezember 2002, 2001.
- Grice, P.: "Meaning", *Philosophical Review*, 377-388, 1957.
- Gruber, T. R.: "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases", in: *Second International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, Amsterdam u.a., 1991.
- Gruber, T. R.: "A Translation Approach to Portable Ontology Specifications", *Knowledge Acquisition*, (5), 1993.
- Gruninger, M.; Lee, J.: "Introductino to SPECIAL ISSUE: Ontology applications and design", *Communications of the ACM*, 45 (2), 39-41, 2002.
- Guttman, R. H.; Moukas, A. G.; Maes, P.: "Agents as mediators in electronic commerce", *Electronic Markets*, 8 (1), 22-27, 1998.
- Habermas, J.: "Theorie des kommunikativen Handelns, Teil 1: Handlungsrationalität und gesellschaftliche Rationalisierung", Suhrkamp, Frankfurt am Main, 1981.
- Haft, F.: "Verhandlung und Mediation - Die Alternative zum Rechtstreit", Verlag C. H. Beck, München, 2000.
- Hanne, T.: "Evolutionäre Algorithmen für multikriterielle Optimierung", Sonderforschungsbereich Computational Intelligence, Universität Dortmund, Dortmund, 1999.
- Hill, N. C.: "Electronic Data Interchange: A Definition and Perspective", *The Journal of Electronic Commerce*, 1989.
- Huhns, M. N.; Stephens, L. M.: "Multiagent Systems and Societies of Agents", in: Weiss, G. ed, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge,, 1999.
- Jennings, N. R., K. ; Sycara, al., e.: "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-Agent Systems*, 1, 275-306, 1998.
- Jones, D.; Bench-Capon, T.; Visser, P.: "Methodologies for Ontology Development", in: *T&KNOWS Conference, XV IFIP World Computer Congress*, Budapest, 1995.
- Kersten, G.; Noronha, S.: "Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution", *Interneg, Research Report*, <http://interneg.org/interneg/research/papers/1999/02.pdf>, 1999.
- Kersten, G. E.; Lo, G.: "Negotiation Support Systems and Software Agents in E-Business Negotiations", in: *The First International Conference on Electronic Business*, Hong Kong, 2001.
- Kersten, G. E.; Noronha, S. J.; Teich, J.: "Are All E-Commerce Negotiations Auctions?" in: *COOP'2000 - Fourth International Conference on Design of Cooperative Systems*, Sophia-Antipolis, France, 2000.
- Kettunen, E.: "Interactive Methods for Group Decision and Negotiation Support", *University of Technology, Helsinki, Report*, 1999.

- Kifer, M.; Lausen, G.; Wu, J.: "Logical Foundations of Object Oriented and Frame Based Languages", *Journal of ACM*, 42, 741-843, 1995.
- Klemperer, P.: "Auction Theory: A Guide to the Literature", *Journal of Economic Surveys*, 13 (3), 227-286, 1999.
- Knollmann, J.: "Analyse und Demonstration des Bootstrapping für elektronische Verhandlungssysteme", Fachbereich Informatik, Universität Hamburg, Hamburg, Studienarbeit, 2003.
- Kolonko, M.; Engelhardt-Funke, O.; ". In: , S. . "Mathematische Optimierung in der Praxis - Genetisch optimierte Fahrpläne und Kosten-Nutzen-Analysen für Verkehrsnetze", Technische Universität Clausthal TUC Contact, (10), 31-34, 2002.
- Kropp, A.: "bme.de - das Portal für über 80 Prozent des deutschen Beschaffungsvolumens", *Oracle Magazine - Oracle E-Business Review*, www.oracle.com/oramag, www.oracle.com/de, 2002.
- Kruchten, P.: "The Rational Unified Process: an introduction", Addison Wesley, 2000.
- Kuschke, M.; Wölfel, L.: "Web Services als E-Business-Evolution", iX, 2001.
- Laka, W.: "Prototypische Realisierung von Business Messaging Standards für Portalapplikationen am Beispiel von cXML und E-Stores", Fachbereich Informatik, VSIS, Universität Hamburg, Hamburg, Studienarbeit, 2002.
- Langham, M.; Ziegeler, C.: "Cocoon: Building XML Applications", New Riders Publishing, Boston u.a., 2002.
- Lankhorst, M.; Luttighuis, P. O.; Janssen, W.: "Syntax Amber, Testbed/Notitie/WP4/N014/V6.0", Telematica Institut, 1999.
- Lo, G.; Kersten, G. E.: "Negotiation in Electronic Commerce: Integrating Negotiation Support and Software Agent Technologies", in: 29th Atlantic Schools of Business Conference, Halifax, NS, 1999.
- Losbichler, H.; Oberlehner, G.: "Prozesskosten in der Beschaffungslogistik - Anspruch und Wirklichkeit", *industrie management*, (4), 47-50, 2002.
- Masseti, B.; Zmud, R. W.: "Measuring the extend of EDI usage in complex organizations: Strategies and illustrative examples", *MIS Quarterly*, 331-345, zitiert in Snehamay Banerjee and Ram L. Kumar, *Comm. ACM*, July 2002, Vol. 45, No. 7, 1996.
- McAfee, R. P.; McMillian, J.: "Auctions and Bidding", *Journal of Economic Literature*, 25, 699-738, 1987.
- McGrath, J. E.; Hollingshead, A. B.: "Groups interacting with technology", Sage Publications, Inc., 1994.
- Melnik, S.; Decker, S.: "A Layered Approach to Information Modeling and Interoperability on the Web", Database Group, Stanford University, 2000.
- Merriam-Webster: "Merriam-Webster Online, The Language Center", Merriam-Webster Incorporated, <http://www.m-w.com/>, Zugriff am 2003, 2003.
- Merz; Griffel; Tu; Müller-Wilken; Weinreich; Boger; Lamersdorf: "Supporting Electronic Commerce Transactions with Contracting

- Services", International Journal on Cooperative Information Systems, 7 (4), 1998.
- Merz, M.: "Electronic Commerce. Marktmodelle, Anwendungen und Technologien", dpunkt.verlag, Heidelberg, 1999.
- Merz, M.: "E- Commerce und E- Business. Marktmodelle, Anwendungen und Technologien", dpunkt.verlag, Heidelberg, 2002.
- Merz, M.; Tu, T.; Lamersdorf, W.: "Electronic Commerce: Technologische und organisatorische Grundlagen", Informatik Spektrum, 22, 1999.
- Milgrom, P. R.: "Auctions and Bidding: A Primer", Journal of Economic Perspectives, 3 (3), 3-22, 1989.
- Milgrom, P. R.: "Game Theory and Spectrum Auctions", European Economic Review, 42, 771-778, 1998.
- Milgrom, P. R.; Weber, R. J.: "A Theory of Auctions and Competitive Bidding", Econometrica, 50 (5), 1089-1122, 1982.
- Milojicic, D.; Breugst, M.; al., e.: "Mobility - Processes, Computers and Agents", in: Milojicic, D., Douglis, F. und Wheeler, R. eds, MASIF: The OMG Mobile Agent System Interoperability Facility, ACM Press, New York, 1999.
- Moczar, L.; Aston, J.: "Cocoon Developers Handbook", Sams Publishing, Indianapolis, 2003.
- Neef, D.: "e-Procurement - From Strategy to Implementation", Prentice Hall, New York, 2001.
- Nenninger, M.: "Electronic Procurement - Neue Beschaffungsstrategien durch Desktop Purchasing Systeme", KPMG, Berlin, Technical Report, 1999.
- Newell, A.: "The knowledge level", Artificial Intelligence, (18), 1982.
- Noriega, P.: "Agent Mediated Auctions: The Fishmarket Metaphor", Facultat de Ciencies, Universitat Autonoma de Barcelona, Barcelona, Spain, Doctoral Thesis (Dissertation), 1997.
- North-Holland: "Journal on Decision Support Systems", North Holland, 1985.
- Noy, N. F.; McGuinness, D. L.: "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory, Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
- OASIS: "OASIS Universal Business Language TC", World Wide Web, <http://www.oasis-open.org/committees/ubl/>, Zugriff am 25.06.03, 2003.
- Offermann, S.: "Entwurf und Realisierung einer generischen Portalplattform und prototypischer Komponenten für das PublicationPORTAL", Fachbereich Informatik, VSIS, Hamburg, Hamburg, Studienarbeit, 2001.
- Omelayenko, B.; Fensel, D.: "Scalable Document Integration For B2B Electronic Commerce", Working Paper, 2001.
- OMG: "Mobile Agent System Interoperability Facilities Specification", Object Management Group (OMG), <http://www.omg.org/cgi-bin/doc?orbo/97-10-05>, Zugriff am 25. September 2002, 1997.

- OMG: "Agent Technology Green Paper", Object Management Group, Green Paper, [http://www.objs.com/agent/agents Green Paper v100.doc](http://www.objs.com/agent/agents%20Green%20Paper%20v100.doc), 2000.
- openTRANS: "openTRANS", <http://www.opentrans.org>, Zugriff am 25.06.2003, 2003.
- Oracle: "Oracle Technology Network (Oracle Technet)", Internet / WWW, <http://otn.oracle.com>, Zugriff am 10. Oktober 2002, 2002.
- Ouksel, A. M.; Sheth, A. P.: "Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section", SIGMOD Record, 28 (1), 5-12, 1999.
- Patil, R. S.; Fikes, R. E.; al., e.: "The DARPA Knowledge Sharing Effort: Progress Report. Principles of Knowledge Representation and Reasoning", in, 1992.
- Poslad, S.; Charlton, P.: "Standardizing Agent Interoperability: The FIPA Approach", in: ACAI 2001, Springer Verlag, Berlin, 98-117, 2001.
- Quix, C.; Schoop, M.; Jeusfeld, M.: "Business Data Management for Business-to-Business Electronic Commerce", SIGMOD Record, 31 (1), 49-54, 2002.
- Rahm, E.; Bernstein, P. A.: "A survey of approaches to automatic schema matching", The VLDB Journal, 10, 334-350, 2001.
- Raiffa, H.: "The art and science of negotiation", Harvard University Press, Cambridge, MA, 1982.
- Rebstock, M.: "Elektronische Unterstützung und Automatisierung von Verhandlungen", Wirtschaftsinformatik, 43 (6), 609-617, 2001.
- Reck, M.: "Trading-process characteristics of electronic auctions", Electronic Markets, 7 (4), 17-23, 1997.
- Russel, B.: "On denoting. The logic of grammar." in: Davidson, D. und Harman, G. eds, Dickenson Publishing Company, 184-193, 1995.
- Schilling, D.: "Der B2B-Standard ebXML: Spezifikationsanalyse und praktische Evaluation anhand einer Prototypentwicklung", Fachbereich Informatik, Universität Hamburg, Hamburg, Studienarbeit, 2002.
- Schmelzer, R.; Vandersypen, T.; Bloomberg, J.; Siddalingaiah, M.; Hunting, S.; Qualls, M.; Darby, C.; Houlding, D.; Kennedy, D.: "XML and Web Services UNLEASHED", Sams Publishing, Indianapolis, 2002.
- Schmid, B.; al., e.: "Electronic Mall: Banking und Shopping in globalen Netzen", Teubner, Stuttgart, 1995.
- Schneider, D.; Schnetkamp, G.: "E-Markets", Gabler, Wiesbaden, 2000.
- Schoop, M.: "Cooperative document management", Technische Hochschule Aachen, Fachgruppe Informatik, Aachen, Aachener Informatik-Berichte, 50, <http://aib.informatik.rwth-aachen.de/2000/00-05.ps.gz>, 2000.
- Schoop, M.; Jertila, A.: "An Electronic Business-to-Business Negotiation System", in: Ninth Research Symposium on Emerging Elec-

- tronic Markets, Basel, Switzerland, University of Applied Sciences Basel (FHBB), Institute for Business Economics, 89-94, 2002.
- Schoop, M.; Quix, C.: "DOC.COM: a framework for effective negaotiation support in electronic marketplaces", *Comupter Networks*, 37 (Special Issue on Electronic Business Systems), 153-170, 2001.
- Schubert, P.: "Procurement im E-Business : Einkaufs- und Verkaufsprozesse elektronisch optimieren ; Begriffe - Konzepte - Fallstudien", Hanser, München u.a., 2002.
- Searle, J. R.: "Speech Acts: An Essay in the Philosophy of Language", Cambridge University Press, Cambridge, 1969.
- Searle, J. R.: "Sprechakte: Ein sprachphilosophischer Essay", Frankfurt am Main, 1971.
- Searle, J. R.: "Expression and Meaning - Studies in the Theory of Speech Acts", Cambridge University Press, Cambridge, 1979.
- Sebenius, J. K.: "Negotiation Analysis: A Characterization and Review", *Management Science*, 38 (1), 18-38, 1992.
- Shim, J.; Hsiao, N.: "A Literature Review on Web-Based Negotiation Support System", WNTS, http://cpol.albany.edu/wnts/WNSS_Literature_Reivew.pdf, Zugriff am 10. Oktober 2001, 1999.
- Sieber, P.: "Virtuelle Unternehmen in der IT-Branche", Paul Haupt, Bern, 1998.
- Skiera, B., Lambrecht, A.: "Erlösmodelle im Internet", in: Herrmann, A., Albers, S. ed, *Handbuch Produktmanagement*, Gabler, Wiesbaden, 2000.
- Smith, D. C.; Cypher, A.; al., e.: "KidSim: Programming Agents Without a Programming Language", *Communications of the ACM*, 37 (7), 55-67, 1994.
- Smith, M.; Bailey, J.; Brynjolfsson, E.: "Understanding Digital Markets: Review and Assessment", <http://e-commerce.mit.edu/>, Zugriff am 15 April 2002, 1999.
- Smith, R. G.: "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Trans. Comput.*, 29 (12), 1104-1113, 1980.
- Snell, J.; Tidwell, D.; Kulchenko, P.: "Webservice-Programmierung mit SOAP", O'Reilly, Beijing, Cambridge, Farnham, Köln, Paris, Sebastopol, Taipei, Tokyo, 2002.
- Sokol, P. K.: "From Edi to Electronic Commerce: A Business Initiative", McGraw Hill, 1994.
- Steen; Lankhorst; Wetering, v. d.: "Modelling Networked Enterprises", in: EDOC2002, Lausanne, Switzerland, IEEE Computer Society, 2002.
- Steen, M.; Fielt, E.; Janssen, W.; Luttighuis, P. O.; Smit, A.: "Modelling Networked Enterprises: concepts, metamodel and notation", <https://extranet.telin.nl/docuserver/dscgi/ds.py/Get/File-9152>, Zugriff am, 2000.
- Ströbel, M.: "Engineering Electronic Negotiations: A Guide to Electronic Negotiation Technologies for the Design and Implementation of

- Next-Generation Electronic Markets - Future Silkroads of eCommerce", Kluwer Academic / Plenum Publishers, New York, Boston, Dordrecht, London, Moscow, 2003.
- Stultz, R. A.: "Demystifying EDI, A Practical Guide to Electronic Data Interchange Implementation, Transactions, and Systems", Wordware Publishing, Inc., Plano, Texas, USA, 2001.
- Teich, J.: "NegotiAuction: A Hybrid, Web-based Auction Procedure Combining Aspects of Negotiation and Auctions", in: 31st Annual Southwest Decision Sciences Institute Conference, San Antonio, TX, USA, 2000.
- Thelen, T.: "Spieltheorie und das Gefangenendilemma", Internet, <http://tobiasthelen.de/ipd/gesamt.html>, Zugriff am 20. Januar 2003, 2000.
- Thome, R.: "e-Business", Informatik-Spektrum, 25 (2), 151-153, 2002.
- Thun, P.; Rebstock, M.: "Generische Konzeption eines Verhandlungsprozesses in ebXML", in: Net.ObjectDays 2002, Ilmenau, 205-218, 2002.
- Thun, P.; Rebstock, M.: "Developing a Generic Concept of an Electronic Negotiation Process with ebXML", in: Aksit, M., Mezini, M. und Unland, R. eds, Objects, Components, Architectures, Services, and Applications for a Networked World, Springer, Berlin/ Heidelberg/ New York etc., 2003.
- Timmers, P.: "Business Models for Electronic Markets", Electronic Markets, 8 (2), 3-8, 1998.
- Timmers, P.: "Electronic Commerce: Strategies and Models for Business-to-Business Trading", Wiley, Chichester, 2000.
- Traum, D. R.: "20 Questions for Dialogue Act Taxonomies", Journal of Semantics, 17 (1), 2000.
- Tu, M. T. V.: "Steuerung offener verteilter Anwendungssysteme: Konstruktion interaktionsorientierter Regelverarbeitungs- und Verhandlungsmechanismen; dargestellt am Beispiel elektronischer Dienstmärkte", Fachbereich Informatik, Universität Hamburg, Hamburg, Doktorarbeit, 379, 2000.
- Uschold, M.; Gruninger, M.: "Ontologies: principles, methods, and applications", Knowledge Engineering Review, 11 (2), 93-155, 1996.
- Uschold, M.; King, M.: "Towards a methodology for building ontologies", in, 1995.
- VW: "VW Group supply.com - Die Lieferantenplattform des Volkswagen Konzerns", World Wide Web, <http://www.vwgroupsupply.com>, Zugriff am 25.06.03, 2003.
- Waldau, B.: "Erweiterung von Web-Services um eine Verhandlungsfunktionalität", Fachbereich Informatik, Universität Hamburg, Hamburg, Studienarbeit, 2002.
- Weigand, H.; Schoop, M.; Moor, A. d.; Dignum, F.: "B2B Negotiation Support: the need for a communication perspective", Group Decision and Negotiation, 12 (1), 3-29, 2003.

- Weiss, G. (ed.), "Multiagent systems : a modern approach to distributed artificial intelligence", MIT Press, Cambridge, Mass. [u.a.], 1999.
- Weitzel, T.; Harder, T.; Buxmann, P.: "Electronic Business und EDI mit XML", dpunkt.verlag, Heidelberg, 2001.
- Widhani, A.: "Software-Architekturen von Online-Shopssystemen", Fachbereich Informatik, VSIS, Universität Hamburg, Hamburg, Studienarbeit, 2002.
- Widhani, A.; Böge, S.; Bartelt, A.; Lamersdorf, W.: "Software Architecture and Patterns for Electronic Commerce Systems", in: Ninth Research Symposium on Emerging Electronic Markets (RSEEM'02), 2002.
- Willmott, S.; Calisti, M.; Boi, F.; Santiago, M.-G.; Belakhdar, O.; Torrens, M.: "CCL: Expressions of Choice in Agent Communication", in: 4th International Conference on Multi-Agent Systems (ICMAS), 2000.
- Witten, I. H.; Bainbridge, D.: "How to Build a Digital Library", Morgan Kaufmann Publishers, Amsterdam u.a., 2002.
- Wooldridge, M.: "Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence." in: Weiss, G. ed, MIT Press, Cambridge, 1999.
- Wooldridge, M.: "An Introduction to MultiAgent Systems", John Wiley and Sons Ltd., Chichester, 2002.
- Wooldridge, M.; Jennings, N. R.: "Intelligent agents: Theory and practice", Knowledge Engineering Review, 1995.
- Wriggers, A.: "Konzepte des E-Business Modellings und beispielhafte Anwendung im Bereich der E-Cooperation", Fachbereich Informatik, VSIS, Universität Hamburg, Hamburg, Diplomarbeit, 2003.
- xCBL: "XML Common Business Library (xCBL)", World Wide Web, <http://www.xcbl.org>, Zugriff am 25.06.03, 2003.
- Xie, Z.; Liu, K.: "Improving Business Modelling with Organisational Semiotics", University of Reading, Reading, UK, 2001.
- Yen, B. P.-C.; Ng, E. O. S.: "Migrating Procurement onto the Internet", Electronic Commerce Research, 2, 113–134, 2002.
- Yuan, Y., Suraga; Rose, J. B.; Archer, N.: "A Web-Based Negotiation Support System", Electronic Markets, 8 (3), 13-17, 1998.
- Zirpins, C.; Weinreich, H.; Bartelt, A.; Lamersdorf, W.: "Advanced Concepts for Next Generation Portals", in: 12th International Workshop on Database and Expert Systems Applications, München, IEEE Computer Society, Los Alamos, California, 501-506, 2001.
- Zwißler, S.: "Electronic Commerce Electronic Business: Strategische und operative Einordnung, Techniken und Entscheidungshilfen", Springer, Berlin, Heidelberg, New York u.a., 2002.