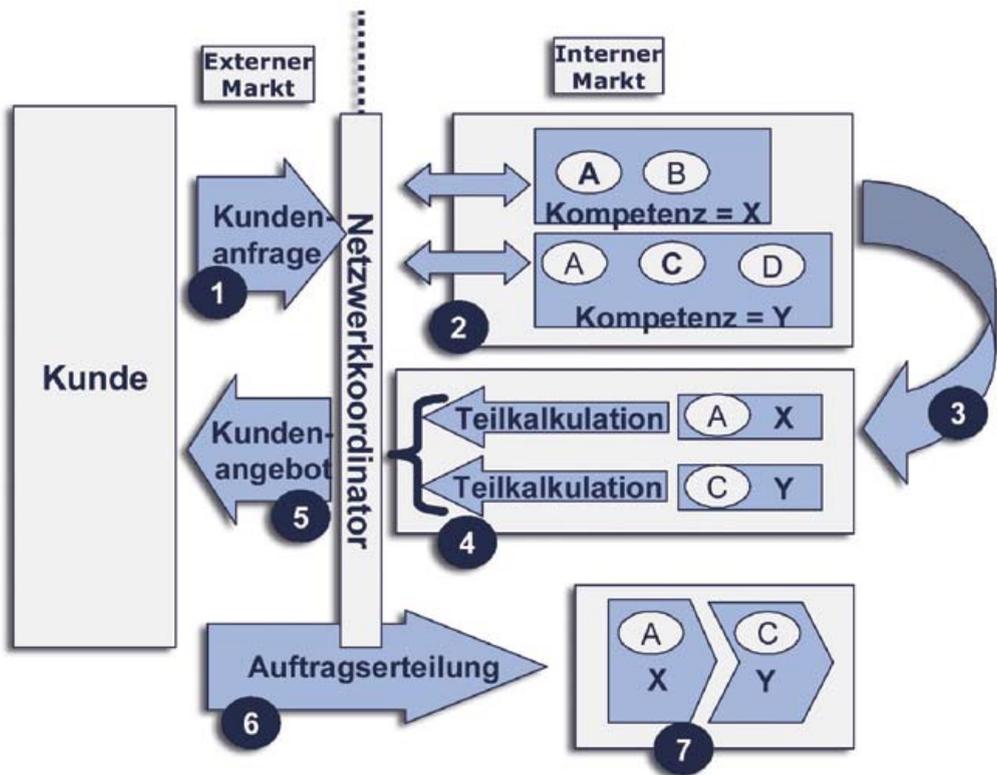


Matthias Schumann/Thomas Hess/
Stefan Wittenberg/Markus Burghardt/Thomas Wilde

Management von virtuellen Unternehmen

Softwareunterstützung und innovative Technologien



Matthias Schumann/Thomas Hess/Stefan Wittenberg/
Markus Burghardt/Thomas Wilde
Management von virtuellen Unternehmen

erschienen in der Reihe der
Universitätsdrucke Göttingen 2004

Matthias Schumann
Thomas Hess
Stefan Wittenberg
Markus Burghardt
Thomas Wilde

Management von virtuellen Unternehmen

Softwareunterstützung und
innovative Technologien



Universitätsverlag Göttingen
2004

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Georg-August-Universität Göttingen
Wirtschaftswissenschaftliche Fakultät
Institut für Wirtschaftsinformatik II
Prof. Dr. Matthias Schumann

Ludwig-Maximilians-Universität München
Department für Betriebswirtschaft
Institut für Wirtschaftsinformatik und Neue Medien
Prof. Dr. Thomas Hess

Die vorliegende Publikation entstand im Rahmen des durch das BMBF geförderten Vorhabens "Werkzeuggestütztes Management wissensintensiver Dienstleistungen in virtuellen Unternehmen"
FKZ: 01HW0141, Projektlaufzeit: 01.06.2001-30.11.2003

© Alle Rechte vorbehalten, Universitätsverlag Göttingen 2004
Umschlaggestaltung: Margo Bargheer

ISBN 3-930457-68-7

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	IX
1 Einleitung	1
2 Virtuelle Unternehmen	5
2.1 <i>Virtuelle Unternehmen als Netzwerkorganisation</i>	5
2.2 <i>Bedeutung der IT für virtuelle Unternehmen</i>	7
3 Werkzeugunterstützung für das Management virtueller Unternehmen	8
3.1 <i>Aufgaben, Instrumente und Werkzeuge</i>	8
3.2 <i>Basisanforderungen an IT in virtuellen Unternehmen</i>	10
3.3 <i>Funktionale Anforderungen für die Werkzeugunterstützung</i>	10
3.3.1 <i>Informationsphase</i>	11
3.3.2 <i>Vereinbarungsphase</i>	12
3.3.3 <i>Durchführungsphase</i>	13
3.3.4 <i>Auflösungsphase</i>	14
3.4 <i>Werkzeuge für die Unterstützung wichtiger Managementinstrumente</i>	14

3.4.1 Informationsphase	15
3.4.2 Vereinbarungsphase.....	15
3.4.3 Durchführungsphase	16
3.4.4 Auflösungsphase	17
4 Softwareunterstützung durch VICOPLAN	19
4.1 Hintergrund des Entwicklungsprojektes	19
4.2 Auswahl eines Vorgehensmodells für die Entwicklung.....	20
4.3 Anforderungsanalyse.....	24
4.3.1 Organisationssicht.....	25
4.3.2 Funktionssicht	26
4.3.3 Datensicht.....	27
4.4 Softwarekonzeption.....	28
4.4.1 Fachliche Spezifikation.....	28
4.4.2 Technische Spezifikation - Softwarearchitektur	38
4.5 Softwarerealisierung	39
4.5.1 Implementierung	39
4.5.2 Technische Basis	40
4.5.3 Evaluation.....	45
4.6 Funktionsumfang von VICOPLAN.....	48
4.6.1 Konfiguration des Systems.....	48
4.6.2 Das Kundeninterface.....	48
4.6.3 Koordinatoren-Seite I - Ausschreibung	49
4.6.4 Partnerseite.....	51
4.6.5 Koordinatoren-Seite II - Erstellen des Kundenangebots	52
4.6.6 Kundenseite	53
4.6.7 Koordinatoren-Seite III - Auftragsplanung.....	54
5 Analyse ausgewählter innovativer Technologien	57
5.1 Extensible Markup Language (XML)	58
5.1.1 Grundlagen.....	59
5.1.2 Einsatzmöglichkeiten in virtuellen Unternehmen	67
5.1.3 Beurteilung	70
5.2 Web Services.....	72

5.2.1 Grundlagen.....	72
5.2.2 Einsatzmöglichkeiten in virtuellen Unternehmen	79
5.2.3 Beurteilung	82
5.3 <i>Online Analytical Processing (OLAP)</i>	82
5.3.1 Grundlagen.....	83
5.3.2 Einsatzmöglichkeiten in virtuellen Unternehmen	88
5.3.3 Beurteilung	94
6 Zusammenfassung und Ausblick.....	97
Literaturverzeichnis	99
Anhang.....	107
<i>Pflichtenheft von VICOPLAN in der Kurzfassung</i>	107
<i>Ausgewählte relevante Veröffentlichungen, durchgeführte Workshops, Messeteilnahmen sowie aktuelle Informationen</i>	115

Abbildungsverzeichnis

Abbildung 1:	Aufbau eines virtuellen Unternehmens	1
Abbildung 2:	Grundtypen von Unternehmensnetzwerken	6
Abbildung 3:	Systematisierung von Aufgaben, Instrumenten, Werkzeugen und Technologien	9
Abbildung 4:	Matching im Netzwerk	11
Abbildung 5:	Tailoring	20
Abbildung 6:	Auswahl Vorgehensmodell	22
Abbildung 7:	Baseballmodell	23
Abbildung 8:	Organisationssicht	25
Abbildung 9:	Funktionssicht Auftragslebenszyklus	27
Abbildung 10:	Ereignisgesteuerte Prozesskette als Soll-Prozess	30
Abbildung 11:	Anwendungsfalldiagramm	32
Abbildung 12:	Zusammenhang von OOA, OOD und OOP	33
Abbildung 13:	Sichten und Gliederung der OOA	34
Abbildung 14:	Zentrales Klassenmodell	36
Abbildung 15:	Klassenmodell Dokumentation	37
Abbildung 16:	E-Mail-Benachrichtigung	37
Abbildung 17:	Systemarchitektur	39
Abbildung 18:	Entwurf der Datenbankkomponente	41
Abbildung 19:	Klassendiagramm des Graphical User Interface (GUI)	42
Abbildung 20:	Entwurf GUI – Dynamik (Beispiel)	43
Abbildung 21:	Beispiel eines XML-Dokuments	61
Abbildung 22:	Beispiel einer Document Type Definition (DTD) .	62
Abbildung 23:	Beispiel eines XML Schema	64
Abbildung 24:	Fachliche Spezifikation des Datenaustauschs bei Fortschrittsabfrage	68

Abbildung 25:	Beispiele für Strukturdefinitionen für die identifizierten Nachrichtentypen	69
Abbildung 26:	Ablauf einer Datenübertragung	70
Abbildung 27:	Architekturmodell für Web Services	74
Abbildung 28:	Lebenszyklus eines Web Services	77
Abbildung 29:	Zwiebelschalenmodell zur Anordnung der Basistechnologien bei Web Services	78
Abbildung 30:	Kommunikation der Partnerunternehmen über Web Service Schnittstellen.....	80
Abbildung 31:	Gekapselte zentrale Anwendungslogik auf Basis von Web Services	81
Abbildung 32:	Architekturvarianten von OLAP-Systemen.....	87
Abbildung 33:	Controllingaufgaben in virtuellen Unternehmen	90
Abbildung 34:	OLAP-Architektur für das Controlling in virtuellen Unternehmen	91
Abbildung 35:	Beispiel einer multidimensionalen Sichtweise auf die Netzwerkerlöse	93

Tabellenverzeichnis

Tabelle 1:	Gegenüberstellung von Aufgaben und Instrumenten in der Informationsphase	12
Tabelle 2:	Gegenüberstellung von Aufgaben und Instrumenten in der Vereinbarungsphase	13
Tabelle 3:	Gegenüberstellung von Aufgaben und Instrumenten in der Durchführungsphase	14
Tabelle 4:	Gegenüberstellung von Aufgaben und Instrumenten in der Auslösungsphase.....	14
Tabelle 5:	Notwendige Systemfunktionen von VICOPLAN .	32
Tabelle 6:	Langfristig zu speichernde Daten	45
Tabelle 7:	Relationenschema.....	45
Tabelle 8:	Umsetzung der Musskriterien-Funktionalität.....	46

Abkürzungsverzeichnis

d. h.	das heißt
DOM	Document Object Model
DSSSL	Document Style Semantics and Specification Language
DTD	Document Type Definition
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IT	Informationstechnologie
IuK	Information und Kommunikation
OOA	Objektorientierte Analyse
OOD	Objektorientiertes Design
OOP	Objektorientierte Programmierung
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
SAX	Simple API for XML
SGML	Standard Generalized Markup Language
u. a.	unter anderem
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSL-FO	XSL Formatting Objects
XSLT	XSL Transformations

1 Einleitung

Wissensintensive Dienstleistungen werden verstärkt in Kooperationen erbracht. In den letzten Jahren haben dabei die virtuellen Unternehmen als spezielle Form der zwischenbetrieblichen Kooperation besonders an Bedeutung gewonnen. In virtuellen Unternehmen arbeiten eine größere Zahl von Unternehmen auf partnerschaftlicher Basis ohne Aufgabe der rechtlichen Selbständigkeit langfristig zusammen.¹ Aus diesem Pool von Unternehmen werden auftragsbezogene Teams gebildet, die am Markt gemeinsame Projekte abwickeln oder Leistungen anbieten. Abbildung 1 zeigt ein einfaches Beispiel.

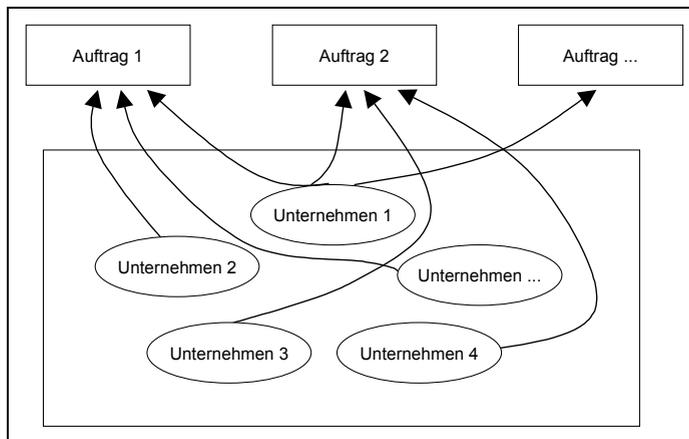


Abbildung 1: Aufbau eines virtuellen Unternehmens²

¹ Vgl. Hess (2002a), S. 14-16.

² Vgl. Hess (2002b), S. 24-33.

Ebenfalls charakteristisch für virtuelle Unternehmen ist der massive Einsatz von Informations- und Kommunikationssystemen (IuK-Systemen) zur Unterstützung der internen Kommunikation und Koordination.

Ziel dieser Veröffentlichung ist es, auf der einen Seite sowohl die fachlichen Anforderungen an ein Werkzeug niederzulegen als auch basierend auf diesen Anforderungen ein umfassenden Softwareprototypen zur Unterstützung des Managements dienstleistungsorientierter virtueller Unternehmen zu beschreiben. Der Prototyp entstand im Rahmen eines durch das Bundesministerium für Bildung und Forschung (BMBF) geförderten Vorhabens mit dem Titel „Werkzeuggestütztes Management wissensintensiver Dienstleistungen in virtuellen Unternehmen“ (Förderkennzeichen: 01HW0141) in Zusammenarbeit der Universitäten Göttingen (Institut für Wirtschaftsinformatik Abteilung Wirtschaftsinformatik II, Prof. Dr. Matthias Schumann) und München (Institut für Wirtschaftsinformatik und Neue Medien, Prof. Dr. Thomas Hess). Neben der Entwicklung eines lauffähigen Prototyps hinweg war aus technischer Sicht ferner zu analysieren, ob virtuelle Unternehmen durch die Nutzung von innovativen Technologien wie XML, Web Services und OLAP-Systemen, sinnvoll unterstützt werden können. Dabei war vor allem die Kopplung bzw. Integration bestehender IT-Systeme von großer Relevanz.

Wie bereits im vorigen Abschnitt skizziert, war die Entwicklung eines prototypischen Softwaretools zur Unterstützung des Managements virtueller Unternehmen neben der Untersuchung relevanter innovativer Technologien die im Projekt verfolgte Zielsetzung. Um dieses Ziel zu erreichen, wurden zuerst auf Basis eines Literaturstudiums die managementrelevanten Funktionalitäten innerhalb virtueller Unternehmen identifiziert. Diese theoretisch gewonnenen Erkenntnisse wurden gemeinsam mit dem Praxispartner Techno-Pool, einem virtuellen Unternehmen aus Groß-Gerau, verfeinert und um spezielle Anforderungen des Praxispartners angereichert. Das Ergebnis dieses Prozesses war ein Pflichtenheft für ein idealtypisches Softwaretool, welches dann unter Anwendung bewährter Methoden des Software-Engineering prototypisch implementiert wurde. Das Ergebnis dieses Entwicklungsprozesses ist der Softwareprototyp VICOPLAN (Virtual Corporation Planning System). Die Kurzfassung des erarbeiteten Pflichtenhefts ist im Anhang dieser Veröffentlichung zu finden.

Aus dem hier beschriebenen methodischen Vorgehen ergibt sich unmittelbar der Aufbau dieser Veröffentlichung:

Im Kapitel zwei werden zunächst der Begriff des „Virtuellen Unternehmens“ erläutert und Netzwerke als eine besondere Form einer Kooperation eingeordnet. Weiterhin wird auf die Bedeutung der IT für virtuelle Unternehmen eingegangen.

Im dritten Kapitel werden die grundlegenden Anforderungen an Softwaretools für das Management virtueller Unternehmen untersucht und bis auf die Ebene konkret zu unterstützender Managementinstrumente dargestellt. Weiterhin werden

im Kapitel drei bestehende Werkzeuge diskutiert, welche die geforderten Anforderungen bereits zumindest teilweise erfüllen.

Das Kapitel vier bildet mit der Darstellung der Prototypentwicklung den Kern dieser Veröffentlichung. Hierzu werden zunächst die Rahmenbedingungen und der Hintergrund des durch das Bundesministerium für Bildung und Forschung geförderten Projektes erläutert. Anschließend wird der Entscheidungsprozess für die Auswahl des gewählten Vorgehensmodells zur Softwareentwicklung innerhalb des Projektes skizziert. Aufbauend auf diesen Grundlagen werden anschließend die fachliche und technische Umsetzung des Prototypen diskutiert.

Das fünfte Kapitel fasst die Ergebnisse der Untersuchung ausgewählter innovativer Technologien hinsichtlich der Einsatzmöglichkeiten in virtuellen Unternehmen zusammen. Als innovative Ansätze werden in diesem Kapitel die Metasprache XML, Web Services sowie OLAP-Technologien untersucht.

Im sechsten Kapitel wird eine kurze Zusammenfassung der Ausführungen vorgenommen sowie ein Ausblick auf die Übertragbarkeit der Projektergebnisse gegeben. Auch werden Anknüpfungspunkte für die Weiterentwicklung des Prototyps aufgezeigt.

Im Anhang sind dann auf der einen Seite eine Kurzfassung des Pflichtenhefts von VICOPLAN zu finden, auf der anderen Seite werden relevante Veröffentlichungen, durchgeführte Workshops, Messeteilnahmen der Autoren sowie aktuelle Informationsquellen im Umfeld von VICOPLAN überblickartig aufgeführt.

2 Virtuelle Unternehmen

Nachfolgend wird zunächst der Begriff des „Virtuellen Unternehmens“ eingeführt, da für diesen Unternehmenstyp ein Softwarewerkzeug für die Unterstützung managementrelevanter Funktionalitäten sowohl fachlich konzipiert als auch prototypisch implementiert wurde. Um die Vollständigkeit zu wahren, wird der Begriff des „virtuellen Unternehmens“ als eine besondere Netzwerkform in das Kooperationsumfeld eingeordnet. Anschließend wird auf die besondere Bedeutung der Informationstechnologie für virtuelle Unternehmen eingegangen.

2.1 Virtuelle Unternehmen als Netzwerkorganisation

Unternehmensnetzwerke stellen neben Joint Ventures und strategischen Allianzen den dritten Grundtyp einer zwischenbetrieblichen Kooperation dar. Im Rahmen einer derartigen Kooperation arbeiten in einem Netzwerk mindestens drei, oftmals aber zehn oder mehr, rechtlich selbständige Unternehmen zum Zwecke der gemeinsamen Leistungserstellung auf langfristiger Basis zusammen.³ Die rechtlich selbständigen Unternehmen verfolgen mit der Zusammenarbeit im Netzwerk ein kollektives Ziel, das in der Regel auf die optimale Gestaltung der gemeinschaftlichen Wertschöpfung und Verteilung der Kooperationserlöse abzielt.⁴ Unternehmensnetzwerke lassen sich nach der Stabilität der Konfiguration und der Steuerungsform unterscheiden.⁵

In stabilen Netzwerken werden mehrere Aufträge in gleicher Konfiguration ausgeführt, während sich in instabilen Netzwerken die Zusammensetzung der ausführenden Partnerunternehmen abhängig vom Auftrag ändert.

Hinsichtlich der Steuerungsform können fokale Netzwerke von polyzentrischen Netzwerken abgrenzt werden. Bei fokalen Netzwerken liegt die Entscheidungsgewalt in den Händen eines oder einer Gruppe von Partnerunternehmen. Polyzentrische Netzwerke zeichnen sich demgegenüber durch eine

³ Vgl. Wohlgemuth (2002), S. 17-18.

⁴ Vgl. Siebert (2001), S. 8-27.

⁵ Vgl. Hess (2002a), S. 16.

verteilte, gleichberechtigte Machtstruktur aus. Wie in der Abbildung 2 dargestellt, lassen sich anhand der Merkmale Stabilität und Steuerungsform mit Projektnetzwerken, strategischen Netzwerken, virtuellen Unternehmen und Verbundnetzwerken vier Grundtypen von Unternehmensnetzwerken unterscheiden.⁶

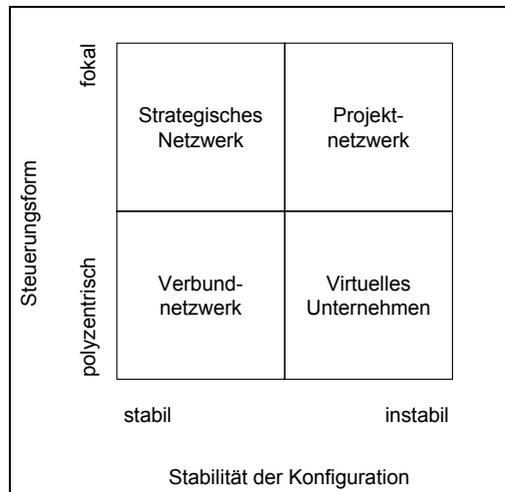


Abbildung 2: Grundtypen von Unternehmensnetzwerken⁷

Die älteste Ausprägung sind Projektnetzwerke, bei denen ein fokaler Partner projektspezifisch ein Auftrags-team zusammenstellt. Projektnetzwerke findet man häufig in der Baubranche und der Filmproduktion.⁸ Strategische Netzwerke, die sich durch eine stabile Konfiguration und eine fokale Steuerung auszeichnen, sind oftmals im Automobilssektor anzutreffen.⁹ Typischerweise organisiert dort ein als fokaler Partner auftretender Hersteller seine Zulieferbetriebe in Form eines Netzwerks. Als letzter Grundtyp von Netzwerken, hier mit einer polyzentrischen Steuerungsform und stabiler Konfiguration, sind Verbundnetzwerke zu nennen, die bisher in der Literatur kaum behandelt wurden und in der betrieblichen Praxis vorwiegend im Bereich von Verkehrsverbänden existieren.¹⁰ Als bekanntestes Beispiel ist in diesem Zusammenhang das Luftverkehrsnetzwerk „Star Alliance“ zu nennen, in dessen Rahmen sich zurzeit 14 Partnerunternehmen zu einem Verbundnetzwerk zusammengeschlossen haben.¹¹

Die in dieser Veröffentlichung thematisierten virtuellen Unternehmen stellen eine seit etwa 10 Jahren vielfach diskutierte Form der Netzwerke dar. In virtuellen

⁶ Vgl. Sydow (1999), S. 279-314.

⁷ Hess (2002a), S. 16.

⁸ Vgl. Windeler/Lutz/Wirth (2000), S. 178-205.

⁹ Vgl. Köhler (2000), S. 281-300; Semlinger (2001), S. 30-74.

¹⁰ Vgl. Wohlgemuth (2002), S. 23.

¹¹ Im Luftverkehr werden Netzwerke zumeist im Sinne einer strategischen Allianz diskutiert, vgl. z.B. Netzer (1999) oder Oum/Park/Zhang (2001).

Unternehmen besteht eine polyzentrische Machtstruktur im Rahmen derer sich auftragsabhängig verschiedene Konfigurationen zur Auftragsdurchführung bilden. Nach obiger Einordnung sind virtuelle Unternehmen somit instabil.

2.2 Bedeutung der IT für virtuelle Unternehmen

Die Entstehung und der Erfolg von Unternehmensnetzwerken im Allgemeinen und virtuellen Unternehmen im Speziellen werden oftmals mit den neuen Potenzialen der Informationstechnologie (IT) begründet. Die IT ermöglicht danach im Sinne eines Enablers die unternehmensübergreifende Koordination der Wertschöpfung, die ohne IT aus technischer und ökonomischer Sicht nicht sinnvoll wäre.¹²

Die Rahmenbedingungen für den IT-Einsatz sind jedoch grundlegend anders als in herkömmlichen Unternehmen: So erfolgt die Leistungserstellung in virtuellen Unternehmen räumlich und zeitlich verteilt, wodurch die Schnittstellen zwischen den Partnerunternehmen an Bedeutung gewinnen und zusätzlicher Koordinationsaufwand entsteht. Wie oben beschrieben bleiben die Partnerunternehmen des virtuellen Unternehmens rechtlich und wirtschaftlich unabhängig, so dass sich die IT im virtuellen Unternehmen zwingend an der bereits vorhandenen IT-Infrastruktur der Partnerunternehmen ausrichten muss. Die bestehende IT-Infrastruktur bleibt im Regelfall für Aufträge, die nicht durch das gebildete virtuelle Unternehmen durchgeführt und abgewickelt werden, im Einsatz.

¹² Vgl. Zerbe/Krcmar (1999), 189ff.; Picot/Reichwald/Wiegand (2001), S. 437; Hedberg/Dahlgren/Hansson/Nils-Göran (1997), S. 20.

3 Werkzeugunterstützung für das Management virtueller Unternehmen

In diesem Kapitel werden die grundlegenden Anforderungen an Softwaretools für das Management virtueller Unternehmen dargestellt. Dabei werden zuerst die an dieser Stelle benutzten Begriffe „Aufgabe“, „Instrument“ und „Werkzeug“ voneinander abgegrenzt sowie ihre Beziehung untereinander erläutert sowie die Basisanforderungen an die Informationstechnologie innerhalb solcher virtueller Unternehmen skizziert. Daran anschließend werden managementrelevante Aufgaben herausgearbeitet und eine Kausalkette von der Aufgabe über die Instrumente hin zu den Werkzeugen erarbeitet. Innerhalb der betrachteten Werkzeuge können dann unter anderem die betrachteten innovativen Technologien zum Einsatz kommen.

3.1 Aufgaben, Instrumente und Werkzeuge

In der Literatur lassen sich zahlreiche Ansätze zur Systematisierung der Informationstechnologie in virtuellen Unternehmen identifizieren. Diese beschränken sich allerdings oftmals auf eine Beschreibung der potenziell einsetzbaren Basistechnologien und beziehen die betriebswirtschaftlichen Anforderungen an den Einsatz der IT nur unzureichend ein.

Um von den Anforderungen Rückschlüsse auf den Einsatz möglicher Softwarefunktionen und Technologien ziehen zu können, wird ein mehrstufiges Vorgehen gewählt, welches nachfolgend erläutert wird:

Die Anforderungen an eine IT-Unterstützung leiten sich zunächst aus den anfallenden Managementaufgaben im Netzwerk ab. Zur Bewältigung dieser Aufgaben stehen dem Management des virtuellen Unternehmens verschiedene betriebswirtschaftliche Instrumente zur Verfügung. Unter einem Instrument ist in

diesem Zusammenhang eine feste Abfolge von Schritten zur zielorientierten Problemlösung zu verstehen. Wird ein Instrument informationstechnologisch umgesetzt, spricht man von einem Werkzeug. Zur Realisierung der Werkzeuge sind wiederum bestimmte Basistechnologien erforderlich. Diese zunächst abstrakte Beschreibung soll nachfolgend an einem konkreten Beispiel verdeutlicht werden.

Das Management des virtuellen Unternehmens muss mit seinen Partnerunternehmen über Preise für die Ausführung von Teilleistungen verhandeln. Die Aufgabe, die sich hieraus ergibt ist also eine Preisverhandlung. Preisverhandlungen lassen sich über eine Auktion abwickeln, dabei stellt die Auktion das betriebswirtschaftliche Instrument dar. Es besteht auch die Möglichkeit, eine Auktion elektronisch über das Internet abzuwickeln, das hierzu erforderliche Werkzeug wäre dann eine elektronische Auktion. Um eine elektronische Auktion realisieren zu können sind bestimmte Basistechnologien, wie beispielsweise Internet-technologien oder Web Services, einsetzbar.

Die Abbildung 3 verdeutlicht den eben beschriebenen Zusammenhang.



Abbildung 3: Systematisierung von Aufgaben, Instrumenten, Werkzeugen und Technologien

Im nächsten Teilkapitel werden zunächst grundlegende Anforderungen aus der Literatur dargestellt, um anschließend anhand dieser Systematik konkrete Aufgaben und Instrumente herauszuarbeiten, die später idealtypisch von einem Werkzeug unterstützt werden sollten.

3.2 Basisanforderungen an IT in virtuellen Unternehmen

Ein Blick in die einschlägige Literatur vermittelt ein breites Spektrum an Anforderungen, die an IT in virtuellen Unternehmen gestellt wird. Upton / McAfee argumentieren aus betriebswirtschaftlicher Sicht und betonen auch „weiche“ Faktoren. So fordern sie zum Beispiel, dass durch IT in virtuellen Unternehmen alle notwendigen Funktionen abgedeckt werden müssen, wobei aber auch unterschiedliche IT-Erfahrungen der Partner berücksichtigt werden müssten.

Müller unterteilt die Anforderungen an die IT in virtuellen Unternehmen in die zur „Aufgabenerfüllung notwendigen Funktionen“ und weitere, „ganz konkrete Anforderungen an die IT-Architektur“.¹³ Seine Argumentation fokussiert sich auf die technische Sicht, wobei er auf die Notwendigkeit einer offenen Vernetzungsstruktur, einer modularen Gestaltung der Systeme und Kopplungsfähigkeit der IT hinweist. Weiterhin betont er die Potenziale der Standardisierung für den IT-Einsatz in verteilten Unternehmen. Unter Bezug auf Mertens und Faisst unterscheidet er mit dem Applikations-Sharing, Daten-Sharing und der Applikations-Kommunikation drei verschiedene Stufen der Kopplung verschiedener IT-Systeme nach abnehmender Kopplungsintensität.¹⁴

Besondere Potenziale sieht Müller in Internettechnologien, die auf offenen Standards basieren. Dazu zählen beispielsweise die Kommunikationsprotokolle TCP/IP, HTTP sowie Metasprachen wie XML und ebXML. Diese Standards können nach Müller die Anforderungen an die IT in virtuellen Unternehmen hervorragend erfüllen. Als Ziel der IT-Unterstützung soll eine schnelle, flexible und unabhängige Interaktion von Partnern verfolgt werden. Dabei können sich die in einem virtuellen Unternehmen zusammenwirkenden Partnerunternehmen an verschiedenen Standorten befinden.¹⁵

3.3 Funktionale Anforderungen für die Werkzeugunterstützung

Die Herleitung der funktionalen Anforderungen an ein Softwaretool oder Werkzeug, welches die managementrelevanten Funktionalitäten unterstützt, kann anhand eines Phasenkonzeptes erfolgen. Mertens, Griese und Ehrenberg unterscheiden zum Beispiel bei der Abwicklung eines Auftrags durch ein virtuelles Unternehmen die Phasen der Identifikation, der Anbahnung, der Vereinbarung sowie die Auflösung. Diese auftragsabhängigen Phasen werden wiederum durch phasenübergreifende Aufgaben ergänzt, welche die strategische Ausrichtung

¹³ Vgl. Müller (2002), S.8.; Upton/McAfee (1996).

¹⁴ Vgl. Mertens/Griese/Ehrenberg (1998), S. 9-10.

¹⁵ Vgl. Müller (2002), S. 11-12.

des Netzwerks unterstützen. Im Rahmen dieser Veröffentlichung wird genau diese Phasenunterteilung genutzt und anhand dieser Unterteilung die Anforderungen an ein Softwaretool erarbeitet. Zusätzlich fließen in die Betrachtung die phasenübergreifenden Aufgaben ein. Zur Vereinfachung der nachfolgenden Ausführungen wird angenommen, dass ein konkreter Kundenauftrag oder zumindest eine konkrete Kundenanfrage an das virtuelle Unternehmen vorliegt. Es wird also ein idealtypischer Prozessablauf zugrunde gelegt. Die hier diskutierten Unterstützungsmöglichkeiten sind direkt in das zu erstellende Pflichtenheft des Softwarewerkzeuges eingeflossen und sind somit im Prototypen, der in Kapitel vier näher beschrieben wird, berücksichtigt.

3.3.1 Informationsphase

In der Informationsphase sind die passenden Partner für die Übernahme von Teilleistungen zu identifizieren und zu kontaktieren. Hierzu muss der Koordinator oder Manager eines virtuellen Unternehmens zunächst ein Anforderungsprofil für die Teilleistung festlegen, aus dem hervorgeht, welche Kompetenzen, Kapazitäten und sonstigen Anforderungen für die Durchführung benötigt werden. Um die für die Abwicklung der Teilleistung geeigneten Partnerunternehmen zu identifizieren, werden anschließend im Rahmen eines so genannten Matching die Kompetenzen und Kapazitäten der Partnerunternehmen mit dem Anforderungsprofil verglichen. In der Abbildung 4 ist das Matching durch die Ellipse dargestellt, die den gesamten Pool der Partnerunternehmen auf die relevanten Partnerunternehmen einschränkt.

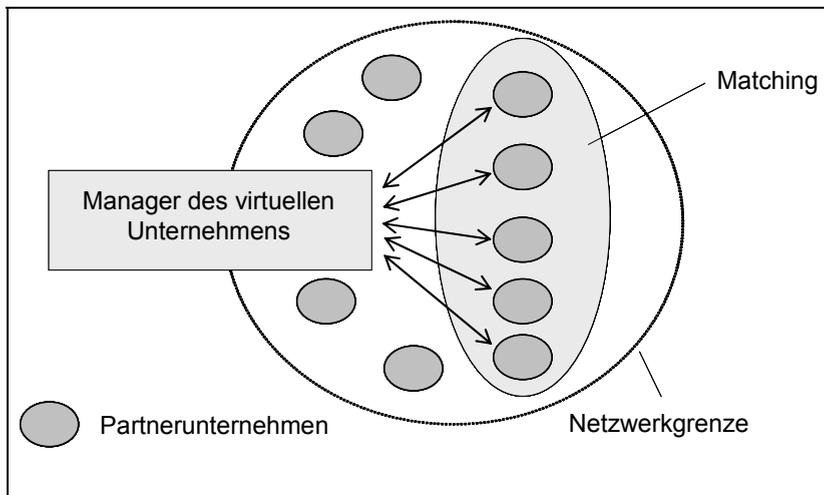


Abbildung 4: Matching im Netzwerk¹⁶

¹⁶ Vgl. Wittenberg/Hess(2002), S. 117-123.

Nach dem Matching nimmt das Netzwerkmanagement zu den ausgewählten Partnerunternehmen Kontakt auf und informiert sie über Art und Umfang der zu vergebenden Teilleistung. Aus diesen Aufgaben lassen sich die Instrumente Anforderungskatalog, Matching und Leistungskatalog ableiten, die idealerweise in einem solchen Softwaretool zu berücksichtigen sind. Tabelle 1 stellt die Aufgaben in der Informationsphase den möglichen Instrumenten gegenüber.

Aufgabe	Instrument
Definition Teilleistungen	Anforderungskatalog
Identifizieren geeigneter Partner	Matching
Information über Teilleistungen	Leistungskatalog

Tabelle 1: Gegenüberstellung von Aufgaben und Instrumenten in der Informationsphase

3.3.2 Vereinbarungsphase

Im Rahmen der Vereinbarungsphase verhandelt das Netzwerkmanagement mit den Partnerunternehmen über Preis und Leistungsumfang der Teilleistungen und fixiert nach erfolgreicher Verhandlung die Leistungskonditionen. Bei den Verhandlungen kann nach Anzahl der beteiligten Transaktionspartner zwischen freien Verhandlungen, Auktionen und Börsen unterschieden werden.

Freie Verhandlungen in einem virtuellen Unternehmen finden im Rahmen einer 1:1 Beziehung zwischen Netzwerkmanagement und einem Partnerunternehmen statt, wenn das Netzwerkmanagement direkt mit einem der Partnerunternehmen verhandelt. Freie Verhandlungen sind vor allem für schwer beschreibbare und wenig standardisierte Leistungen zu empfehlen, bei denen mehrere Iterationsschritte im Verhandlungsprozess zu durchlaufen sind.¹⁷

Bei Auktionen steht dem Netzwerkmanagement eine größere Zahl von anbietenden Partnerunternehmen gegenüber (1:n). In klassischen Auktionen konkurrieren mehrere Nachfrager um die Leistung eines Anbieters. In Netzwerken werden die beiden Rollen des Anbieters und des Nachfragers vertauscht: Das Netzwerkmanagement tritt als Nachfrager auf und spezifiziert eine Leistung, um die mehrere Partnerunternehmen konkurrieren. Diese Form der Auktion, die Ausschreibung bzw. Reverse Auction genannt wird, läuft nach dem gleichen Muster ab wie herkömmliche Auktionen.

Der dritte Typ von Transaktionen auf Märkten sind Börsen, bei denen sich auf beiden Marktseiten mehrere Anbieter und Nachfrager gegenüberstehen (m:n). Für den Fall eines Netzwerkunternehmens mit einem einzigen Nachfrager ist diese Form der Marktorganisation nicht von Bedeutung und wird in dieser Veröffentlichung nicht weiter vertieft.

¹⁷ Vgl. Hess/Wittenberg (2002).

Das Ergebnis der Vereinbarungsphase auf externen Märkten ist ein Kaufvertrag. In virtuellen Unternehmen ist in Abhängigkeit vom Vertrauensverhältnis zwischen Netzwerkmanagement und Partnerunternehmen die Ausgestaltung der Vertragsformen anzupassen. Unter Umständen wird sogar auf die explizite Formulierung eines Vertrages verzichtet und stattdessen auf die gute Reputation des Partners vertraut.¹⁸ An die Ausschreibung schließt sich allerdings im Regelfall eine Verhandlung zwischen Koordinator und Partnerunternehmen an, in der über weitere Leistungskonditionen verhandelt wird, die bei der Ausschreibung ex ante nicht fixiert wurden. Zusammenfassend lassen sich somit freie Verhandlungen, Auktionen sowie Verträge als zu unterstützende Instrumente dieser Phase festhalten. Tabelle 2 stellt diese Aufgaben der Vereinbarungsphase den möglichen Instrumenten gegenüber.

Aufgabe	Instrument
Preisverhandlungen durchführen	Verhandlung, Auktion
Fixieren von Verhandlungsergebnissen	Vertrag

Tabelle 2: Gegenüberstellung von Aufgaben und Instrumenten in der Vereinbarungsphase

3.3.3 Durchführungsphase

In dieser Phase findet die eigentliche Wertschöpfung des virtuellen Unternehmens statt, die vom Netzwerkmanagement durch ein konkret ausgestaltetes Projektmanagement und -controlling zu planen, zu steuern und zu überwachen ist. Aufgrund der räumlichen und zeitlich verteilten Leistungserstellung verbunden mit den beschränkten Sanktionsmöglichkeiten des Netzwerkmanagements ergeben sich jedoch für ein Projektmanagement in virtuellen Unternehmen besondere Herausforderungen. So ist zum Beispiel ein Partnerunternehmen für die Durchführung der Teilleistung in der Regel voll verantwortlich und übernimmt die lokale Feinplanung, während der Netzwerkkordinator lediglich im Rahmen einer Grobplanung die Aktivitäten der verschiedenen Partnerunternehmen durch Zeit- und Ressourcenplanung aufeinander abstimmt.

Um ein effektives Projektmanagement sicherzustellen, ist es Aufgabe des Netzwerkmanagements über die typischen Projektmanagementaufgaben hinaus den zusätzlich entstehenden Informationsbedarf an den Schnittstellen zwischen den verschiedenen Teilaufträgen zu ermitteln und die entsprechenden Informationen an die Partner weiterzugeben.

Im Rahmen eines Projektcontrollings ist es Aufgabe des Netzwerkmanagements, die Einhaltung der aus der Projektplanung abgeleiteten Soll-Daten zu überwachen, Abweichungsanalysen durchzuführen und über Steuerungsmaßnahmen auf die Leistungserstellung Einfluss zu nehmen. Allerdings ist das Netzwerkmanagement auf die Aktualität und formale Korrektheit der von den

¹⁸ Vgl. Wittenberg/Hess (2002).

Partnerunternehmen übermittelten Statusdaten angewiesen, die nicht immer sichergestellt ist. Weiterhin fallen die Abrechnung und der Zahlungsverkehr der Leistungen in den Aufgabenbereich des Managements des virtuellen Unternehmens, der durch standardisierte Kalkulationsmethoden zu unterstützen ist. Tabelle 3 stellt die Aufgaben den Instrumenten in der Durchführungsphase gegenüber.

Aufgabe	Instrument
Grobe Auftragsplanung	Balkendiagramme, Zeit- und Kostenpläne
Auftragsüberwachung	Projektmanagementmethoden
Rechnungs- und Zahlungsabwicklung	Kalkulationssystematik

Tabelle 3: Gegenüberstellung von Aufgaben und Instrumenten in der Durchführungsphase

3.3.4 Auflösungsphase

Auch nach der erbrachten Leistung steht das virtuelle Unternehmen dem Kunden für Servicearbeiten bzw. Nachbesserungen zur Verfügung. Dementsprechend sind die Partnerunternehmen in die Auflösungsphase involviert und werden bei Bedarf aktiv. Eine wichtige Aufgabe für das Netzwerkmanagement die und Partnerunternehmen ist in diesem Zusammenhang die Dokumentation und Auswertung des Auftragsablaufs im Sinne eines Wissensmanagement. Somit kann bei zukünftigen Aufträgen auf Erfahrungen und erworbene Spezialkenntnisse zurückgegriffen werden.

Aus instrumenteller Sicht lässt sich somit eine Dokumentationsfunktion ableiten, die im Tool berücksichtigt werden sollte. Ergänzend hierzu kann es unter Umständen sinnvoll sein, die Qualität der von den Partnerunternehmen erbrachten Leistungen zu bewerten. Um die Zahlungsabläufe zu unterstützen, kann ein Buchhaltungssystem eingesetzt werden. In Tabelle 4 findet sich eine Gegenüberstellung der Aufgaben und der möglichen Werkzeuge in der Auflösungsphase.

Aufgabe	Instrument
Auftragsdokumentation	Dokumentationssystem
Auftragsüberwachung	Projektmanagementmethoden
Rechnungs- und Zahlungsabwicklung	Buchhaltungssystem

Tabelle 4: Gegenüberstellung von Aufgaben und Instrumenten in der Auslösungsphase

3.4 Werkzeuge für die Unterstützung wichtiger Managementinstrumente

Auch im nachfolgenden Abschnitt wird analog zu Abschnitt 3.3 im Rahmen dieser Veröffentlichung nach Informations-, Vereinbarungs-, Durchführungs- sowie der Auflösungsphase unterschieden.

3.4.1 Informationsphase

In der Informationsphase muss das Netzwerkmanagement Informationen für die spätere Auftragsausschreibung einholen und auswerten. Als wichtigste Aufgabe in dieser Phase ist das Matching zu nennen, also ein Abgleich der für den Auftrag benötigten Kompetenzen und Kapazitäten mit den Kompetenzen und Kapazitäten der Partnerunternehmen. Aus informationstechnischer Sicht bietet sich an, eine Datenbank einzusetzen, in der die Kompetenzen und Kapazitäten der Partnerunternehmen hinterlegt sind. Bei einer Kundenanfrage werden dann automatisch durch das Werkzeug die auftragsspezifisch benötigten Kompetenzen und Kapazitäten mit den Kompetenzen und Kapazitäten der Partnerunternehmen verglichen.

Hierzu ist vom Netzwerkmanagement in Zusammenarbeit mit den Partnerunternehmen ein Kompetenz- und Kapazitätskatalog abzubilden, in dem die Kompetenzen und Kapazitäten der Partnerunternehmen hinterlegt sind. Der Kompetenzkatalog sollte dabei nach Auftragsstypen gegliedert sein, damit dem Netzwerkmanagement bei der Bearbeitung einer Kundenanfrage im Kompetenzkatalog nur die wirklich benötigten Kompetenzen präsentiert werden. Einschränkend ist zu erwähnen, dass sich ein Kompetenzkatalog aus Gründen der Komplexitätsbeschränkung auf die Kernkompetenzen der Partnerunternehmen beschränken sollte. Hier wird ersichtlich, dass von einem System nur standardisierte Anfragen mit einfach zu bestimmenden Kompetenzen bearbeitet werden können. Bei komplexen Anfragen, die ex ante nicht zu antizipierende Kompetenzen benötigen, stößt die Automatisierung durch ein System an die Grenzen. Die Kapazitätsdaten sollten aufgrund des Informationsvorsprungs der Partnerunternehmen von diesen dezentral gepflegt werden. Allerdings ist kritisch anzumerken, dass die Qualität der Kapazitätsdaten nur mit großem Aufwand sichergestellt werden kann, da eine ständige Aktualisierung der Daten durch die Partner notwendig ist und diese oftmals nicht fristgerecht erfolgt.

Auch die Abbildung von Partnerprofilen im System mit der Bewertung der Qualität abgeschlossener Aufträge kann bei der Auswahl geeigneter Partnerunternehmen das Netzwerkmanagement effektiv unterstützen.

3.4.2 Vereinbarungsphase

Für die Vereinbarungsphase wurden zuvor freie Verhandlungen, Auktionen und Börsen unterschieden. Zur der informationstechnischen Unterstützung von freien Verhandlungen existieren in Theorie und Praxis zahlreiche Ansätze. Bereits seit 20 Jahren wird im Forschungsfeld der Künstlichen Intelligenz die Vollautomatisierung von Verhandlungen durch die IT thematisiert, dennoch konnten sich derartige Ansätze, etwa in Form von Multiagentensystemen, in der Praxis bisher nicht etablieren. Das zentrale Problem bei der Vollautomatisierung freier Verhandlungen ist die Komplexität der Verhandlungsparameter. Im Regelfall

werden im Rahmen von Verhandlungen neben dem Preis noch weitere Parameter wie Qualität, Lieferzeitpunkt oder Gewährleistungsansprüche verhandelt, die sich nur schwierig in einem System abbilden lassen. Man spricht in diesem Zusammenhang von multiattributiven bzw. multidimensionalen Verhandlungen.

Auch für Netzwerkunternehmen erscheint die vollautomatisierte Unterstützung freier Verhandlungen durch die Informationstechnologie derzeit noch nicht realisierbar, da - wie oben festgestellt - wenig standardisierte Leistungen mit zahlreichen Parametern verhandelt werden, die sich durch die IT bisher nicht entsprechend abbilden lassen.

Von größerer Relevanz für Netzwerke sind informationstechnisch unterstützte Auktionen, zum Beispiel in Gestalt einer internetbasierten Höchstpreisauktion, bei denen für den Fall hochstandardisierter Leistungen der Preis das einzige verhandelte Attribut ist und alle anderen Leistungskonditionen ex ante fixiert sind. Auf externen Märkten werden derartige Auktionen, meist in Gestalt einer internetbasierten englischen Auktion, bereits erfolgreich eingesetzt (z.B. www.ebay.de). Bei der Anwendung von Auktionen in Netzwerken bietet sich eine verdeckte reverse Auktion als Ausschreibung an. Durch die verdeckte Gebotsabgabe können im Vergleich zu offenen Auktionen, wie oben beschrieben, Transaktionskosten eingespart werden. Erste praktische Erfahrungen mit dem System bestätigen die grundsätzliche Anwendbarkeit dieser Auktionsform. Eine weitere Aufgabe der Vereinbarungsphase ist die Fixierung des Leistungsumfangs. Aus informationstechnischer Sicht bieten sich hierfür elektronische Verträge an, die direkt in ein Auktionssystem integriert werden können.¹⁹

3.4.3 Durchführungphase

Die oben beschriebenen Aufgaben dieser Phase können zumindest teilweise durch Werkzeuge unterstützt werden. So bieten moderne Projektmanagementtools bereits Funktionen, die auch eine verteilte Zusammenarbeit ermöglichen. Mit dem Tool Microsoft Projekt können so zum Beispiel über einen zentralen Server im Rahmen eines Projekts per Internetschnittstelle zentrale Kompetenzen und Kapazitäten der Unternehmen des virtuellen Unternehmens eingepflegt und aktualisiert werden. Da MS Projekt weitgehend als bekannt vorausgesetzt werden kann, wird an dieser Stelle nicht vertiefend darauf eingegangen.

Wesentlich weniger Verbreitung als herkömmliche Projektmanagementtools haben bisher Workflow- bzw. Groupwaresysteme gefunden, die für eine Zusammenarbeit im virtuellen Unternehmen viele interessante Funktionen bereitstellen, die auch teilweise über die Durchführungsphase hinausgehen und den gesamten Kommunikationsfluss abwickeln können. Workflow- und Group-

¹⁹ Vgl. Schopp/Stanoevska-Slabeva (2000).

waresysteme werden unter dem Begriff „Computer Supported Cooperative Work“ zusammengefasst.

Auch der eigentliche Leistungsaustausch kann von Werkzeugen unterstützt werden, wenn die zu erbringende Leistung informationsbasiert ist und im Rahmen eines IT-Systems ausgetauscht werden kann. Ein derartiger Fall kann zum Beispiel die Entwicklung eines Webauftritts für den Kunden sein, bei der das Partnerunternehmen nach Fertigstellung der Leistung die fertigen Dateien per Email oder über andere elektronische Kommunikationskanäle verschicken kann.

Weiterhin sind in dieser Phase elektronische Rechnungen und ein elektronisch abgewickelter Zahlungsverkehr zwischen Netzwerkmanagement und Partnerunternehmen von Bedeutung und können transaktionskostensenkend eingesetzt werden.

3.4.4 Auflösungsphase

Nach dem Austausch der Leistung gilt es die Erfahrungen zu dokumentieren und für zukünftige Transaktionen verfügbar zu machen. Hier bietet sich der Aufbau einer Wissensdatenbank an, im Rahmen derer implizit vorliegende Projekterfahrungen gespeichert und damit zu explizitem Wissen umgewandelt werden.

Insbesondere die Bewertung der Qualität der von den Partnerunternehmen erbrachten Leistungen sollte dem Netzwerkmanagement bei sich anschließenden Entscheidungen über die Vergabe neuer Teilaufträge zur Verfügung stehen. Zur Abbildung eines Qualitätsbewertungsprozesses im System ist die Einführung eines formalisierten Reputationsmechanismus, wie er bereits bei Auktionsplattformen wie Ebay zum Einsatz kommt, denkbar.

4 Softwareunterstützung durch VICOPLAN

In diesem Kapitel wird mit VICOPLAN (Virtual Corporation Planning System) ein Softwaretool zur Unterstützung der managementrelevanten Funktionen in virtuellen Unternehmen vorgestellt. Dieses Softwaretool ist das zentrale Ergebnis des bereits in der Einleitung vorgestellten Projektes „Werkzeuggestütztes Management wissensintensiver Dienstleistungen in virtuellen Unternehmen“. Hierzu werden einleitend im nächsten Abschnitt die Hintergründe des Projekts skizziert. Darauf aufbauend wird im Kapitel 4.2 die Auswahl eines geeigneten Vorgehensmodells dargelegt. Das Kapitel 4.3 fasst die Ergebnisse der Anforderungsanalyse zusammen, die im Kapitel 4.4 in die Softwarekonzeption einfließen. Im Kapitel 4.5 wird dann die eigentliche Softwarerealisierung beschrieben. Abschließend wird dem Leser im Kapitel 4.6 der Funktionsumfang von VICOPLAN erläutert und die Abwicklung eines Auftrags innerhalb des Softwaretools anhand von Screenshots veranschaulicht.

4.1 Hintergrund des Entwicklungsprojektes

Wie bereits in der Einleitung beschrieben, war es das Ziel dieses vom Bundesministerium für Bildung und Forschung geförderten Projektes, ein prototypisches Softwaretool für das Management virtueller Unternehmen zu entwickeln. Für dieses Vorhaben war der Zeitraum von Juni 2001 bis November 2003 vorgesehen. Die Konzeption und technische Umsetzung wurde von einem wissenschaftlichen Mitarbeiter während der gesamten Projektlaufzeit durchgeführt. Für die in Kapitel fünf beschriebene Untersuchung innovativer Technologien wurde ein wissenschaftlicher Mitarbeiter für ein Jahr eingesetzt. Die beiden wissenschaftlichen Mitarbeiter wurden jeweils von studentischen Hilfskräften unterstützt.

Um neben aus der Theorie und Literatur abgeleiteten Anforderungen (vgl. Kapitel drei) praxisorientierte Anforderungen erheben zu können, wurde mit dem virtuellen Unternehmen Techno-Pool aus Groß-Gerau ein Praxispartner gewonnen. Techno-Pool tritt in der Region Rhein-Main als Komplettanbieter von IT-Dienstleistungen, Software und Hardware auf und bietet seinen Kunden einen festen Ansprechpartner (Koordinator), der das Netzwerk nach außen vertritt und

intern für die Sicherstellung einer zeit- und qualitätsgerechten Auftragsdurchführung durch die Partnerunternehmen verantwortlich ist. Techno-Pool formulierte regelmäßig in Workshops Praxisanforderungen an das Softwaretool und unterzog die jeweiligen Projektergebnisse einer kritischen Prüfung. Aufgrund hoher Belastungen durch das Tagesgeschäft war es Techno-Pool jedoch nicht möglich, dem Entwicklungsprozess ständig beizuwohnen.

Vor dem Hintergrund der begrenzten Personalressourcen, des recht engen Zeitrahmens und der beschränkten Verfügbarkeit von Techno-Pool war es notwendig, ein Vorgehensmodell für die Entwicklung auszuwählen, das diese besonderen Restriktionen berücksichtigt. Diese Auswahl ist im folgenden Kapitel dargestellt.

4.2 Auswahl eines Vorgehensmodells für die Entwicklung

„Softwareentwicklung befasst sich mit „der Planung, dem Entwurf und der Implementierung von Softwaresystemen“²⁰ und findet im Rahmen eines Softwareentwicklungsprozesses (synonym Softwarelebenszyklus²¹) statt.²² Ein Entwicklungsschema umfasst alles Wissen einer soziotechnischen Umgebung über die Gestaltung und Betreuung eines Software-Systems. Das Vorgehensmodell (Abkürzung VGM, synonym Prozessmodell) ist eine Beschreibung eines Softwareentwicklungsprozesses auf Basis eines Entwicklungsschemas (synonym Vorgehensstrategie). Es liegt zunächst als Referenzmodell²³ bzw. generisches Vorgehensmodell vor und muss an das konkrete Projekt angepasst werden („Tailoring“, vgl. Abbildung 5).²⁴

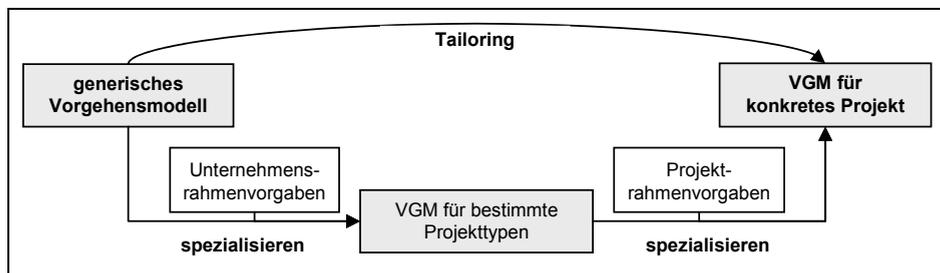


Abbildung 5: Tailoring

²⁰ Vgl. Schader / Rundshagen (1996), S. 2.

²¹ Der Entwicklungsprozess umfasst optimalerweise auch Wartung und Pflege der Software.

²² Zu diesem Absatz vgl. Fischer / Biskup / Müller-Luschnat (1998), S. 18.

²³ „modellhafte, abstrahierende Beschreibung von Vorgehensweisen, Richtlinien, Empfehlungen oder Prozessen, die für einen abgegrenzten Problembereich gelten und in einer möglichst großen Zahl von Einzelfällen anwendbar sind“, Stahlknecht / Hasenkamp (2002), S. 219.

²⁴ Vgl. Fischer / Biskup / Müller-Luschnat (1998), S. 28.

Im konkreten Projekt wurde die Software nicht im Team, sondern im Wesentlichen von einer Einzelperson entwickelt und hatte ex ante schwer zu definierende Anforderungen. Außerdem hatte der Praxispartner zu Beginn des Projektes noch keine präzise Vorstellung von der fertigen Software. So wurde ein „leichtes“ Vorgehensmodell benötigt, welches von einer Person handhabbar ist, dem eine ex ante Grobspezifikation des Projekts genügt, das Erweiterungen und Korrekturen an dieser gestattet, eine modularisierte Entwicklung unterstützt (zur Teilrealisierung), eine Dokumentation in ausreichendem Maße vorsieht und in einem Zeitraum von ca. 2 Jahren zum Abschluss führen kann.

Aus den in der Literatur diskutierten, etablierten Vorgehensmodellen war nun ein geeignetes Modell auszuwählen. Die wichtigsten Vertreter sind das Wasserfallmodell, das Spiralmodell, das V-Modell, die evolutionäre Entwicklung, die inkrementelle Entwicklung, Prototyping, Agile Entwicklung, das Baseballmodell, das Fontänenmodell und der Rational Unified Process (RUP).

Nach dem Ausschlussprinzip wurden zunächst in drei Schritten (nach Rahmenbedingungen, Zielen und Restriktionen) die geeigneten Modelle isoliert. Abbildung 6 veranschaulicht dieses Verfahren.

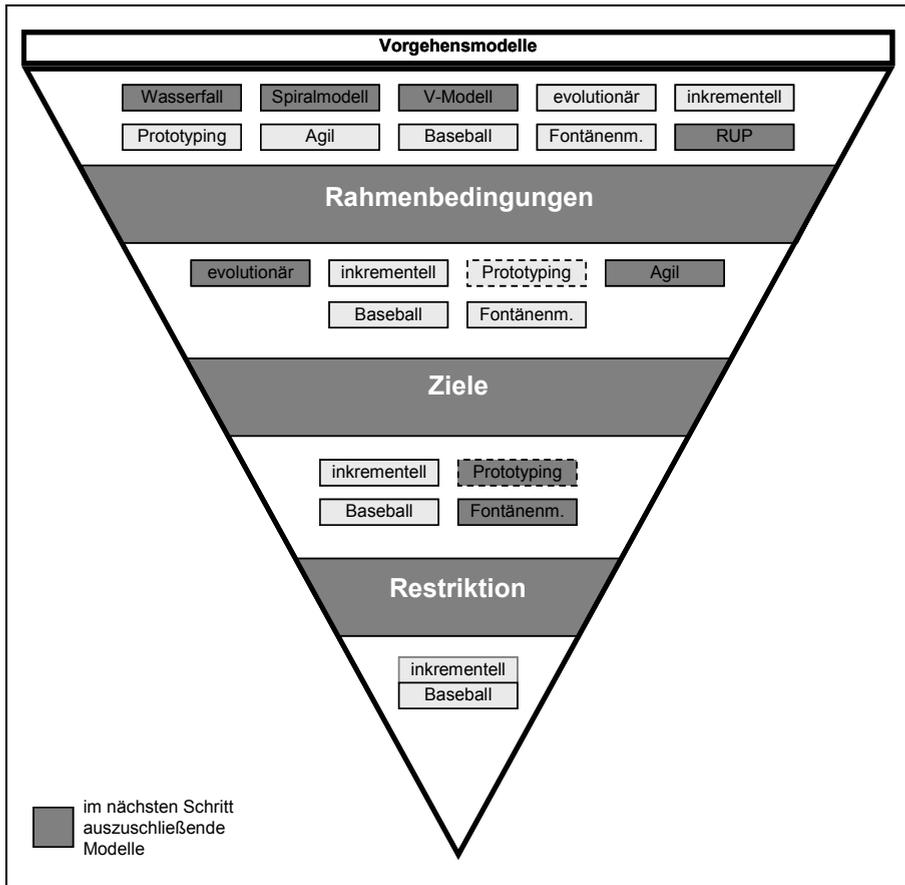


Abbildung 6: Auswahl Vorgehensmodell

Durch die unklaren Anforderungen ist ein sequentielles Vorgehen wie im Wasserfall- oder Spiralmodell nicht möglich. Da das Projekt von einer einzelnen Person durchgeführt wird, sind die umfassenden Risiko- und Projektmanagementregelungen von Spiralmodell, V-Modell und RUP nicht notwendig.

Evolutionäre sowie Agile Entwicklung sind mit den Projektzielen inkompatibel. Da zu Beginn ein umfassendes Konzept zu erstellen war, ist eine evolutionäre Vorgehensweise unpassend. Agile Entwicklung stellt die Dokumentation in den Hintergrund, was im vorliegenden Fall einen Ausschlussgrund darstellte, da die Ergebnisse nur in dokumentierter Form auch für andere virtuelle Unternehmen nutzbar sind. Werden die beschränkten Ressourcen und die nur unregelmäßige Verfügbarkeit des Praxispartners Techno-Pool berücksichtigt, schien ein ressourcenintensives Prototyping als Gesamtansatz problematisch. Es bestand jedoch grundsätzlich die Möglichkeit in bestimmten Bereichen gezielt exploratives oder experimentelles Prototyping einzusetzen. Das

Fontänenmodell definiert sechs Entwicklungsphasen mit Iterationen, was verglichen mit dem Baseballmodell den kurzen Entwicklungsprozess unnötigerweise zergliedert.

Es verbleiben zwei geeignete Modelle, die inkrementelle Entwicklung (IE) und das Baseballmodell (BM). Da die inkrementelle Entwicklung im Wesentlichen auf die Erstellung und Verfeinerung von Inkrementen abzielt, im Projekt jedoch aufgrund von beschränkter Verfügbarkeit des Praxispartners keine zahlreichen Reviews von Inkrementen möglich waren, wurde das Baseball-Modell ausgewählt, das als Vertreter eines objektorientierten Vorgehens nachfolgend detaillierter dargestellt wird.

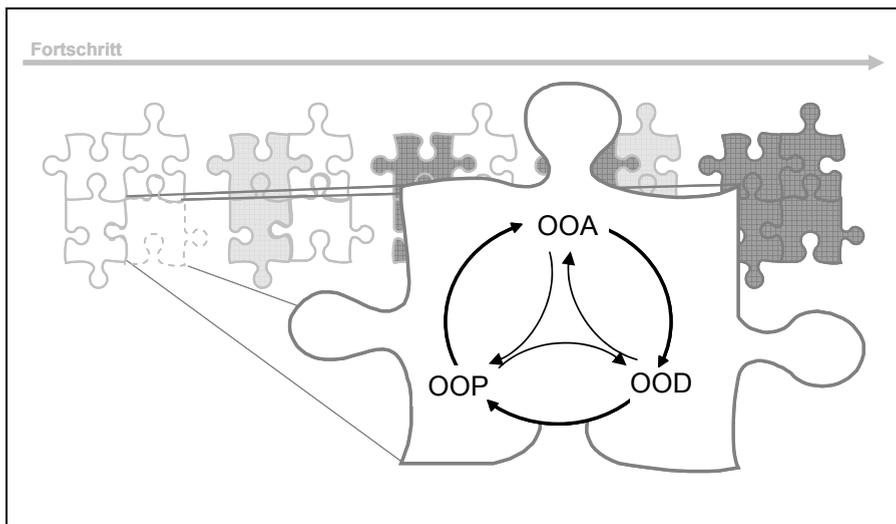


Abbildung 7: Baseballmodell

In objektorientierten Modellen lassen sich drei Grundphasen identifizieren:²⁵ „Objektorientierte Analyse“ (OOA), „Objektorientiertes Design“ (OOD) und „Objektorientierte Programmierung“ (OOP). Während die OOA Objekte des Problembereichs (der realen Welt) beschreibt und analysiert, versucht das OOD diese Objekte in den Lösungsbereich (DV-System) zu übertragen und diesem anzupassen. Die OOP kodiert die ermittelten Objekte schließlich, damit sie im Lösungsbereich ausgeführt werden können. Der iterierende Wechsel von OOA, OOD, OOP ist als Baseballmodell bekannt (Abbildung 7). Hauptmerkmal ist „eine schnelle, innovative Codeerzeugung, die rasch erste greifbare (wenn auch nur partielle) Resultate erbringt und die Diskussion mit den Systemanwendern auf

²⁵ Zu diesem Absatz vgl. Schader / Rundshagen (1996), S. 30-31 oder Biethahn / Muksch / Ruf (2000), S. 293-296.

eine konkretere Basis stellt“²⁶. Durch die Auswahl des Baseball-Modells ergaben sich folgende Implikationen:

- Die Software ist im Verlauf des Projekts wiederholt dem Kunden zu Testzwecken zu übergeben, damit dessen Erfahrungswerte in die weitere Entwicklung einfließen.²⁷
- Das Projekt ist mit objektorientierten Methoden durchzuführen.
- Die Dokumentation erfolgt durch Verankerung des Produktmodells im Programmcode und Assoziation von Ergebnisdokumenten (z.B. UML-Modellen) mit diesen Strukturkomponenten.
- Der Entwicklungsprozess ist in zwei Grobbereiche zu gliedern: Im Rahmen einer Grundlegenden objektorientierten Analyse ist der Problembereich zunächst fachlichformal zu spezifizieren. Dieses Ausgangsmodell wird im Verlauf des Projekts iterativ vervollständigt und detailliert (OOA/OOD/OOP-Zyklen).
- Die gewählte Entwicklung verlangt zunächst eine möglichst vollständige Produktspezifikation. Die Produktspezifikation kann in diesem Stadium durch Pflichtenheft und Produktmodell von fachlicher, durch die Softwarearchitektur von technischer Seite beschrieben werden.

An dieser Stelle beginnt die Entwicklung nach Baseballmodell. Die objektorientierte Analyse des ersten zu realisierenden Systems wird vervollständigt und der entsprechende Bereich des Systems softwaretechnisch spezifiziert (OOD). Diese Schritte können zusammen mit der objektorientierten Programmierung (OOP) in mehreren Iterationen erfolgen. Mit Abschluss dieser Arbeiten war ein Release entstanden, das dem Praxispartner zum Review vorzulegen war. Die oben beschriebenen Iterationen führten zu einer zunehmenden Verbesserung und Annäherung der Software an die tatsächlichen Anforderungen.

4.3 Anforderungsanalyse

Es wird dem Vorgehen nach Mertens et. al.²⁸ zur Ist-Analyse gefolgt. Ähnliche Vorgehensbeschreibungen finden sich bei Heib²⁹ und Stahlknecht/Hasenkamp³⁰. Der Erhebungsbereich umfasst den Geschäftsbereich des Techno-Pool Netzwerks. Es sollen die Prozesse „Auftragsablauf“ und „Analyse“ beschrieben

²⁶ Schader / Rundshagen (1996), S. 30.

²⁷ Vgl. Balzert (2000), S. 58.

²⁸ Vgl. Mertens / Bodendorf / König / Picot / Schumann (2001), S. 172-173.

²⁹ Vgl. Heib (1998), S. 150-152.

³⁰ Vgl. Stahlknecht / Hasenkamp (2002), S. 229.

werden. Die Erhebung wurde über Dokumentenanalyse, Interviews und Workshops mit den potentiellen Nutzern durchgeführt.

Zur besseren Übersicht soll nach verschiedenen Sichten gegliedert werden:³¹ (Aufbau-) Organisationssicht, Funktionssicht und Datensicht. Im Folgenden werden die Resultate des Entwicklungsprozesses den drei Sichten zugeordnet. Dabei ist anzumerken, dass die Beschreibung nur den Idealablauf (ca. 70% der Fälle) umfasst. Bei etwa 30% der abgewickelten Prozesse treten Besonderheiten auf, die eine sinnvolle IT-Unterstützung stark erschweren.

4.3.1 Organisationssicht

Die nachfolgende Abbildung gibt eine Übersicht über die organisatorische Aufteilung des virtuellen Unternehmens Techno-Pool. Bedingt durch die Organisationsform eines virtuellen Unternehmens, ist die Struktur sehr schlank und weist nur zwei Hierarchieebenen auf, die projektabhängig um eine Zwischenebene ergänzt werden können. An der Spitze des Praxispartners Techno-Pool steht ein Koordinator (Broker), der projektabhängig von einem Auftragskoordinator unterstützt werden kann. Da der Auftragskoordinator systemtechnisch wie ein Koordinator zu behandeln ist, wird im weiteren Verlauf nicht zwischen diesen Rollen unterschieden. Die zweite Ebene bilden die Partnerunternehmen, die im Rahmen von Netzwerkaufträgen vom Koordinator aktiviert werden.

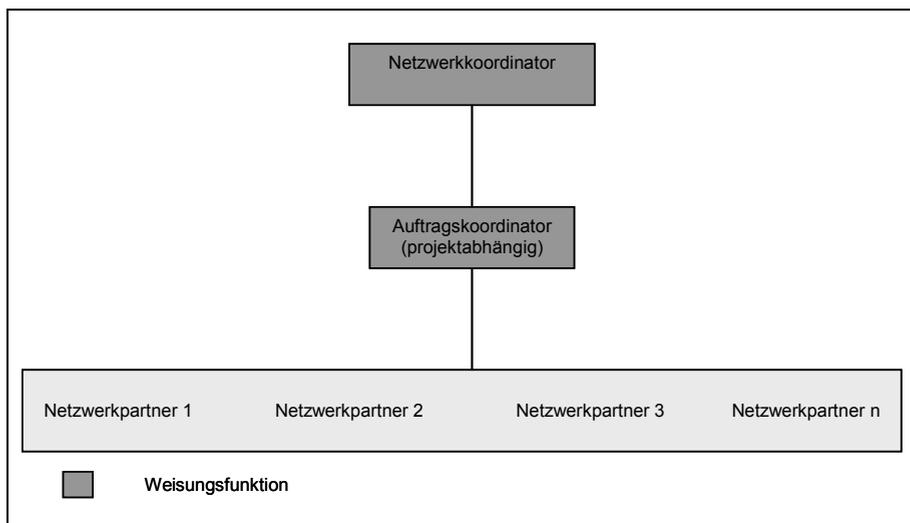


Abbildung 8: Organisationssicht

³¹ Vgl. Scheer (1998), S. 33-34, die Leistungssicht ist hier zu vernachlässigen.

4.3.2 Funktionssicht

An dieser Stelle werden die zu unterstützenden Funktionen anhand eines Auftragslebenszyklus dargestellt, wie er im Rahmen der Anforderungsanalyse bei Techno-Pool erhoben wurde. Dabei werden nur die wesentlichen Prozessschritte abgebildet, da der Ablauf in der späteren Systemspezifikation detaillierter dargestellt wird.

Es lassen sich sieben grundlegend zu unterstützenden Funktionen identifizieren, die in der Abbildung 9 dargestellt und nachfolgend aufgeführt werden:

1. Ein Kunde stellt eine Anfrage an das Netzwerk:
2. Der Netzwerkkoordinator teilt die Anfrage in Arbeitspakete mit fest definierten Kompetenzen auf und schreibt diese in einem netzwerk-internen Markt aus. Die Partner geben im Rahmen dieser Ausschreibung ein Angebot für die Durchführung von Teilleistungen ab.
3. Der Netzwerkkoordinator wählt unter Berücksichtigung des Angebotspreises der Partner und weiterer Kriterien (Referenzen, Qualität in abgeschlossenen Aufträgen etc.) Teilangebote aus, die er in die Kalkulation eines Kundenangebots übernimmt.
4. Die Teilkalkulationen der Partner werden zu einem Kundenangebot aggregiert.
5. Das Kundenangebot wird dem Kunden unterbreitet.
6. Der Kunde beauftragt das Netzwerk.
7. Der Koordinator bildet eine auftragsspezifische Wertschöpfungskette und führt eine Auftragsüberwachung durch.

Ergänzend zu diesen auftragsbezogenen Aufgaben fallen weitere auftragsübergreifende Aufgaben an. Die Schwerpunkte, die Techno-Pool für die Berücksichtigung im späteren System wichtig waren, sind dabei die Bewertung von Partnerunternehmen nach dem Abschluss von Aufträgen und eine Analyse der Netzwerkumsätze aufgeteilt nach Partnerunternehmen.

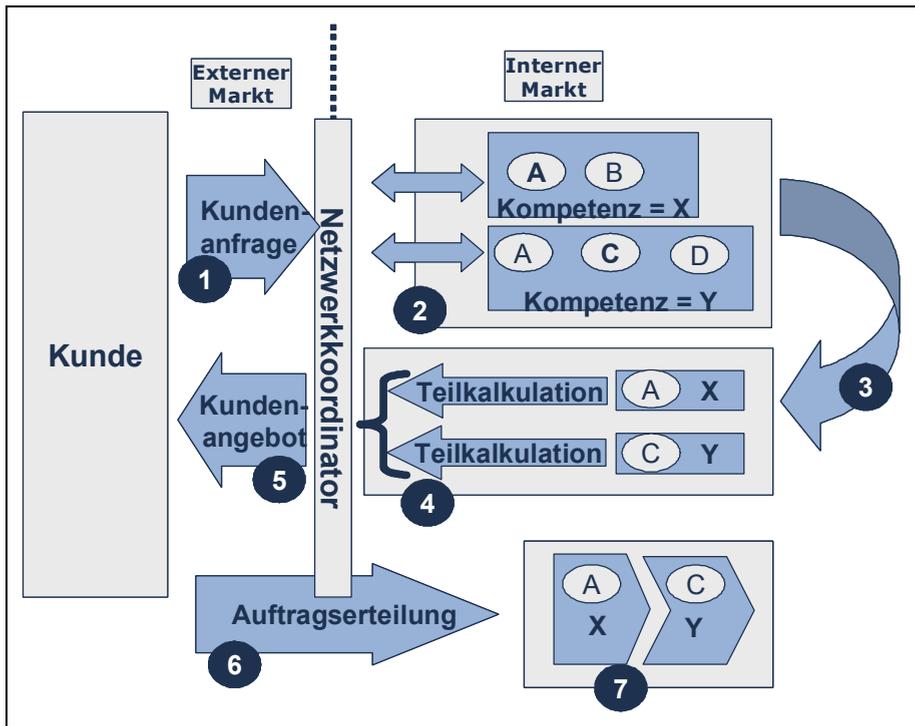


Abbildung 9: Funktionssicht Auftragslebenszyklus

4.3.3 Datensicht

Die oben dargestellte Funktionssicht wurde losgelöst von vorhandenen IT-Systemen und Datenfluss beschrieben. Bezieht man diese in die Untersuchung mit ein, werden folgende Problemfelder erkennbar:

- Die Kundenanfrage wird dateiorientiert in einer Excel-Maske erfasst und für die sich anschließende Ausschreibung nicht verwendet.
- Die Ausschreibung erfolgt je nach Partnerunternehmen auf verschiedene Medien verteilt (Email, Fax, Telefon).
- Die Teilangebote der Partnerunternehmen treffen in verschiedenen Datenformaten ein (E-Mail, Email-Anhang in Excel/Word, Fax, schriftlich) und müssen für das spätere Kundenangebot in ein Format überführt werden.
- Das Auftreten von mehreren Medienbrüchen erschwert die Statusdokumentation und damit die Koordination.

- Wenn keine aktuelle Übersichtsliste verfügbar ist gerät der Prozess durch notwendige Email-Rückfragen ins stocken. Die Abwicklungsdauer steigt um die Antwort-Wartezeit.

Anhand des oben dargestellten Auftragslebenszyklus und aufgrund der identifizierten Schwachstellen ergibt sich ein erster, grober Anforderungskatalog an das Softwaresystem. Dieser stellt das Lastenheft für das Entwicklungsvorhaben dar und repräsentiert das Ergebnisdokument „Anforderungsanalyse“. Das Lastenheft findet unter Berücksichtigung von Projektrestriktionen und weiterer Analyseergebnisse Eingang im Pflichtenheft. Alle wesentlichen geforderten Funktionalitäten wurden in das Pflichtenheft übernommen, daher sei an dieser Stelle auf das Pflichtenheft verwiesen.

4.4 Softwarekonzeption

In diesem Kapitel wird die Systemanalyse durchgeführt.³² Die Software soll spezifiziert und die weiteren Entwicklungsschritte geplant werden. Zu diesem Zweck werden zwei Sichtweisen herangezogen: die fachliche oder Kundensicht, aus der Leistungsanforderungen und fachliche Funktionalität definiert werden und die technische oder DV-Sicht, die den technischen Aufbau der Software beschreibt.

4.4.1 Fachliche Spezifikation

Im Rahmen der fachlichen Spezifikation sind als zentrale Ergebnisse zu erstellen: das Pflichtenheft und, darauf aufbauend, das Produktmodell (fachliche Spezifikation im engeren Sinne).

Das Pflichtenheft umfasst eine detaillierte, hauptsächlich verbale Beschreibung der zu erstellenden Software³³ und definiert somit die Zielkriterien des Entwicklungsprozesses. Das Produktmodell ist eine formale Betrachtung der Anforderungen³⁴ und wird im vorliegenden Fall durch die objektorientierte Analyse ermittelt.

Um das Pflichtenheft systematisch zusammenzustellen soll vor der weiteren Erhebung von Anforderungen aus Ist-Prozess und identifizierten Schwachstellen ein Soll-Prozess entwickelt werden, der die Produktfunktionen zum Teil³⁵ definiert und die Grundlage für die Abgrenzung von Anwendungsfällen (use-cases) liefert. Das Resultat ist in der nachfolgend abgebildeten ereignisgesteuerten

³² Vgl. hierzu Balzert (2000), S. 118.

³³ Vgl. Balzert (2000), S. 111.

³⁴ Balzert (2000), S. 118.

³⁵ Das Pflichtenheft enthält zusätzlich eine kurze verbale Beschreibung.

Prozesskette abgebildet. Die Semantik der ereignisgesteuerten Prozessketten wurde gewählt, da der Praxispartner mit dieser bewährten Modellierungsmethode vertraut ist und daher bei der Entwicklung und Verifizierung der Prozesse unterstützend tätig werden konnte. In diesem Fall wurde aus Gründen der Praktikabilität in der Zusammenarbeit mit dem Praxispartner vom sonst streng konsequent verfolgten objektorientierten Paradigma abgewichen, allerdings ohne die Prozessergebnisse zu verändern. Im Anhang sind überblicksartig einige Aktivitätsdiagramme des objektorientierten Vorgehens dargestellt, die an dieser Stelle anstatt der ereignisgesteuerten Prozessketten eingesetzt werden.

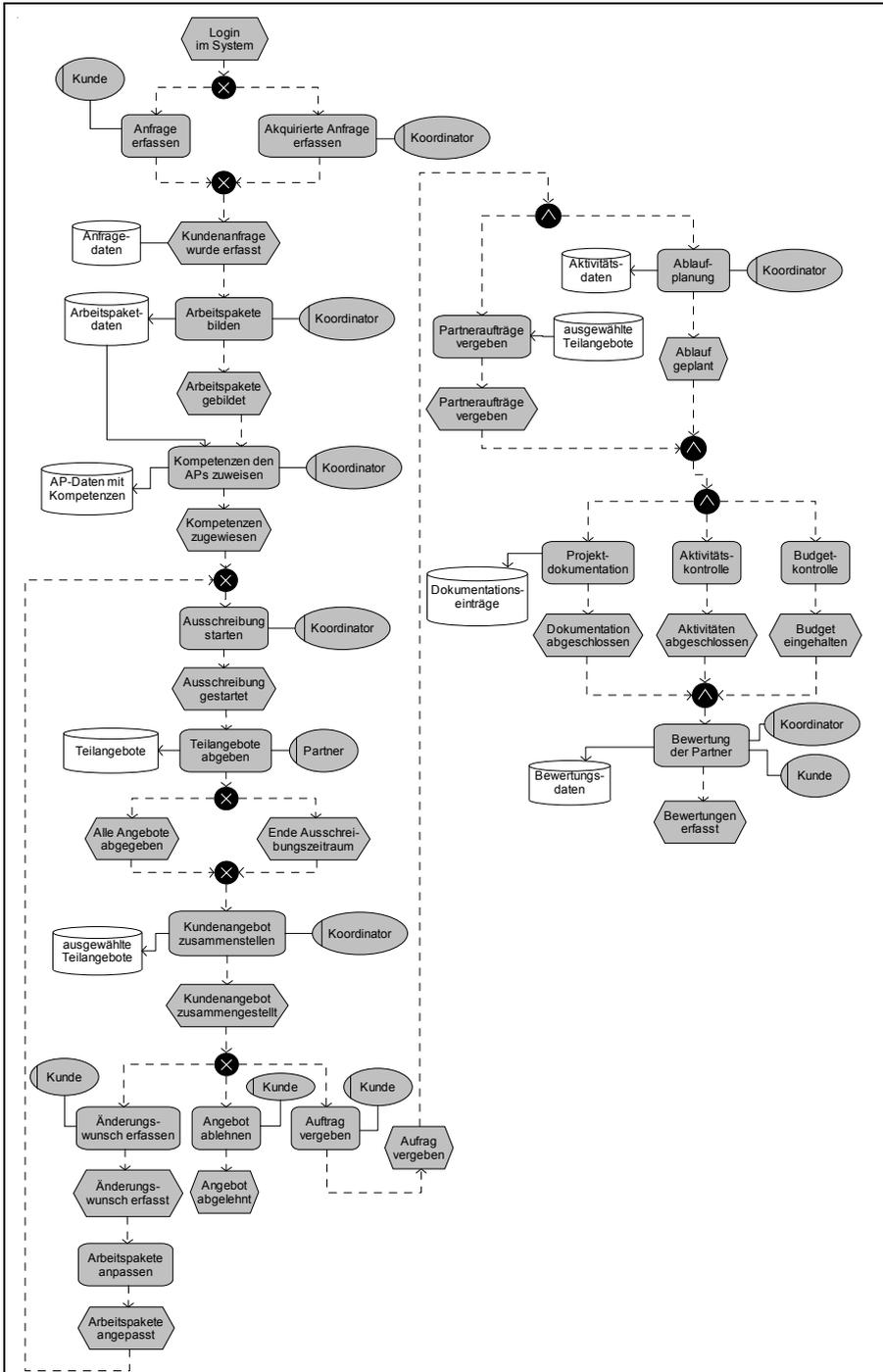


Abbildung 10: Ereignisgesteuerte Prozesskette als Soll-Prozess

Aus der Erarbeitung dieses Soll-Prozesses lassen sich nun konkrete Funktionen ableiten, die das System unterstützen soll.

Funktion	Beschreibung
F10 Kundenanfrage im System erfassen	Der Kunde soll über eine Maske (z.B. im Internetauftritt des Unternehmensnetzwerks integriert) eine Anfrage in das System einstellen.
F20 Kunden- und Anfragedaten anzeigen	In einem Informationsmodul sollen sämtliche Anfrage-, Auftrags- und Kundendaten in verschiedenen Sichten (Liste, Detailfenster) abrufbar sein.
F30 Datei Up- und Download	Dateien sollen mit Kundenanfragen, Aufträgen und Partnerangeboten assoziiert werden können.
F40 Definition von Arbeitspaketen	Zu einer Kundenanfrage sollen Arbeitspakete definiert werden können, die bestimmte Kompetenzen erfordern.
F50 Ausschreibungsfunktionalität	Das System soll anhand der benötigten Kompetenzen für jedes Arbeitspaket die in Frage kommenden Partner ermitteln und diese von der Ausschreibung benachrichtigen.
F55 Internes Angebot erstellen	Während der Laufzeit einer Ausschreibung soll es den beteiligten Partnern möglich sein, im System in standardisierter Form ihr Angebot zu erfassen. Sie sollen einzelne Angebotspositionen mit Mengenangaben und Einzelpreisen in tabellarischer Form eingeben können.
F60 Kundenangebot erstellen	Der Koordinator soll aus den in /F55/ erfassten Partnerangeboten ein Kundenangebot zusammenstellen können. Hierfür sollen in einer Übersichtsmaske die einzelnen Anbieter gelistet werden, woraus der Koordinator dann ein Angebot in das Kundenangebot eingefügt werden kann. Zusätzlich soll es möglich sein, für die einzelnen internen Angebote einen prozentualen Koordinationsaufschlag festzulegen.
F70 Auftragsvergabe	Der Kunde kann das Angebot online (PDF-Format) einsehen und ggf. den Auftrag bestätigen. In diesem Fall soll dem Kunden zusätzlich ein Auftragsbestätigungsfeld übermitteln werden, welches ergänzend in Schriftform einzusenden ist.

F80 Zeitliche Ablaufplanung für Aufträge	Ist ein Auftrag vergeben, so sollen die einzelnen Arbeitspakete in Aktivitäten zerlegt werden, um die zeitliche Abfolge festzulegen (z.B. Paket „Projektcontrolling“ in Aktivitäten „Anforderungsanalyse“, „Abschlusskontrolle“ ...)
F90 Budgetplanung und -kontrolle	Die durch die Partnerangebote feststehenden Sollkosten für die einzelnen Arbeitspakete sollen den tatsächlich anfallenden Kosten gegenübergestellt werden.
F100 Dokumentationssystem	In einer Art Forum sollen auftragsbezogene Ereignisse dokumentiert werden.
F110 Analysemodul	Das System soll aus den Daten der abgeschlossenen Aufträge Auswertungen über Kunden, Partner und Kompetenzen ermöglichen.

Tabelle 5: Notwendige Systemfunktionen von VICOPLAN

Um Anwendungsfälle abzugrenzen, werden nun Prozessabschnitte zusammengefasst: Im use-case-Diagramm werden die Anwendungsfälle den identifizierten Akteuren (Technopool, Partner, Kunde) zugeordnet.

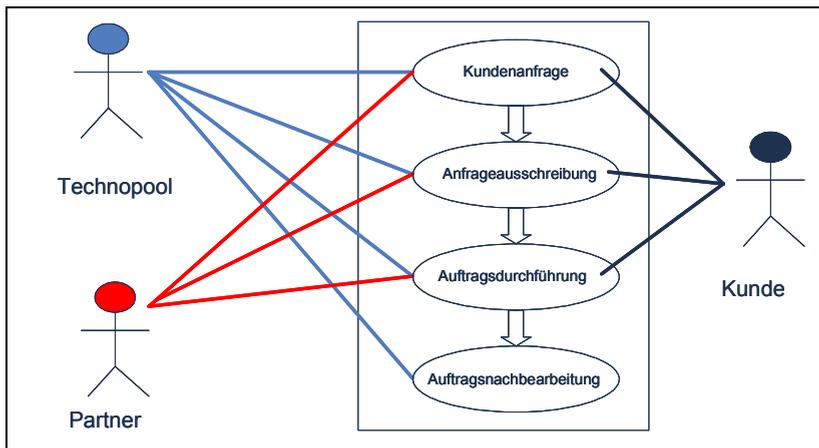


Abbildung 11: Anwendungsfalldiagramm

Auf diese Diskussionsgrundlage aufsetzend wurden in einem Workshop weitere Daten erhoben. Basierend auf diesen Informationen, dem Soll-Prozess und dem Anwendungsfalldiagramm wird nun das Pflichtenheft zusammengestellt. Das IEEE³⁶ hat in Form der „Software Requirements Specification“ (SRS) einen umfassenden Gliederungsvorschlag herausgegeben, der allerdings für das vor-

³⁶ IEEE: Institute of Electrical and Electronics Engineers, www.ieee.org.

liegende Projekt zu mächtig ist. Aus diesem Grund wurde das vereinfachte Schema von Balzert³⁷ als Vorlage genommen. Das so erstellte Pflichtenheft findet sich in einer Kurzzusammenfassung in Anhang.

Ziel der nachfolgenden objektorientierten Analyse ist die Beschreibung der Objekte im Problembereich und damit die Vorbereitung von objektorientiertem Design („Entwurf“) und objektorientierter Programmierung („Implementierung“).

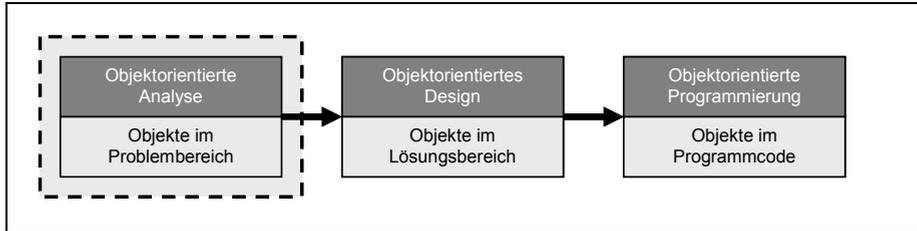


Abbildung 12: Zusammenhang von OOA, OOD und OOP³⁸

Der Problembereich soll aus verschiedenen Perspektiven analysiert werden. In der Literatur finden sich bezüglich Notation, Sichtengliederung und durchzuführenden Arbeitsschritten verschiedene Ansätze.³⁹ Betreffend die Notation hat sich in den letzten Jahren die Unified Modeling Language (UML) als Standard etabliert.⁴⁰ Die Gliederung in verschiedene Sichten ist üblich und variantenreich. Grundsätzlich kann zwischen einer statischen (struktureller Aufbau, Klassen) und einer dynamischen (Objektkommunikation, Ablauf) Sicht unterschieden werden. Über die durchzuführenden Arbeitsschritte herrscht in der durchgesehenen Literatur Übereinstimmung. So sind Klassen, Objekte, Strukturen, Attribute und Methoden zu identifizieren, die Beziehungen und Kommunikation zwischen Objekten zu beschreiben und wenn möglich eine Einteilung in Pakete vorzunehmen.⁴¹

³⁷ Vgl. Balzert (2000), S. 114-118.

³⁸ In Anlehnung an Schader / Rundshagen (1996), S. 31.

³⁹ Vgl. Coad / Yourdon (1991), S. 51,79,106,119,143, Neumann (2002), S. 203, Balzert (1996), S. 29-30, Balzert (2000), S. 386-391, Schader / Rundshagen (1996), S. 39-152, Volkmer (1997), S. 26-29, Schach (2002), S. 366-382, Oestereich (1998), S. 121-160.

⁴⁰ Vgl. o.A. (2003a).

⁴¹ Vgl. auch Schader / Rundshagen (1996), S. 49, die Paketbildung ist in diesem Fall aufgrund geringer Komplexität nicht notwendig.

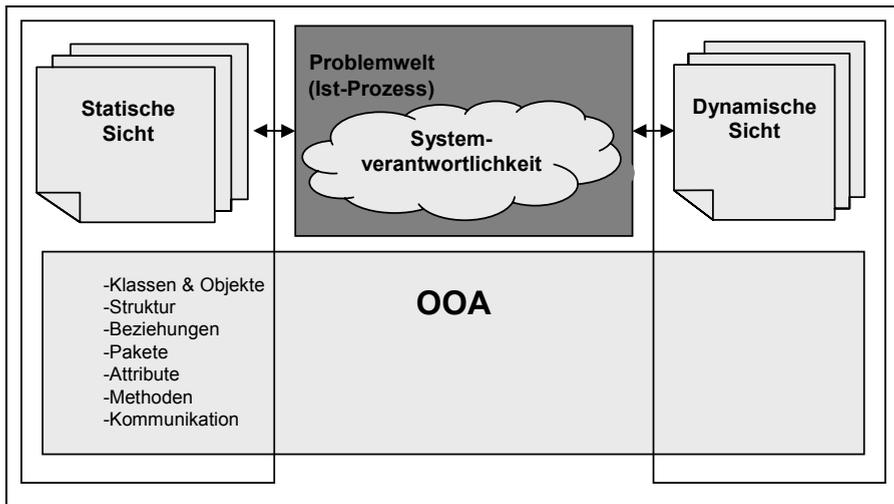


Abbildung 13: Sichten und Gliederung der OOA

Konkreter Gegenstand der OOA ist im vorliegenden Fall der Teil des Prozesses, der im Verantwortlichkeitsbereich des späteren Systems liegt. Zur Durchführung der Arbeitsschritte der OOA wird den praxisnahen Anweisungen von Coad/Yourdon⁴², Schach⁴³ und Balzert⁴⁴ gefolgt.

Den genannten Autoren zufolge sind der zukünftige Verantwortlichkeitsbereich der Software und bereits vorhandene Analysedokumente systematisch auf Klassen abzusuchen. Dabei soll insbesondere auf organisatorische Einheiten, Ereignisse, „benachbarte“ Systeme, häufig vorkommende Substantive etc. geachtet werden. Bevor die auf diese Weise identifizierten Klassen jedoch als solche akzeptiert werden, sind sie anhand einer Liste von Prüfkriterien⁴⁵ zu filtern. Nach Coad/Yourdon (1991) sind im nächsten Schritt Vererbungen und Aggregationen zu beschreiben und weitere Klassen in diesem Kontext zu suchen. Hierzu sind die bereits bekannten Klassen als potentielle Spezialisierung bzw. als potentieller Teil eines Ganzen anzusehen und entsprechende Generalisierungen bzw. „Ganzes“-Konstrukte zu suchen. Analog ist der Analysebereich auf Spezialisierungen und „Teil-eines-Ganzen“-Konstrukte zu prüfen.

Als Ergebnis dieses Vorgehens wurden folgende Klassen definiert:

- **User**

Hier werden die Systemnutzer erfasst. Dabei werden Kunden, Partner und Broker (Koordinator) unterschieden und mit unterschiedlichen Attributen sowie Methoden versehen.

⁴² Vgl. Coad / Yourdon (1991), S. 52-118.

⁴³ Vgl. Schach (2002), S. 371-375.

⁴⁴ Vgl. Balzert (1996), S. 31-91.

⁴⁵ Vgl. Coad / Yourdon (1991), S. 66-72.

- **Process**

Diese Klasse bildet den Kern des Systems ab und wird mit dem Anlegen einer Kundenanfrage bis zum Abschluss des Auftrags aktiviert. Die Attribute geben die benötigten Informationen wider, die zunächst bei einer Anfrage (z.B. description, startwunsch, location), bei der Ausschreibung (startsoll) später beim Auftrag (startist) im Prozess fortlaufend ergänzt werden. Diese geben Auskunft über den Status des Prozesses. Durch die Methode „change state“ wird der Status verändert.
- **Attachment**

Dort werden Dateianlagen verwaltet.
- **Package**

Dies sind die Arbeitspakete (mit Datum, Titel), die vom Broker intern ausgeschrieben und später zu internen Teilaufträgen werden. Diese Arbeitspakete können ausgeschrieben werden (publish), vom Broker zurückgerufen werden (callback) oder beim Erreichen des Endes der Ausschreibung auslaufen (expires).
- **Competency**

Hier werden Kompetenzen hinterlegt, die einen Partner oder einem Package zugeordnet werden.
- **InternalOffer**

Diese Klasse sind die internen Angebote der Partnerunternehmen, die sich jeweils auf die Klasse package beziehen.
- **IntOfferPosition**

Das sind die Angebotspositionen, die zum InternalOffer aggregiert werden.
- **CustomerOffer**

Hier wird das Kundenangebot abgebildet, das mit einem eindeutigen Datum und einer Signatur versehen wird.

Auf die bisherigen Analyseergebnisse aufbauend können nun Beziehungen (synonym Assoziationen, engl. „instance connections“) und deren Kardinalitäten (engl. „multiplicity“) eingetragen werden. Sie sind intuitiv erfassbar.

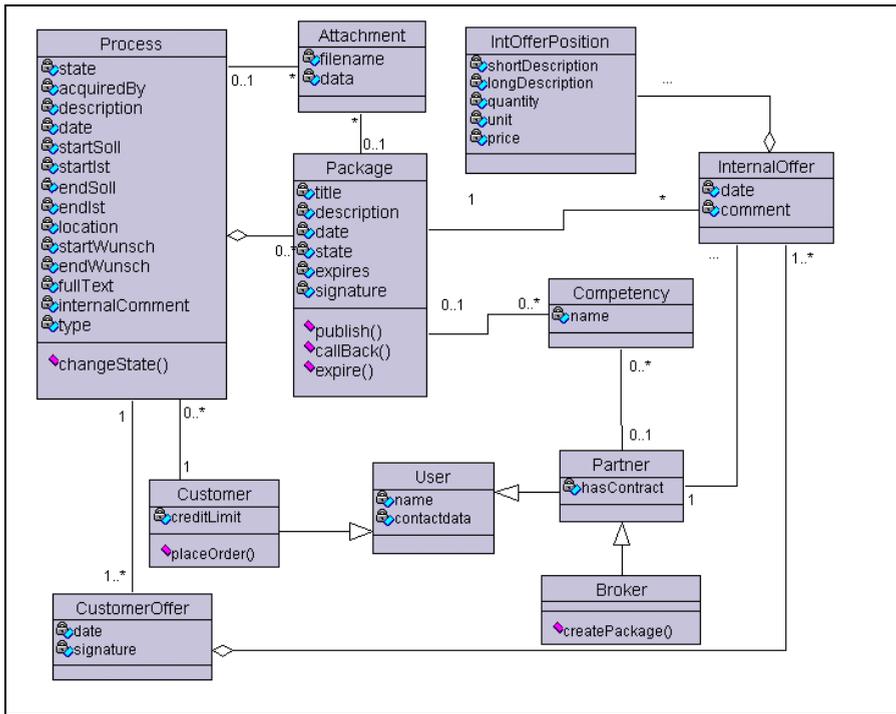


Abbildung 14: Zentrales Klassenmodell

Die geforderte Dokumentations- und Analysefunktion ist im nachfolgenden Diagramm dargestellt. Das zentrale Element stellt die Prozessdokumentation dar, die mit einer Aktivitätskontrolle und verschiedenen Dokumentationseinträgen (Kundenzahlungen, Aktivitätsbeginn/-ende, Partnerbewertung, Leistungsnachweisen) verknüpft ist.

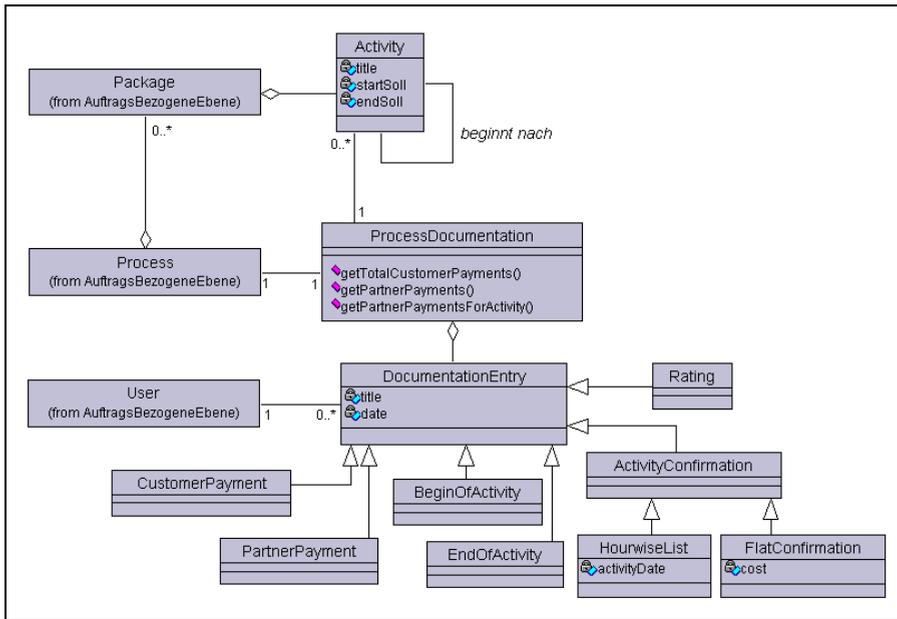


Abbildung 15: Klassenmodell Dokumentation

Eine weitere umzusetzende Funktion ist die im Pflichtenheft geforderte E-Mail-Benachrichtigung bei organisatorischen Brüchen. Da der Prozess diese Benachrichtigung auslöst, wurde diese Funktionalität der Prozesskomponente zugeordnet. Sie kann folgendermaßen realisiert werden (siehe Abbildung 16):

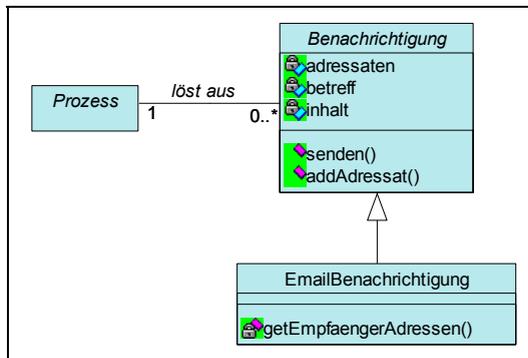


Abbildung 16: E-Mail-Benachrichtigung

Die Vererbungsstruktur wurde eingeführt, um später weitere Benachrichtigungstypen (z.B. SMS, Fax) hinzufügen zu können.

4.4.2 Technische Spezifikation - Softwarearchitektur

Die Softwarearchitektur beschreibt die Struktur des Software-Systems durch Spezifizierung von Komponenten und deren Beziehungen untereinander. Unter einer Komponente ist ein Ausschnitt der Softwarestruktur zu verstehen, der weitere Komponenten oder Klassen enthält und sich durch hohe Kohäsion und geringe Kopplung mit anderen Komponenten auszeichnet. Statische oder dynamische Beziehungen verbinden Komponenten über exakt zu definierende Schnittstellen. Um die Software weiter zu strukturieren, können Komponenten zu Schichten zusammengefasst werden. Ein klassisches Beispiel ist die Gliederung in Datenhaltungsschicht, Anwendungsschicht und Präsentationsschicht.

Ausgangspunkt für die folgenden Betrachtungen sind die technischen Rahmenbedingungen und die fachliche Grobgliederung des Systems aus dem Pflichtenheft (vgl. Anhang) sowie der bereits erstellte Teil des statischen Produktmodells. Über diese fachlichen Informationen hinaus müssen Benutzerinterface, Datenbank und Fehlerbehandlung mit einbezogen werden.

Der Verantwortlichkeitsbereich der Software umfasst im vorliegenden Fall die Geschäftslogik, deren Datenhaltung und Präsentation. Das zu entwickelnde Softwarearchitekturmodell setzt auf einem relationalen Datenbankserver und einem Webserver mit PHP-Interpreter auf. Diese und die darunter liegenden Abstraktionsebenen (Betriebssystem, Hardware, Netzwerk) sind nicht weiter zu betrachten. Da die Präsentation von einem Webbrowser durchgeführt wird und der Sachverhalt damit ausreichend beschrieben ist, wird das Clientsystem aus der Betrachtung ebenfalls ausgeklammert. Weiterhin von Bedeutung sind drei technisch notwendige Komponenten:

- GUI (zuständig für das Benutzerinterface, definiert Aussehen der GUI und generiert HTML-Code)
- Datenbank (liest/schreibt Objektdaten und deren Zuordnungen)
- Fehlerbehandlung (sorgt für eine einheitliche Meldung, Protokollierung und Behandlung von Fehlern in der ganzen Software; enthält eine Prüfroutinenbibliothek)

Eine weitere Gliederung des Systems in Schichten scheint nicht notwendig, da eine potenzielle GUI- oder Datenhaltungsschicht aus lediglich einer Komponente bestehen würde. Damit ist eine Softwarearchitektur erstellt; ein Gesamtüberblick findet sich in Abbildung 17.

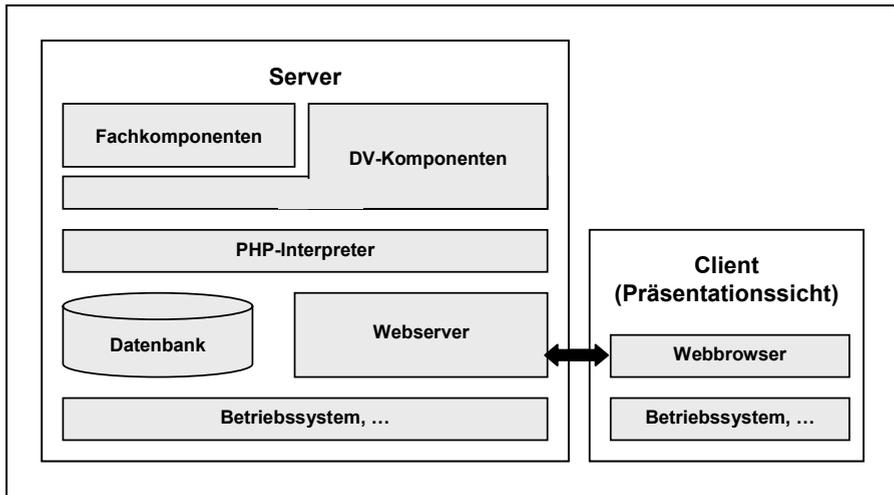


Abbildung 17: Systemarchitektur

4.5 Softwarerealisierung

Dieses Kapitel beschreibt die Entwicklung des definierten Systems. Auf dieser Ebene werden jeweils objektorientierte Analyse und Design vervollständigt sowie in Programmcode umgesetzt (OOP). Nach Abschluss dieser Arbeiten wird ein Release generiert und vom Praxispartner bewertet („Review“). Die Software wird im Folgenden als „VICOPLAN“ (Virtual Corporation Planning System) bezeichnet.

4.5.1 Implementierung

Im Rahmen der Dynamik-Modellierung sind Methoden zu identifizieren. Weiterhin können, abhängig von der diesbezüglichen Komplexität des Systems, Kollaborationsdiagramme, Sequenzdiagramme und Zustandsdiagramme erstellt werden um die Dynamik konkreter zu beschreiben.⁴⁶

Damit ist die Analyse abgeschlossen und der Entwurf (OOD) des Inkrements kann beginnen. Auf die Entwurfsgrundlagen aus Kapitel 3.4 aufbauend, werden die fachlichen Komponenten mit den benötigten Teilen der technischen Komponenten entworfen.⁴⁷ Zudem sind einige Konstrukte der objektorientierten Analyse aufzulösen,⁴⁸ um in der Zielprogrammiersprache realisierbar zu sein. Ebenso ist der Einsatz von Entwurfsmustern zu prüfen. „Ein Entwurfsmuster

⁴⁶ Vgl. Zuser / Biff / Grechenig / Köhle (2002), S. 106.

⁴⁷ Vgl. Zuser / Biff / Grechenig / Köhle (2002), S. 121.

⁴⁸ Zum Beispiel sind Aggregationen in entsprechende Attribute und Methoden umzusetzen.

(design pattern) gibt eine bewährte generische Lösung für ein häufig wiederkehrendes Entwurfsproblem an, das in bestimmten Situationen auftritt.“⁴⁹ Auf diese Weise können gesamte Lösungsideen wieder verwendet werden. Im vorliegenden Fall mussten keine weiteren Wechsel in OOA oder OOD erfolgen.

Damit ist die Analyse des Systems abgeschlossen. Es wird bewusst auf die Typisierung von Attributen (z.B. String, Integer) verzichtet, da die Zielsprache PHP dies nicht unterstützt.

Der Entwurf dieses Abschnitts lässt sich in drei Bereiche gliedern: Entwurf der technischen Basis (Fehlerbehandlung, Datenbank, GUI), von Steuerung und Fachschichtkopplung (Schnittstelle zwischen Mikro- und Makrosteuerung) sowie der ausgewählten Fachklassen.

Aufgrund der Komplexität der Steuerung von Webapplikationen (hier: Model-View-Controller-Muster) wird dieser Bereich ausführlicher betrachtet.

4.5.2 Technische Basis

Der Entwurf der Datenbankkomponente ist zweigeteilt; es sind Klassen zum Datenbankzugriff und das Datenmodell zu bearbeiten. Dabei wird das Datenmodell mit der Umsetzung der Fachklassen schrittweise erstellt und aus der jeweils zugrunde liegenden Klassenattributstruktur hergeleitet. Die technische Datenbankbindung wird in einer datenbanksystemspezifischen Klasse gekapselt (z.B. Klasse „MySQL“ für die MySQL-Datenbank), damit die darunter liegende Datenbanktechnik für die Fachschicht transparent ist. Die Konfiguration des Datenbankzugangs wird in ein Konfigurationspaket ausgelagert und kann über die Schnittstelle „DatenbankKonfiguration“ abgefragt werden. Der Zugriff auf die Datenbank erfolgt, über die Schnittstelle „Datenbankzugriff“ die von der Datenbankbindung realisiert wird. Das entsprechende Modell findet sich in Abbildung 18.

⁴⁹ Balzert (2000), S. 844.

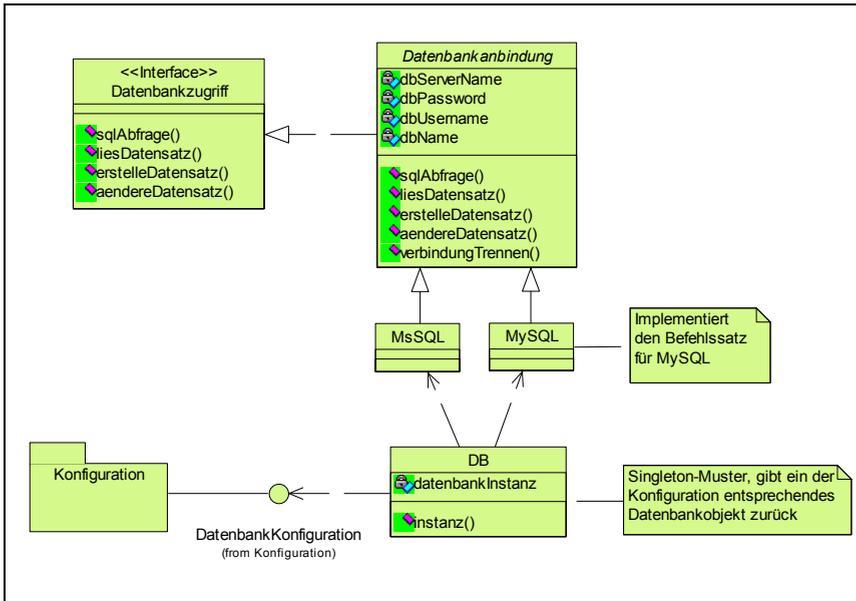


Abbildung 18: Entwurf der Datenbankkomponente

Beim Entwurf der GUI wird analog zur Datenbankkonzeption die Klassenstruktur mit den Fachklassen entworfen. Vor der eigentlichen Entwurfsfähigkeit sind zunächst einige grundsätzliche Überlegungen anzustellen.

Da es sich bei der zu erstellenden Software um einen Prototyp handelt, ist die Entwicklung eines vollständig neuen GUI-Designs zu aufwändig, weshalb sich die Oberfläche von VICOPLAN an einem bestehenden Konzept anlehnen sollte. Aufgrund der Vertrautheit aller potentiellen Benutzer mit der Oberfläche von Microsoft Windows sollen deren grundsätzliche Konzepte adaptiert werden.

Damit stellt sich die Frage, wie entsprechende Schaltflächen, Textfelder etc. zunächst in einen Objektentwurf (zur einfachen Maskendefinition) und im nächsten Schritt in HTML (zur Darstellung im Client-Browser) umzusetzen sind. Während auf der HTML-Ebene Formulare, Layer und Cascading-Stylesheets definiert werden müssen,⁵⁰ soll auf logischer Ebene mit Fenster-Objekten und darin enthaltenen Bedienelementen gearbeitet werden können.

Um dies zu Verknüpfen wird als zentrale Klasse *Fenster* gewählt, welches sich aus verschiedenen Bedienelementen (Spezialisierungen von *FensterInhaltsElement*) zusammensetzt. Das vollständig definierte Fenster ist schließlich in eine Seite einzubetten, welche wiederum in eine Klasse gekapselt ist, um die Generierung des HTML-Codes einfacher zu machen.

⁵⁰ Vgl. o.A. (2003b).

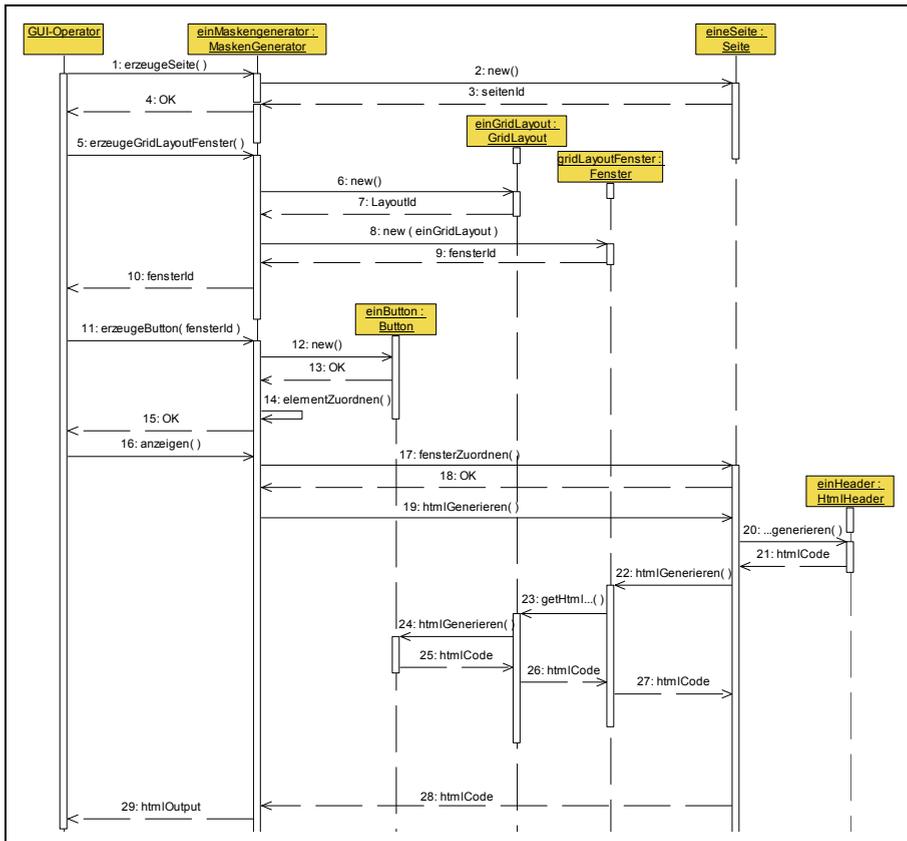


Abbildung 20: Entwurf GUI – Dynamik (Beispiel)

Steuerung und Fachschichtkopplung

Als Problematisch erweist sich bei einem solchen Modell allerdings der Webapplikationscharakter. Intuitiv würde man die Programmsteuerung drei Schritte iterierend ausführen lassen: bestimmte Maske anzeigen, auf Eingaben und „OK-Klick“ warten, Daten verarbeiten. Da die Verbindung zwischen Client und Server aber nach dem Anzeigen der Maske zunächst getrennt wird, ist das Skript, ein Warten auf den Wiederaufbau der Verbindung ist technisch nicht möglich. In einer ersten Näherung sei die Lösung als Zirkelaufruf eines Skripts mit vertauschter Reihenfolge der genannten Schritte beschrieben. Verarbeitet das Skript zunächst alle übergebenen Daten und präsentiert daraufhin eine Maske mit dem Verweis, die dort eingegebenen Daten wieder an dasselbe Skript zurückzusenden ist ein solcher Zirkelaufruf implementiert. Im Rahmen der Fachschichtkopplung ist dieser Mechanismus näher zu betrachten und detailliert zu entwerfen.

Ausgangsposition soll das eben erläuterte Skript sein. Zunächst ist die in der Analyse betrachtete Makrosteuerung zu implementieren. Auf der Ebene der

Arbeitsschritte steht schließlich fest, welche Daten zunächst zu verarbeiten und danach abzufragen sind. An dieser Stelle findet der Übergang zur Mikrosteuerung statt.

Datenverarbeitung und Maskengenerierung fallen in den Verantwortungsbereich des entsprechenden Arbeitsschritt-Objekts. Durch eine Vererbungsstruktur kann für jeden Arbeitsschritt eine Klasse definiert werden, die die Charakteristika eines „generischen Arbeitsschritts“ aufweist (erbt), aber zwei eigenständige Methoden *maskeGenerieren()* und *daten-Verarbeiten()* implementiert, die auf die GUI-Schnittstellen „Masken“ und „Daten“ zugreifen.

Die Prozesskomponente stellt die Controller-Objekte, die GUI-Komponente realisiert den „View“ und die noch nicht näher betrachteten Fachklassen können weitestgehend unabhängig arbeiten und werden von den jeweiligen Arbeitsschritten aufgerufen.

Fachklassen

Wie bereits erwähnt, gehen mit der Entwicklung der Fachklassen die Gestaltung der GUI und die Erstellung des Datenmodells einher. Auf eine eingehende Analyse der Benutzeroberfläche kann aufgrund der Orientierung an der Windows-Oberfläche verzichtet werden. Sämtliche im System benötigten Fenster sind an den Aufbau der aus Microsoft Windows und Microsoft Office bekannten Assistenten (engl. „Wizards“) angelehnt.

Aus Datensicht müssen folgende Daten langfristig im System hinterlegbar sein (Vgl. Tabelle 6):

Funktion	Beschreibung
D10 Anfragen / Aufträge	Bezeichnung, Anfragetext, Akquisitionsdatum, Dateianhänge, Ausführungsort, Angebotsfrist
D15 Arbeitspakete	Bezeichnung, ausführliche Beschreibung, Gültigkeitszeitraum für Ausschreibung, benötigte Kompetenzen
D20 Partnerangebote	Partner, Positionen mit Grobbeschreibung, Detailbeschreibung, Positionsnr., Menge, Einzelpreis, Kommentar, Dateianhänge
D30 Kompetenzdaten	Kompetenzbezeichnung
D40 Koordinatoren- und Partnerdaten	Firma, Kontaktdaten, Kompetenzzuordnungen, Vorliegen eines Partnervertrages, Bankverbindung
D50 Kundendaten	Name, Firma, Kontaktdaten

D60 Aktivitäten und zugehörige Dokumentationseinträge	Zugehörigkeit zu Arbeitspaket, Beginn- datum, Abschlussdatum, Ereignisse (Zahl- ungen, Leistungsnachweise, Bewertungen der Beteiligten)
---	--

Tabelle 6: Langfristig zu speichernde Daten

Aus den bisherigen Ausführungen und der OOA lässt sich für die Datenbank folgendes Relationenschema (Auszug) herleiten (siehe Tabelle 7):⁵¹

Relation	Felder
Attachment	<u>attachmentnr</u> , filename, data
Attachment2Package	<u>attachmentnr</u> , <u>packagenr</u>
Package	<u>packagenr</u> , title, description, date, state, expires, signature
Package2Competency	<u>packagenr</u> , <u>competencnr</u>
Competency	<u>competencnr</u> , name
Competency2User	<u>competencnr</u> , <u>usern</u>
User	<u>usern</u> , name, contactdata
User2Package	<u>usern</u> , <u>packagenr</u>

Tabelle 7: Relationenschema

Im Anhang ist das zugehörige vollständige Datenbankmodell im Querformat abgebildet.

4.5.3 Evaluation

Die Evaluation der erlangten Ergebnisse ist aus verschiedenen Perspektiven möglich.

Zunächst soll eine Ergebnisbewertung in Form einer einfachen Zielerreichungsbetrachtung angestellt werden. Dies entspricht der Perspektive des Praxispartners und wird durch die Resonanz der potentiellen VICOPLAN-Anwender ergänzt. Eine zweite Perspektive eröffnet sich durch die aus der Literatur erhobenen Anforderungen in Kapitel vier.

Musskriterium	Umsetzungsgrad
<u>Kundenangebot im System erfassen</u> Der Kunde soll über eine Maske (z.B. im Internetauftritt des Unternehmensnetzwerks integriert) eine Anfrage in das System einstellen.	vollständig umgesetzt

⁵¹ Primärschlüssel und Fremdschlüssel sind unterstrichen.

<p><u>Kunden- und Anfragedaten anzeigen</u> In einem Informationsmodul sollen sämtliche Anfrage-, Auftrags- und Kundendaten in verschiedenen Sichten (Liste, Detailfenster) abrufbar sein.</p>	vollständig umgesetzt
<p><u>Datei Up- und Download</u> Dateien sollen mit Kundenanfragen, Aufträgen und Partnerangeboten assoziiert werden können.</p>	vereinfacht umgesetzt, Dateiattachments werden mit Prozess verknüpft.
<p><u>Definition von Arbeitspaketen</u> Zu einer Kundenanfrage sollen Arbeitspakete definiert werden können, die bestimmte Kompetenzen erfordern.</p>	vollständig umgesetzt
<p><u>Ausschreibungsfunktionalität</u> Das System soll anhand der benötigten Kompetenzen für jedes Arbeitspaket die in Frage kommenden Partner ermitteln und diese von der Ausschreibung benachrichtigen.</p>	vollständig umgesetzt, Benachrichtigung derzeit per Email
<p><u>Internes Angebot erstellen</u> Während der Laufzeit einer Ausschreibung soll es den beteiligten Partnern möglich sein, im System in standardisierter Form ihr Angebot zu erfassen. Sie sollen einzelne Angebotspositionen mit Mengen und Einzelpreisen in tabellarischer Form eingeben können.</p>	grundsätzlich umgesetzt, in einer Pilotphase müsste die Benutzbarkeit evaluiert und optimiert werden. Ein Problem stellt die Verknüpfung mit bestehenden Angebotssystemen dar. Evtl. sind für den produktiven Einsatz Schnittstellen zu operativen Systemen zu entwerfen.
<p><u>Kundenangebot erstellen</u> Der Koordinator soll aus den erfassten Partnerangeboten ein Kundenangebot zusammenstellen können. Hierfür sollen in einer Übersichtsmaske die einzelnen Anbieter gelistet werden, woraus der Koordinator dann ein Angebot in das Kundenangebot eingefügt werden kann. Zusätzlich soll es möglich sein, für die einzelnen internen Angebote einen prozentualen Koordinationsaufschlag festzulegen.</p>	vollständig umgesetzt.
<p><u>Auftragsvergabe</u> Der Kunde kann das Angebot online einsehen und den Auftrag bestätigen. In diesem Fall soll zusätzlich ein Auftragsbestätigungsformular übermittelt werden.</p>	vollständig umgesetzt.

Tabelle 8: Umsetzung der Musskriterien-Funktionalität

Die in Kapitel 3.4 aufgestellten, aus der Literatur abgeleiteten Anforderungen an die Werkzeugunterstützung, wurden in die vier aus der Literatur abgeleiteten Phasen Informationsphase, Vereinbarungsphase, Durchführungsphase und Auslösungsphase unterteilt. Diese Phasen werden grundsätzlich vom Funktionsumfang von den in VICOPLAN implementierten Funktionalitäten abgedeckt.

Die Anforderungen waren im Einzelnen:

- **Kompetenz- und Kapazitätskatalog**
In VICOPLAN wurde lediglich ein Kompetenzkatalog realisiert, da sich die Pflege aller Kapazitätsdaten der Partnerunternehmen als zu zeitaufwändig erwies. Möglicherweise macht ein Kapazitätskatalog jedoch in einem produktionsorientierten Netzwerk Sinn. Im Rahmen des Projektes waren jedoch dienstleistungsorientierte virtuelle Unternehmen zu unterstützen.
- **Partnerprofile**
Diese Funktion spiegelt sich in den Systemfunktionen wider. Zu jedem Partner sind Kompetenzen und Referenzprojekte hinterlegbar. Auch die Qualitätsbewertung ist umgesetzt.
- **Auktionen und elektronische Verträge**
Durch die internen Ausschreibungen werden Auktionen eingesetzt. Eine elektronische Vertragsverhandlung ist nicht realisiert. Durch die Attachmentfunktion können jedoch Verträge hinterlegt werden.
- **Projektmanagement**
In VICOPLAN ist eine einfache Projektüberwachung mit Vorgänger-/Nachfolgerbeziehungen realisiert. Ein vollständiges Projektmanagement mit Funktionen wie Netzplantechnik hätte den Umfang des Projektes gesprengt.
- **Workflow-/Workgroupssysteme**
Der prozessorientierte Ablauf in VICOPLAN enthält einen fest definierten Workflow mit Push-Technologie (z. B. E-Mail-Benachrichtigungen).
- **Elektronische Rechnungen**
Wurde in VICOPLAN nicht realisiert, da umfangreiche Schnittstellen zu Buchhaltungssoftware zu realisieren wären.
- **Auftragsdokumentation**
Im Rahmen der Auftragsdurchführung ist eine aktivitätsbezogene Dokumentation realisiert.

4.6 Funktionsumfang von VICOPLAN

Nachfolgend wird der Funktionsumfang des Softwaretools VICOPLAN veranschaulicht. Dabei wird einleitend kurz die Grundkonfiguration des Werkzeugs beschrieben. Die sich dann anschließenden Ausführungen werden in die Perspektive des Kunden, des Koordinators und der beteiligten Partnerunternehmen untergliedert und folgen einem beispielhaften Durchlauf eines Auftrags.

4.6.1 Konfiguration des Systems

Nach einem Login als Koordinator können im Menü „Administration“ folgende Einstellungen vorgenommen werden:



Im Punkt Kompetenzen lassen sich die vom System verwendeten Kompetenzen definieren, die in der Ausschreibungsphase zum Matching von Arbeitspaketen und Partnern verwendet werden. Über Kunden- und Partnerdaten lassen sich Nutzerkonten einrichten.

4.6.2 Das Kundeninterface

(potentielle) Neukunden

Ein potentieller Kunde kann sich über eine frei zugängliche WWW-Seite folgendes Formular ausfüllen. Alternativ kann dieses Formular vom Koordinator erfasst werden.

The screenshot displays three main sections of the VICOPLAN web interface:

- Kontaktdaten:** A form with fields for Firma, Branche, Ansprechpartner, Straße, PLZ, Ort, Land (pre-filled with 'Deutschland'), Telefon, Fax, Email, and www.
- Anfragedaten:** A form with fields for Anfragebezeichnung, Ausführungsort, Planungszeitraum (with a date picker), Angebot bis, Rückruf bis (pre-filled with '<kein Rückruf gewünscht>'), Datum/Zeit (pre-filled with '10.04.2003 / 10:07'), Projektart (dropdown menu), Vororttätigkeit in %, Anzahl Personen, and Anzahl Manntage.
- Anfragetext:** A text area for entering the request details, with a note: 'Bitte tragen Sie hier Ihre Anfrage ein. ! Zur detaillierten Beschreibung können Sie eine Datei anhängen.' Below this are three 'Datei #' fields with 'Durchsuchen...' buttons, and 'Zurücksetzen' and 'Absenden' buttons.

Anhand der erfassten Daten wird ein Kundenaccount im System angelegt und ein „Prozess“ mit Status „Kundenanfrage“ gespeichert. Der Kunde kann sich mit der ihm mitgeteilten Kunden Nr. und seiner PLZ als Passwort am System anmelden und den Status seines Prozesses (seiner Anfrage) überwachen.

Bereits bekannte Kunden

Ein Kunde, der bereits über einen Account verfügt kann selbstverständlich diesen nutzen um weitere Anfragen einzugeben.

4.6.3 Koordinatoren-Seite I - Ausschreibung

Der Koordinator (oder Broker) kann eine Übersicht über alle Anfragen bis zum Auftrags-Stadium anzeigen lassen. Dies umfasst Kundenanfragen, interne Ausschreibungen und dem Kunden vorliegende Angebote.

Nr.	Firma	Beschreibung	Status	
NEU 17	Testkunde GmbH	Netzwerkinstallation	in Warteschlange	details, Arbeitspakete, Datei hinzufügen, löschen, Email
18	Neukunde GmbH	Erstellung Fachkonzept	Angebot liegt vor	details, Arbeitspakete, Datei hinzufügen, löschen, Email

Nachdem der Koordinator die Details der Anfrage betrachtet hat, ändert sich der Status.

Nr.	Firma	Beschreibung	Status	
17	Testkunde GmbH	Netzwerkinstallation	in Bearbeitung	details, Arbeitspakete, Datei hinzufügen, löschen, Email
18	Neukunde GmbH	Erstellung Fachkonzept	Angebot liegt vor	details, Arbeitspakete, Datei hinzufügen, löschen, Email

Über den Punkt „Arbeitspakete“ kann der Auftrag nun in Arbeitspakete zerlegt werden, die später netzwerkintern ausgeschrieben werden. Ein Beispiel:

Arbeitspaket hinzufügen

Kurzbeschreibung Anforderungsanalyse

Offen bis 11.04.2003 (TT.MM.JJ)

Inhalt
 Nach einer Ortsbegehung sind sämtliche Anforderungen zu erfassen und ...
 Bei Abgabe eines Angebots zu diesem Arbeitspaket sind Sie 4 Wochen an dieses gebunden.

Datei #1

Datei #2

Bitte für Partner verfügbare Anfragenattachments selektieren:
 Konzept ab.ppt

Jedem Arbeitspaket können wiederum neue Attachments hinzugefügt werden oder Attachments des Kunden für die Adressaten des Pakets freigegeben werden. Das Menü von dem aus operiert wird, sieht folgendermaßen aus:

Arbeitspakete

Arbeitspakete in Bearbeitung
 Netzwerk- und Dienstplanung
 Anforderungsanalyse

ausgeschriebene Arbeitspakete

Nachdem dem Beispielpaket über die Schaltfläche „Kompetenzen“ entsprechende Kompetenzen zugeordnet wurden kann es bereits ausgeschrieben werden. Hierfür gibt das System eine Liste der Partner aus, die mindestens eine der benötigten Kompetenzen besitzen.

Ausschreibung eines Arbeitspakets

Folgende Partner verfügen über passende Kompetenzen:

Partner	Kompetenzen
<input checked="" type="checkbox"/> EDV Services GmbH	EDV - Administration Etwas, was jeder kann
<input checked="" type="checkbox"/> Haustechnik OHG	Etwas, was jeder kann
<input checked="" type="checkbox"/> Kompetenz GmbH	Etwas, was jeder kann
<input checked="" type="checkbox"/> Lutz Köhler GmbH	Etwas, was jeder kann
<input checked="" type="checkbox"/> Management KG	Etwas, was jeder kann
<input checked="" type="checkbox"/> Technopool GmbH	Etwas, was jeder kann
<input type="checkbox"/> WIM	EDV - Administration Etwas, was jeder kann

Nach dem Absenden dieses Formulars werden entsprechende Partner per Email benachrichtigt, dass Sie ein Angebot im System abgeben können. Es ist möglich, Unternehmen manuell von der Ausschreibung auszuschließen, wie hier am Beispiel von WIM gezeigt ist.

4.6.4 Partnerseite

Nachdem sich der Partner (im Beispiel Haustechnik OHG) angemeldet hat, kann er nun eine Liste der Ausschreibungen anfordern, an denen er beteiligt ist:

- Partner-Menü
- Übersicht
- Anfragen
- Ausschreibungen
- Übersicht
- Angebot abgeben
- Administration
- Abmelden

Arbeitspaket	Status	offen bis	
Netzwerk- und Dienstplanung	ausgeschrieben	11.04.03	details, Angebot abgeben
Anforderungsanalyse	ausgeschrieben	11.04.03	details, Angebot abgeben

Über den Menüpunkt Angebote kann der Partner sein Angebot abgeben:

Angebot zu Ausschreibung "Anforderungsanalyse" erfassen (1/2)
Datum: 10.04.2003

Pos	Menge	Einheit	Beschreibung	Einzelpreis	Gesamtpreis
1	1	pausch	Anfahrt	50,00 €	50,00 €
[X]					
2	4	Std.	Ortsbegehung	55 €	
[Pos. speichern]					
<input type="button" value="als Entwurf speichern und abbrechen..."/> <input type="button" value="Weiter >"/>					

Der Partner kann noch einen internen Kommentar anhängen und Attachments anhängen. Dann wird sein Angebot im System gespeichert.

4.6.5 Koordinatoren-Seite II - Erstellen des Kundenangebots

Der Koordinator kann über den Punkt „Übersicht“ im Menü „Ausschreibungen“ überwachen, wie viele Angebote bereits eingegangen sind bzw. wie lange die Partner zur Abgabe noch Zeit haben:

Kunde	Anfrage	Arbeitspaket	Status	offen bis	Angebote
Testkunde GmbH	Netzwerkinstallation	Netzwerk- und Dienstplanung	ausgeschrieben	11.04.03	(0/1)
* Testkunde GmbH	Netzwerkinstallation	Anforderungsanalyse	abgeschlossen	11.04.03	(2/2)
* Testkunde GmbH	Netzwerkinstallation	Planung Serverlandschaft	ausgeschrieben	11.04.03	(0/4)
* Testkunde GmbH	Netzwerkinstallation	Planung Clients	ausgeschrieben	11.04.03	(0/7)
Testkunde GmbH	Netzwerkinstallation	Hardwarebeschaffung	ausgeschrieben	11.04.03	(0/2)

** dieser User ist selbst an der Ausschreibung beteiligt*

Er kann ebenso mit Klick auf die Angebotsanzahl eine detailliertere Ansicht aufrufen:

Arbeitspaket	Partner	Datum/Status	Nettobetrag
Planung Serverlandschaft	EDV Services GmbH	10.04.03	1.900,00 €
	Technopool GmbH	kein Angebot	
	wi2	kein Angebot	
	WIM	10.04.03	1.600,00 €

Über einen Klick auf den Namen des Partnerunternehmens kann das jeweilige Angebot angezeigt werden.

Wenn sämtliche Angebote eingegangen sind, ist es möglich ein Kundenangebot aus den einzelnen Partnerangeboten zusammenzusetzen. Dazu steht folgende Kalkulationsmaske im Menü Kundenangebot zur Verfügung:

Angebot "Netzwerkinstallation"

Sehr geehrte Damen und Herren,

PosNr	Bezeichnung	Preis	Aufschlag	Nettobetrag
1	Anforderungsanalyse		10 %	2.090,00 €
	<input checked="" type="radio"/> WIM <input type="radio"/> EDV Services GmbH	1.900,00 € 1.270,00 €		
2	Netzwerk- und Dienstplanung		10 %	2.640,00 €
	<input checked="" type="radio"/> EDV Services GmbH	2.400,00 €		
3	Planung Serverlandschaft		10 %	1.760,00 €
	<input checked="" type="radio"/> WIM <input type="radio"/> EDV Services GmbH <input type="radio"/> Technopool GmbH <input type="radio"/> wi2	1.600,00 € 1.900,00 € 2.400,00 € 0,00 €		
4	Hardwarebeschaffung		10 %	379.500,00 €
	<input checked="" type="radio"/> EDV Services GmbH <input type="radio"/> Management KG	345.000,00 € 412.000,00 €		
Summe netto:				385.990,00 €
zzgl. 16% MwSt.:				61.758,40 €
Bruttosumme:				447.748,40 €

Mit freundlichen Grüßen

Lehrstuhl für Wirtschaftsinformatik

Der Koordinator kann hier über Mausklicks auf die Firmennamen die Angebote einsehen, sich schließlich für ein Angebot entscheiden, seinen Koordinationsaufschlag festlegen und eine Angebotsvorschau im PDF-Format generieren. Über die Schaltfläche „Kalkulieren“ wird der jeweilige Koordinationsaufschlag mit eingerechnet und die Summe aktualisiert. Das generierte PDF-Dokument führt auch die vom Partner eingegebenen Positionen als „Unterpositionen“ auf.

Über den Punkt „Angebot erstellen“ wird ein digital signiertes Dokument im System angelegt, welches dem Kunden zugesandt oder von ihm online abgerufen werden kann.

4.6.6 Kundenseite

Liegt zu einer Kundenanfrage ein Angebot vor, kann der Kunde dieses online betrachten und den Auftrag erteilen. Dies geschieht zum einen über eine Dialogmaske im System, zum anderen wird dem Kunden ein Formular übertragen, mit der Bitte den Auftrag noch schriftlich zu bestätigen.

Kunde	Anfrage	Datum	Status	Signatur	
Testkunde GmbH	Netzwerkinstallation	10.04.03	Angebot liegt vor	not signed yet	anzeigen , beauftragen , Rückfrage

Auftrag vergeben

Sie sind im Begriff, folgendes Angebot von "Lehrstuhl für Wirtschaftsinformatik" zu bestätigen und den Auftrag zu vergeben:

Netzwerkinstallation, Angebot vom 10.04.03, Angebotsnr. 12
 Signatur: not signed yet

Auftrag vergeben

Sie sind im Begriff, folgendes Angebot von "Lehrstuhl für Wirtschaftsinformatik" zu bestätigen und den Auftrag zu vergeben:

Netzwerkinstallation, Angebot vom 10.04.03, Angebotsnr. 12
 Signatur: not signed yet

4.6.7 Koordinatoren-Seite III - Auftragsplanung

Im Menü „Auftrag“, Unterpunkt „Planung“ ist es nun möglich einen vom Kunden vergebenen Auftrag in sog. Aktivitäten zu unterteilen um eine zeitliche Abfolge festzulegen.

Aktivität hinzufügen

Arbeitspaket	Softwareentwicklung für Callcenter	Beginnt nach	Grobkonzept
Soll-Start	14.06.04	Soll-Ende	30.06.04
Aktivitäts-Titel	Feinkonzept		

Arbeitspaket	Aktivität	beginnt nach	Soll-Start	Soll-Ende	Dauer	Puffer	
Softwareentwicklung für Callcenter	Anforderungen aufnehmen ---		01.04.04	01.05.04	30 Tage	k.Vorg.	<input type="button" value="Ändern"/> <input type="button" value="Löschen"/>
Softwareentwicklung für Callcenter	Grobkonzept	Anforderungen aufnehmen	01.05.04	01.06.04	31 Tage	13 Tage	<input type="button" value="Ändern"/> <input type="button" value="Löschen"/>

Nach Abschluss dieser Planung lässt sich der Status des Auftrags über den Menüpunkt „Aufträge“, Unterpunkt „Kontrolle“ überwachen:

Budget-Kontrolle für Prozess 3					
Arbeitspaket/Aktivität	Partner	Ist-Kosten	Soll-Kosten	Verbrauch	
Softwareentwicklung für Callcenter	Projektix OHG	10.000,00 €	85.000,00 €	12%	
- Anforderungen aufnehmen		10.000,00 €			
- Grobkonzept		0,00 €			
- Feinkonzept		0,00 €			
- Entwicklung		0,00 €			
- Software-Einführung		0,00 €			
40 PC-Arbeitsplätze inkl. SW	Computerstore GmbH	2.500,00 €	39.900,00 €	6%	
- Planung für PCs		2.500,00 €			
- PCs für Pilotbetrieb & Test		0,00 €			
- PCs fertigen		0,00 €			
- PCs aufstellen		0,00 €			

Zeitplan-Kontrolle für Prozess 3						
Aktivität	beginnt nach	Partner	Start (Soll/Ist)	Ende (Soll/Ist)	Zeit verstrichen	
Anforderungen aufnehmen	kein Vorgänger	Projektix OHG	01.04.04 / 12.01.04	01.05.04 / -	0%	
Grobkonzept	Anforderungen aufnehmen	Projektix OHG	01.05.04 / -	01.06.04 / -	0%	
Feinkonzept	Grobkonzept	Projektix OHG	14.06.04 / -	30.06.04 / -	0%	
Entwicklung	Feinkonzept	Projektix OHG	01.07.04 / -	01.12.04 / -	0%	
Planung für PCs	Feinkonzept	Computerstore GmbH	01.07.04 / -	15.07.04 / -	0%	
PCs für Pilotbetrieb & Test	Planung für PCs	Computerstore GmbH	01.08.04 / -	15.08.04 / -	0%	
PCs fertigen	PCs für Pilotbetrieb & Test	Computerstore GmbH	01.11.04 / -	20.11.04 / -	0%	
PCs aufstellen	PCs fertigen	Computerstore GmbH	01.12.04 / -	20.12.04 / -	0%	
Software-Einführung	Entwicklung	Projektix OHG	14.12.04 / -	31.12.04 / -	0%	

Gesamt-Zeitverbrauch (Soll):

Auszahlungen an Partner (bereits gezahlt: 12.500,00 €, zu zahlen: 252.700,00 €) :

Zahlungen vom Kunden (bereits gezahlt: 15.000,00 €, zu zahlen: 290.605,00 €) :

Über den Unterpunkt „Dokumentation“ können derzeit folgende Einträge in der Auftragsdokumentation erstellt werden:

- Beginn und Abschluss von Aktivitäten (von Partnern)
- Auszahlungen an Partner
- Zahlungen des Kunden
- Bewertungen von Partner-Unternehmen (durch Kunde und Koordinator)

Die bereits im Menü vorhandenen Leistungsnachweise müssen für jede Firma individuell angepasst werden; hier soll es möglich sein, in einem „digitalen Stundenabrechnungsbogen“ ausgeführte Arbeiten zu dokumentieren.

Ist für alle Aktivitäten der Abschluss dokumentiert, ist der Auftrag abgeschlossen. VICOPLAN unterstützt weiterhin auf auftragsübergreifender Ebene eine Analyse der Partnerbewertungen, der Netzwerkumsätze und der Kompetenzauslastung.

Zusammenfassend besitzt VICOPLAN derzeit folgende Features:

- Benutzerrollen: Broker, Partner, Kunde (Auftrag bereits vergeben), potentieller Kunde (noch kein Auftrag)
- Verwaltung von Kompetenzlisten für Partner und Broker

- Vorgänge werden als Prozess verwaltet, der verschiedene Stufen durchläuft (Anfrage, int. Ausschreibung, Kundenangebot, Auftrag ...)
- Entgegennehmen von Kundenanfragen (via WWW), feldnormierter Teil + Freitext + Upload von Attachments
- Zerlegen von Kundenanfragen in einzelne „Arbeitspakete“ (durch Broker)
- Arbeitspaketen (benötigte) Kompetenzen zuordnen, dann ausschreiben
- mögliche Partner anhand eines Matchings von Kompetenzen identifizieren und per Email von Ausschreibung benachrichtigen
- Partnerangebote via WWW-Formular entgegennehmen
- Kundenangebot aus einzelnen Partnerangeboten zusammenstellen, wobei ein Koordinationsaufschlag eingerechnet werden kann
- Vergabe des Auftrags durch den Kunden
- Zeitliche Ablaufplanung für jeden Auftrag (Zerlegung in Aktivitäten)
- Kontrolle des Projektfortschritts
- Dokumentationssystem
- Grundlegende Analysen und Auswertungen über die im System erfassten Daten

5 Analyse ausgewählter innovativer Technologien

In einem virtuellen Unternehmen kooperieren mehrere Partnerunternehmen mit heterogenen IV-Systemen, um die gemeinsame Durchführung eines Auftrags zu ermöglichen. Daher ist dem Bereich der zwischenbetrieblichen Integration besondere Aufmerksamkeit zu schenken. Für diese Integration ist eine geeignete Netzinfrastruktur zu schaffen und die Kopplung der IV-Systeme zu ermöglichen. Die Aufgabe der Kopplung von IV-Systemen kann dabei in zwei verschiedene Bereiche aufgliedert werden, damit die Interoperabilität gewährleistet wird. Auf der einen Seite muss sich innerhalb des virtuellen Unternehmens auf ein Datenaustauschformat geeinigt werden, auf dem die Nachrichten basieren. Auf der anderen Seite müssen die für diesen Nachrichtenaustausch notwendigen Schnittstellen geschaffen werden. Diese müssen auf standardisierten und offenen Technologien beruhen, um eine hohe Interoperabilität der Kopplung zwischen den heterogenen IV-Systemen der Partnerunternehmen erreichen zu können. Um darüber hinaus die Flexibilität bei den Schnittstellen einer solchen Kopplung zu erhöhen, können Parametrisierungsmechanismen eingesetzt werden, welche die Schnittstellen leicht anpassbar machen.⁵²

Im Bereich der zwischenbetrieblichen Integration lassen sich die beiden Aufgaben der Auswahl eines Datenaustauschformats und der Bereitstellung einer Schnittstelle zur Kopplung der IV-Systeme der Partnerunternehmen identifizieren. Im Bereich der Datenaustauschformate sind wiederum traditionelle Austauschformate (EDIFACT, ANSI X12, etc.) von XML-basierten Austauschformaten zu unterscheiden. Dabei fällt allerdings auf, dass sich traditionelle Datenaustauschformate erst bei „Weltunternehmen“ etabliert haben, die eine Vielzahl von Transaktionen durchführen, die zudem einen zeitkritischen Charakter haben. Die Ursache dafür ist in den Infrastrukturausgaben für die Nutzung dieser Austauschformate zu sehen, deren größte Einzelposition durch die nutzungsunabhängigen Kosten des EDI-Konverters gebildet wird. Dies führte vor allem in kleinen und mittelständischen Unternehmen zu einer sehr schlechten Akzeptanz. Von allen Unternehmen, für die die Nutzung von traditionellen Austauschformaten und die Durchführung von EDI vorteilhaft wären, setzen

⁵² Vgl. Mertens / Griese / Ehrenberg (1998), S. 69ff.; Mertens / Faisst (1996), S. 280ff.

lediglich 5% diese Technologie überhaupt ein.⁵³ Daher werden in diesem Bereich der zwischenbetrieblichen Integration innerhalb des nächsten Kapitels auf der einen Seite Datenaustauschformate auf Basis der Metasprache XML und auch der anderen Seite Web Services als eine Art der Bereitstellung von Schnittstellen fokussiert, durch deren Nutzung vor allem die Umsetzungskosten einer solchen zwischenbetrieblichen Integration gesenkt werden können. Dies kann u. a. durch die freie Verfügbarkeit von XML-Parsern begründet werden. Web Services als Schnittstellentechnologie wurde aufgrund der Offenheit und der Verwendung von etablierten Technologien und Protokollen ausgewählt. Zwar wäre die Realisierung der Schnittstellen auch auf Basis von Java RMI oder Corba möglich gewesen, jedoch ist bei der Nutzung von Java RMI nur die Kommunikation zwischen Objekten bzw. Komponenten realisierbar, die in der Programmiersprache Java umgesetzt wurden. Bei der Verwendung von CORBA wird die Kommunikation über den Object Request Broker (ORB) abgewickelt. Dieser ORB wäre somit zwingender Bestandteil einer solchen Schnittstellenrealisierung. Jedoch treten in der Praxis zahlreiche Kompatibilitätsprobleme bei der Nutzung von ORB's verschiedener Anbieter auf, so dass auch diese Möglichkeit nicht in Betracht gezogen wurde.⁵⁴

Im Bereich des Managements des virtuellen Unternehmens stellt sich die Anforderung der möglichst effektiven Aggregation der operativen Datenbestände zu Kennzahlen, die dann die Entscheidungsbasis bilden können. Für diese Aggregation bietet sich das Konzept des Online Analytical Processing (OLAP) an. Daher wird im Rahmen dieser Veröffentlichung analysiert, inwieweit sich OLAP-Systeme für die Unterstützung von virtuellen Unternehmen eignen.

Nachfolgend werden die Technologien XML, Web Services und OLAP in ausführlicher Form diskutiert. Jeder Abschnitt behandelt zuerst die Grundlagen, um daran anschließend die Einsatzmöglichkeiten aufzuzeigen, die sich für die jeweiligen Technologien in virtuellen Unternehmen ergeben. Am Ende jedes Abschnitts wird eine kurze Bewertung der aufgezeigten Einsatzmöglichkeiten vorgenommen.

5.1 Extensible Markup Language (XML)

Nachfolgend wird die Metasprache XML auf ihre Eignung für den Einsatz innerhalb virtueller Unternehmen untersucht. Dazu werden einleitend zuerst einmal die Grundlagen von XML dem Leser vorgestellt. Daran anschließend werden dann mögliche Einsatzszenarien skizziert sowie zum Abschluss eine Bewertung der Eignung von XML für den Einsatz in virtuellen Unternehmen vorgenommen.

⁵³ Vgl. Weitzel / Harder / Buxmann (2001), S. 7ff.

⁵⁴ Vgl. Upton / McAfee (1996), S. 123ff.

5.1.1 Grundlagen

Um eine strukturierte Darstellung der Grundlagen der Metasprache XML zu erreichen, wurde dieser Grundlagenabschnitt folgendermaßen untergliedert. Zu Beginn wird ein kurzer Überblick über die Metasprache XML gegeben. Auf diesem Überblick aufbauend werden sowohl der Aufbau von XML-Dokumenten als auch die verschiedenen Möglichkeiten der Strukturbeschreibungen von XML-Dokumenten vorgestellt. Abschließend wird der mögliche Zugriff auf XML-Dokumente sowie die Verarbeitung von XML-Dokumenten diskutiert.

5.1.1.1 Überblick

„XML ist eine textbasierte Meta-Auszeichnungssprache, die die Beschreibung, den Austausch, die Darstellung und die Manipulation von strukturierten Daten erlaubt.“⁵⁵ Betrachtet man den Begriff einer Meta-Auszeichnungssprache genauer, so ermöglichen Auszeichnungssprachen prinzipiell die Anreicherung von textorientierten Informationen mit Strukturinformationen und Metadaten über so genannte Marken (engl. Tags). Dabei lassen sich unter anderem folgende Auszeichnungsarten unterscheiden:

- *Semantische Auszeichnungen*: Anreicherung der Inhalte mit Informationen über die Bedeutung der Inhalte
- *Strukturelle Auszeichnungen*: Anreicherung der Inhalte mit Informationen über die Position der Inhalte innerhalb eines Dokuments (Überschriften, Absätze, Referenzen, etc.)
- *Stilistische Auszeichnungen*: Anreicherung der Inhalte mit Informationen über die Formatierung bestimmter Passagen innerhalb eines Dokuments (Schriftgröße, Schriftart, etc.)

Somit stellt XML als Meta-Sprache keine Auszeichnungssprache selbst dar, sondern ermöglicht vielmehr die Definition von neuen Auszeichnungssprachen auf Basis von XML. XML kann also in die Klasse der generischen Auszeichnungssprachen eingeordnet werden.⁵⁶

XML wurde durch die XML Working Group, die durch World Wide Web Consortium (W3C) eingerichtet wurde, entwickelt. Die Verabschiedung als Standard in der Version 1.0 erfolgte im Februar 1998. Als Grundlage des Entwicklungsprozesses diente die Standard Generalized Markup Language (SGML), die bereits seit 1986 in vielen Anwendungsgebieten wie im Verlagswesen und in Software-Systemen eingesetzt wird. Die Weiterentwicklung wurde deswegen fokussiert, da die Komplexität von SGML durch die Nutzer als zu hoch eingestuft wurde. Darüber hinaus bildeten die starren Auszeichnungssprachen wie

⁵⁵ Weitzel / Harder / Buxmann (2001), S. 18.

⁵⁶ Vgl. Rawolle (2002), S. 52.

beispielsweise HTML einen Weiterentwicklungsbedarf, da deren Flexibilität durch die Anwender als unzureichend eingeschätzt wird.⁵⁷

Betrachtet man die Entwicklungsziele von XML, so wurde ein Hauptaugenmerk auf die einfache Handhabung und Beherrschbarkeit von XML gelegt. Darüber hinaus wurden eine Kompatibilität mit SGML und eine hohe Flexibilität bzgl. der genutzten technischen Plattformen und Anwendungsgebiete gefordert.⁵⁸

Als Ergebnis dieses Entwicklungsprozesses unter Einhaltung der Entwicklungsziele entstand die Meta-Sprache XML, die ca. 20 % der Komplexität und ca. 80 % der Funktionalität von SGML umfasst.⁵⁹ Jedoch kann XML nicht als ein Nachfolger oder eine Erweiterung der Hypertext Markup Language (HTML) aufgefasst werden, sondern stellt ein Profil von SGML dar. Damit sind alle XML-Dokumente per Konstruktion konforme SGML-Dokumente, so dass XML häufig auch als Teilmenge von SGML bezeichnet wird. Im Gegensatz dazu repräsentiert HTML eine Anwendung von SGML mit einem festgelegten Satz an Marken.

Als Zeichensatz verwendet XML standardmäßig Unicode, sofern keine anderen Angaben innerhalb der Dokumente gemacht werden. Unicode stellt dabei ein plattform-, betriebssystem- und anwendungsübergreifendes Kodierungssystem zur Verfügung, in welchem die Schriftzeichen aller weltweit gesprochenen Sprachen repräsentiert werden. Daher kann XML plattform- und betriebssystem-unabhängig eingesetzt werden.⁶⁰

5.1.1.2 *Aufbau*

Der XML-Standard spezifiziert eine Menge von Bestandteilen, die entweder obligatorischen oder optionalen Charakter haben. Zu den obligatorischen Bestandteilen sind die Elemente mit optionalen Attributen sowie der Prolog zu rechnen, zu den optionalen Bestandteilen eines XML-Dokuments zählen Kommentare und Verarbeitungsanweisungen (eng. Processing Instruction).

Ein XML-Dokument beginnt mit dem obligatorischen Prolog. Dieser Prolog besteht dann mindestens aus der Deklaration der genutzten XML-Version. Darüber hinaus können dort auch Angaben über den verwendeten Zeichensatz innerhalb des XML-Dokuments gemacht werden.

Darüber hinaus besitzt ein XML-Dokument mindestens ein Wurzelement, welches den eigentlichen Inhalt des Dokuments umschließt. Ein Element wird durch einen Bezeichner gekennzeichnet und besteht in der Regel aus einer Start- und einer Endmarke, die den eigentlichen Inhalt (weitere Elemente oder Zeichen-

⁵⁷ Vgl. Genussa (1999), S. 33f.

⁵⁸ Für einen Überblick über die Entwicklungsziele der Meta-Auszeichnungssprache XML vergleiche u. a. Bray / Paoli / Sperberg-McQueen / Maler (2000).

⁵⁹ Vgl. Light (1997), S. 21f.

⁶⁰ Vgl. Holzner (2002), S. 32.

folgen) einbetten. Aus dieser Schachtelung ergibt sich insgesamt eine hierarchische Struktur der Dokumente selbst, die auch als Baumstruktur abgebildet werden kann. Optional lassen sich den Elementen Attribute in Form von Attribut-Wert-Paaren zuordnen, die eine genauere Beschreibung der Elemente ermöglichen. Zusätzlich können XML-Dokumente, wie bereits erwähnt, weitere optionale Bestandteile wie Kommentare und Verarbeitungsanweisungen enthalten, die entweder durch die Zeichenfolge `<!--` und `-->` oder durch `<?>` und `?>` eingeschlossen werden.

Abbildung 21 stellt ein Beispiel mit den eben erläuterten Elementen dar.

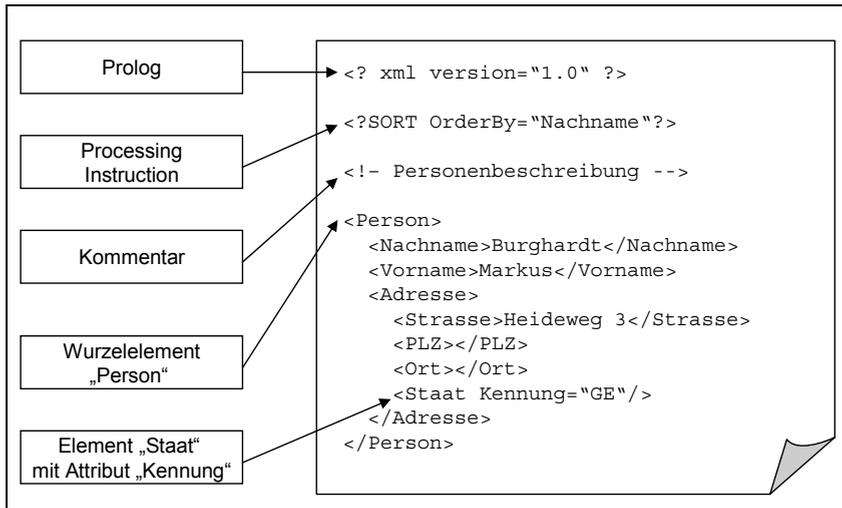


Abbildung 21: Beispiel eines XML-Dokuments

XML-Dokumente, die diese Syntaxregeln einhalten, werden durch die XML-Spezifikation als wohlgeformte XML-Dokumente bezeichnet. Folgt das Dokument darüber hinaus den Strukturbeschreibungen einer zugeordneten Document Type Definition bzw. einem speziellen Schema, wird das XML-Dokument als gültig bezeichnet. Aus dieser Darstellung wird deutlich, dass eine gültiges XML-Dokument immer ein wohlgeformtes XML-Dokument darstellt, die Umkehrung hingegen ist nicht immer gegeben. Der Bezug auf eine Strukturbeschreibung (Document Type Definition / Schema) wird durch einen Verweis im XML-Dokument kenntlich gemacht.

Auch sieht die XML-Spezifikation die Aufteilung der XML-Dokumente auf mehrere Einheiten (engl. Entities) vor, die in verschiedenen physischen Dateien abgelegt werden können. Dadurch wird u. a. eine gewisse Redundanzfreiheit erzeugt, da die Einheiten in verschiedene XML-Dokumente eingebunden werden können.

5.1.1.3 Strukturbeschreibung

Um die Struktur von XML-Dokumenten zu beschreiben und somit gültige XML-Dokumente zu erstellen, sieht die XML-Spezifikation die beiden Mechanismen der Document Type Definition (DTD) und der XML Schema vor. Prinzipiell werden durch die Strukturbeschreibung die zulässigen Elemente mit den dazugehörigen Attributen und die hierarchische Anordnung der Elemente innerhalb eines XML-Dokuments beschrieben. Die Strukturbeschreibung selbst kann dabei entweder als interne Strukturdefinition ein Bestandteil des XML-Dokuments sein oder aber als externe Strukturdefinition in einer separaten Datei vorliegen. XML-Dokumente, die sich auf eine Strukturbeschreibung beziehen, werden als Instanzen dieser Strukturdefinition bezeichnet.⁶¹ Um auch Instanzen basierend auf mehreren Strukturdefinitionen zu ermöglichen, wurden XML Namensräume (XML Namespaces) eingeführt. Die beiden Möglichkeiten einer Strukturbeschreibung und die Verwendung von Namensräumen werden nachfolgend dargestellt.

Abbildung 22 zeigt eine DTD. Dabei wird das in Abbildung 24 aufgezeigte XML-Dokument durch diese Strukturdefinition beschrieben. Als Schlüsselwort zur Deklaration von Elementen dient ELEMENT, die Deklaration von Attributen geschieht über das Schlüsselwort ATTLIST. Mit dem Schlüsselwort #PCDATA (Parsed Character Data) werden Elementinhalte als Text deklariert und das Schlüsselwort EMPTY zeichnet leere Elemente aus. Man erkennt beispielsweise an der Strukturdefinition, dass das Element Person die Elemente Nachname, Vorname und Adresse enthält und kann daraus auf die hierarchische Struktur schließen.

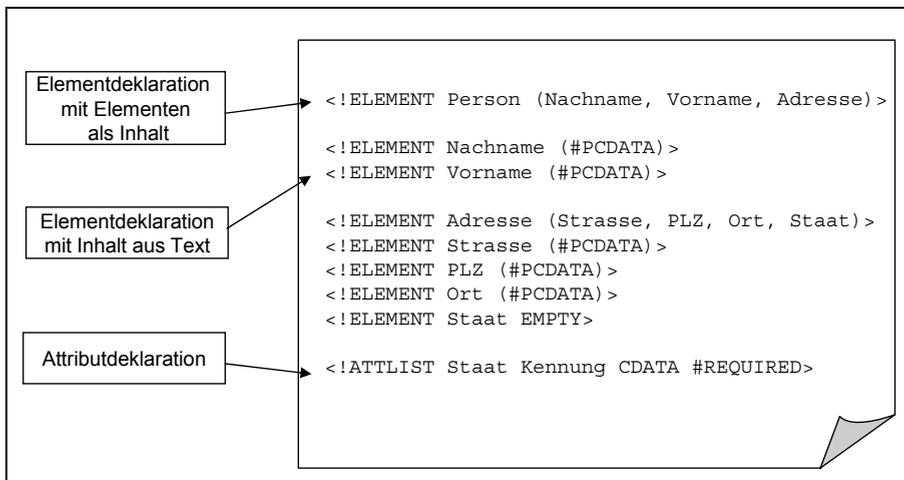


Abbildung 22: Beispiel einer Document Type Definition (DTD)

⁶¹ Vgl. Bray / Paoli / Sperberg-McQueen / Maler (2000).

Auf die gesamten Möglichkeiten der Strukturbeschreibung soll jedoch nicht weiter eingegangen werden, da die Document Type Definition weitere umfangreiche Strukturbeschreibungsmöglichkeiten wie beispielsweise die Definition von optionalen Elementen und Wiederholungsgruppen ermöglicht.

Ein alternativer Spezifikationsmechanismus für die Strukturbeschreibung liegt mit der XML Schema Spezifikation vor, die auf der einen Seite den Beschreibungsumfang von Document Type Definitions erweitert und auf der anderen Seite einige Nachteile der Document Type Definitions aufhebt.

Folgende wichtige Vorteile sind der XML Schema Spezifikation gegenüber Document Type Definitions anzuführen:

- Schemata sind selbst XML-Dokumente und lassen sich daher mit den gleichen Werkzeugen wie XML-Dokumente bearbeiten.
- Schemata unterstützen Datentypen, die sowohl auf Elementebene als auch auf Attributebene greifen und die möglichen Ausprägungen der Inhalte einschränken.
- Die Kardinalität von Elementen innerhalb von Instanzdokumenten kann anhand von Minimal- und Maximalanzahl genau spezifiziert werden.
- Schemata bieten in Anlehnung an die Objektorientierung sowohl Erweiterungs- (u. a. in Bezug auf die möglichen Datentypen) als auch Vererbungsmechanismen.

Abbildung 23 zeigt ein Schema für das Beispieldokument aus Abbildung 21. Man erkennt deutlich, dass es sich um ein wohlgeformtes XML-Dokument handelt und die Ausprägungen der Inhalte der Elemente explizit durch die Verwendung von Datentypen festgelegt werden können. Auch an dieser Stelle sei aufgrund der Mächtigkeit der XML Schema Spezifikation auf die entsprechenden Quellen verwiesen.⁶²

⁶² Vgl. Thompson (2002).

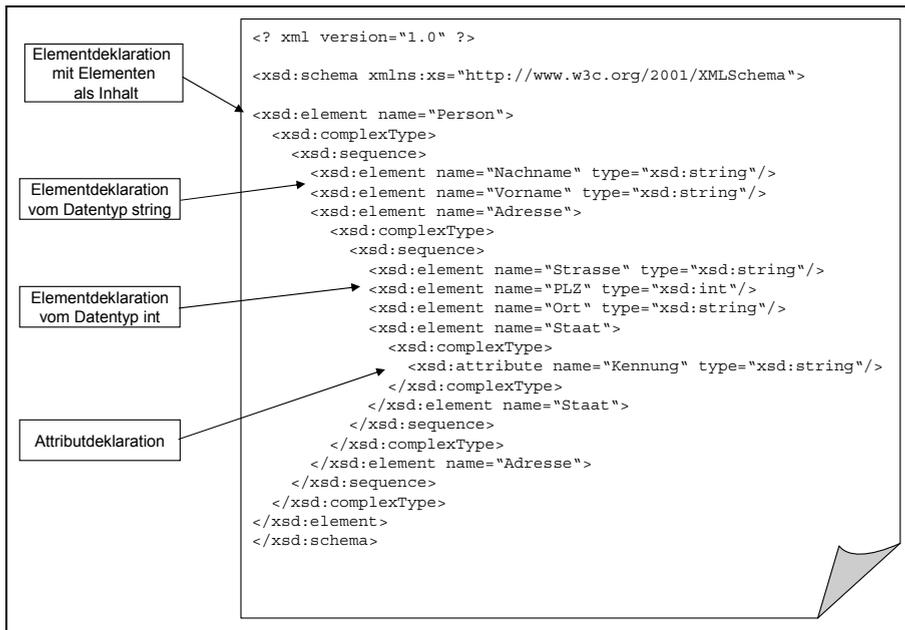


Abbildung 23: Beispiel eines XML Schema

Jedoch sieht die XML Spezifikation prinzipiell vor, dass nur Instanzen auf Basis genau einer Strukturdefinition möglich sind. Dieses Konstrukt stellt sich allerdings als zu unflexibel heraus. Daher wurde durch Namensräume die Möglichkeit geschaffen, mehrere verschiedene Strukturdefinitionen zu nutzen, um daraus XML-Dokumente zu erstellen, die prinzipiell Instanzdokumente aller genutzten Strukturdefinitionen sind, um die Gefahr von Namenskollisionen zu vermeiden. Namenskollisionen entstehen, wenn in mehreren Grammatiken Elemente mit derselben Bezeichnung auftreten und dann keiner Grammatik eineindeutig zugeordnet werden können. Namensräume ordnen somit Elemente mit den dazugehörigen optionalen Attributen über eine Uniform Resource Identifier (URI) einer eindeutigen Strukturdefinition zu. In Abbildung 25 wurden bereits Namensräume verwendet. Die verwendeten Elemente wurden dabei eineindeutig dem Namensraum xsd zugeordnet, der mit der entsprechenden XML Schema Strukturdefinition verknüpft ist.

5.1.1.4 Zugriff

Der Zugriff auf XML-Dokumente durch die nutzenden Anwendungen erfolgt in der Regel über so genannte XML-Parser oder XML-Prozessoren. Dabei handelt es sich um Softwaremodule, die XML-Dokumente einlesen und einen Zugang auf die Strukturen und Inhalte von XML-Dokumenten gestatten. Die Zugriffsmöglichkeiten für die Verarbeitung und die Analyse von XML-Dokumenten werden dabei über standardisierte Application Programming

Interfaces (APIs) der nutzenden Anwendung zur Verfügung gestellt.⁶³ Es lassen sich prinzipiell mit dem Document Object Model (DOM) und dem Simple API for XML (SAX) zwei Arten von APIs unterscheiden, die im Folgenden kurz skizziert werden sollen. Konkrete Umsetzungen der vorgestellten APIs und somit XML-Parser sind für alle wichtigen Programmiersprachen verfügbar.

XML-Parser, die die API von DOM realisieren, repräsentieren XML-Dokumente durch Baumstrukturen, die die hierarchischen Strukturen von XML-Dokumenten abbilden. Dazu werden in dieser Baumstruktur Element-, Attribut- und Inhaltsknoten unterschieden, denen jeweils Zugriffsmethoden u. a. zum Lesen, Suchen, Löschen und Schreiben der Knoteninhalte zugeordnet wurden. Jedoch wird vor dem Zugriff das komplette XML-Dokument analysiert und in eine statische Repräsentation in Form eines Baumes überführt. Dies führt gerade bei umfangreichen XML-Dokumenten zu einem hohen Speicherbedarf und einer gewissen Verzögerung, bis das XML-Dokument für den Zugriff bereitsteht.

Um diese Nachteile zu umgehen, hat sich mit SAX eine zweite Variante für den Zugriff in der Praxis etabliert. SAX zählt zu den ereignisbasierten APIs, die bei jedem Auftreten eines XML-Konstrukts (Start-Marke, End-Marke, Elementinhalt, etc.) während des Analysierens des XML-Dokuments ein Ereignis auslösen, welches die nutzende Anwendung abfangen und weiterverarbeiten muss. Jedoch hat die auf ereignisbasierten Konzepten resultierende Effizienz von SAX negative Auswirkungen auf die Benutzerfreundlichkeit der zugrunde liegenden API.

5.1.1.5 *Verarbeitung*

Neben der Verarbeitung von XML-Dokumenten über die Nutzung von XML-Parsern haben sich deklarative Anfrage- und Transformationssprachen auf der einen Seite und Werkzeuge zur Unterstützung der Arbeit mit XML-Dokumenten auf der anderen Seite etabliert. Diese vereinfachen die Behandlung von XML-Dokumenten und setzen auf XML-Prozessoren auf.

Anfrage und Transformationssprachen unterstützen sowohl die Gewinnung von Informationen aus Datenbeständen als auch die Konvertierung von Datenbeständen aus einer Struktur in eine andere Struktur. Durch das World Wide Web Consortium (W3C) wird die Extensible Stylesheet Language (XSL) entwickelt, die sowohl zur Formatierung als auch zur Transformation von XML-Dokumenten eingesetzt werden kann.⁶⁴ Bei der Spezifikation von XSL wurden die grundlegenden Konzepte der seit 1996 der etablierten Document Style Semantics and Specification Language (DSSSL)⁶⁵ übernommen, da diese bereits erfolgreich

⁶³ Vgl. Goldfarb / Prescod (2002), S. 50f.

⁶⁴ Vgl. Harold (2001), S. 433ff.

⁶⁵ Die DSSSL Spezifikation besteht aus einer Transformations- und einer Stilsprache. Die Transformationsprache wandelt SGML-Dokumente, die Instanzen in einer bestimmten Strukturdefinition sind, in Instanzdokumente einer beliebigen anderen Strukturdefinition um. Die Stilsprache legt das Layout einzelner Bestandteile des SGML-Dokuments fest, wobei

im Umfeld von SGML im Einsatz war. Die Umsetzung jedoch erfolgte in einer syntaktisch anderen Form.⁶⁶ Die Spezifikation von XSL besteht mit XML Path Language (XPath), XSL Transformations (XSLT) und XSL Formatting Objects (XSL-FO) aus drei aufeinander aufbauenden Spezifikationen.⁶⁷ Diese sollen nachfolgend kurz skizziert werden.

XPath wurde im November 1999 als Standard für eine Anfragesprache verabschiedet und dient der Adressierung von Teilen eines XML-Dokuments. So können nicht nur XML-Dokumente als Ganzes über Links angesprochen werden, sondern auch einzelne Abschnitte eines XML-Dokuments oder sogar konkrete Elemente.⁶⁸ Somit werden durch XPath Wegbeschreibungen in Form von Pfadangaben durch XML-Dokumente ermöglicht, die sich syntaktisch an die Notation von URIs halten und somit eine Navigation innerhalb der XML-Dokumente ermöglichen.⁶⁹ Jedoch erwies sich XPath aufgrund der eingeschränkten Funktionalität als nicht ausreichend und es wurde mit der Entwicklung von XQuery begonnen, die bis heute (Stand: Mai 2004) noch nicht abgeschlossen ist.⁷⁰

XSLT wurde ebenfalls im November 1999 als Standard vom W3C verabschiedet. Die Spezifikation umfasst Vorschriften zur Transformation eines XML-Quelldokuments in ein beliebiges Zieldokument, wobei es sich beim Quelldokument um ein wohlgeformtes XML-Dokument handeln muss.⁷¹ Dabei werden die konkreten Umwandlungsvorschriften in einem so genannten Stylesheet hinterlegt. Die Durchführung einer Transformation wird dann von einem XSLT-Prozessor übernommen, der das Quelldokument und das Stylesheet einliest und dann das Stylesheet auf das Quelldokument anwendet. XSLT-Prozessoren unterscheiden sich also grundlegend von XML-Prozessoren, da XSLT-Prozessoren XML-Dokumente transformieren, XML-Prozessoren jedoch lediglich die Struktur und den Inhalt von XML-Dokumenten für andere Anwendungen zur Verfügung stellen.

XSL-FO wurde im Oktober 2001 als Standard vom W3C verabschiedet und beinhaltet Mechanismen zur Formatierung von XML-Dokumenten. Dazu werden XML-Quelldokumente mit so genannten Formatting Objects (FOs) ausgestattet, in dem ein XSL-FO-Stylesheet auf die Quelldokumente angewendet wird.⁷² XSL-FO teilt die XML-Dokumente in verschiedene Bereiche auf. Dabei ist eine Überlappungen oder eine Schachtelung von verschiedenen Abschnitten möglich. Die einzelnen Teile werden dann mittels Formatting Objects formal beschrieben,

Ausgabeformate in Form von Portable Document Form (PDF), Postscript und Rich Text Format (RTF) unterstützt werden. Vgl. u. a. Behme / Mintert (2001), S. 123ff.

⁶⁶ Vgl. Behme / Mintert (2001), S. 134f.

⁶⁷ Vgl. Thompson (2002).

⁶⁸ Vgl. Goldfarb / Prescod (2002), S. 1005ff.

⁶⁹ Vgl. Clark / DeRose (1999).

⁷⁰ Vgl. Boag / Chamberlin / Fernandez / Florescu / Robie / Siméon (2002).

⁷¹ Vgl. Weitzel / Harder / Buxmann (2001), S. 40f.

⁷² Vgl. Bach (2000), S. 206ff.

indem beispielsweise Ränder und Abstände zwischen Inhalten festgelegt oder bestimmten Bereichen Hintergrundfarben zugewiesen werden können. Für weitere Ausführungen sei jedoch aufgrund der Komplexität auf die Spezifikation verwiesen.⁷³

5.1.2 Einsatzmöglichkeiten in virtuellen Unternehmen

In diesem Abschnitt soll nun die Einsatzmöglichkeit der Metasprache XML als Basis für Regelwerke im Bereich des zwischenbetrieblichen Datenaustauschs in virtuellen Unternehmen aufgezeigt werden. Daher wird nachfolgend zuerst einmal das prinzipielle Vorgehen bei der Erstellung solcher Regelwerke auf Basis der Metasprache XML fokussiert und dieses Vorgehen beispielhaft für einen konkreten Vorgang durchgeführt. Daran anschließend wird eine prinzipielle Systemarchitektur vorgestellt, welche die wesentlichen Komponenten eines Systems für den XML-basierten Nachrichtenaustausch aufzeigt, und der Ablauf eines Nachrichtenaustauschs beschrieben.

Betrachtet man nun das Vorgehen, so sind folgende Schritte bei der Erstellung der Regelwerke zu durchlaufen:⁷⁴

1. Spezifikation des Nachrichtenaustauschs

Im ersten Schritt werden die relevanten Nachrichtentypen für einen Nachrichtenaustausch ermittelt und deren konkrete Abfolge festgelegt.

2. Identifikation der relevanten Daten

Im zweiten Schritt werden nun die relevanten Daten für die festgelegten Nachrichtentypen bestimmt, die zur korrekten Verarbeitung auf der Empfängerseite benötigt werden.

3. Modellierung der Daten in einer Strukturdefinition

Im dritten Schritt erfolgt dann die Strukturdefinition der Nachricht, die die identifizierten Daten überträgt. Diese erfolgt auf Basis von Document Type Definition (DTD) oder XML Schema.

Dieser Ablauf soll nun beispielhaft aufgezeigt werden. Dabei wurde der Nachrichtenaustausch für eine Status- bzw. Fortschrittsabfrage eines Teilauftrags durch den Auftragskoordinator an die beteiligten Partner ausgewählt, da ein solcher Nachrichtenaustausch häufig bei der Auftragsabwicklung in virtuellen Unternehmen stattfindet. Darüber hinaus ist diesem Vorgang auch eine zeitkritische Komponente zuzuordnen, da der Auftragskoordinator anhand der gelieferten Informationen die Einhaltung von Terminen überprüft. Bei einer Fortschrittsabfrage für spezifizierte Teilaufträge wird die entsprechende Anfrage vom Auftragskoordinator an die Partnerunternehmen übermittelt, die dann

⁷³ Vgl. Adler / Berglund / Caruso / Deach / Graham / Grosso / Gutentag / Milowski / Parnell / Richman / Zilles (2001).

⁷⁴ Vgl. Fischer (1993), S. 241ff.

ihrerseits als Antwort auf diese Anfrage die entsprechenden Fortschrittsinformationen an den Auftragskoordinator zurücksenden. Abbildung 24 veranschaulicht diesen Sachverhalt und zeigt die beiden für diesen Vorgang relevanten Nachrichtentypen auf.

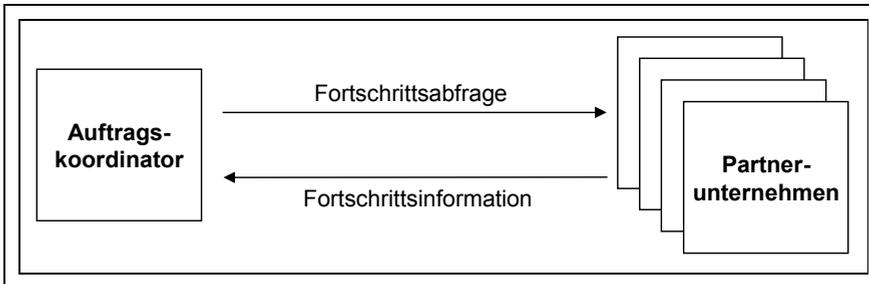


Abbildung 24: Fachliche Spezifikation des Datenaustauschs bei Fortschrittsabfrage

Analysiert man nun, wie im zweiten Schritt vorgesehen, die relevanten Daten, so werden vom Auftragskoordinator an das entsprechenden Partnerunternehmen die eindeutige Kennung des Teilauftrags übermittelt, auf dessen Basis dann das Partnerunternehmen die relevanten Fortschrittsdaten aus dem eigenen Datenbestand extrahieren und an den Auftragskoordinator zurück senden kann. Als Vorschlag wären hier beispielsweise eine Prozentangabe über den Absolvierungsfortschritt, die Auftragskennung und die Partnerkennung denkbar.

Im dritten Schritt werden die identifizierten Daten nun in eine Strukturdefinition für den entsprechenden Nachrichtentypen eingebettet. Diese Strukturdefinition kann dann als Regelwerk für den Datenaustausch aufgefasst werden. Abbildung 25 veranschaulicht die jeweilige Strukturdefinition für die beiden Nachrichtentypen.



Abbildung 25: Beispiele für Strukturdefinitionen für die identifizierten Nachrichtentypen

Nachdem die entsprechenden Nachrichtentypen ermittelt wurden, soll nun der Ablauf einer Kommunikation dargestellt werden. Basisbausteine für die Architektur sind auf der einen Seite der XML Transformator und das Kommunikationsmodul auf der anderen Seite. Der XML Transformator interpretiert bei den jeweiligen am Datenaustausch beteiligten Parteien die ankommenden Nachrichten, nimmt eine Validierung anhand der zugrunde liegenden Strukturdefinition vor und aktualisiert die relevanten Daten in den entsprechenden Datenbeständen oder aber extrahiert die relevanten Daten aus den Datenbeständen und stellt diese zu einer konformen Nachricht in Bezug auf die entsprechende Strukturinformation zusammen. Das Kommunikationsmodul ist für die Übermittlung, also das korrekte Senden und Empfangen der Nachricht, verantwortlich. Abbildung 26 veranschaulicht diesen Sachverhalt. In der Abbildung fungiert Partner A als Sender und somit in Bezug auf das angeführte Beispiel als Auftragskoordinator und Partner B als Empfänger und somit als ausführendes Partnerunternehmen.⁷⁵

⁷⁵ Vgl. Neuburger (1994), S. 23ff.

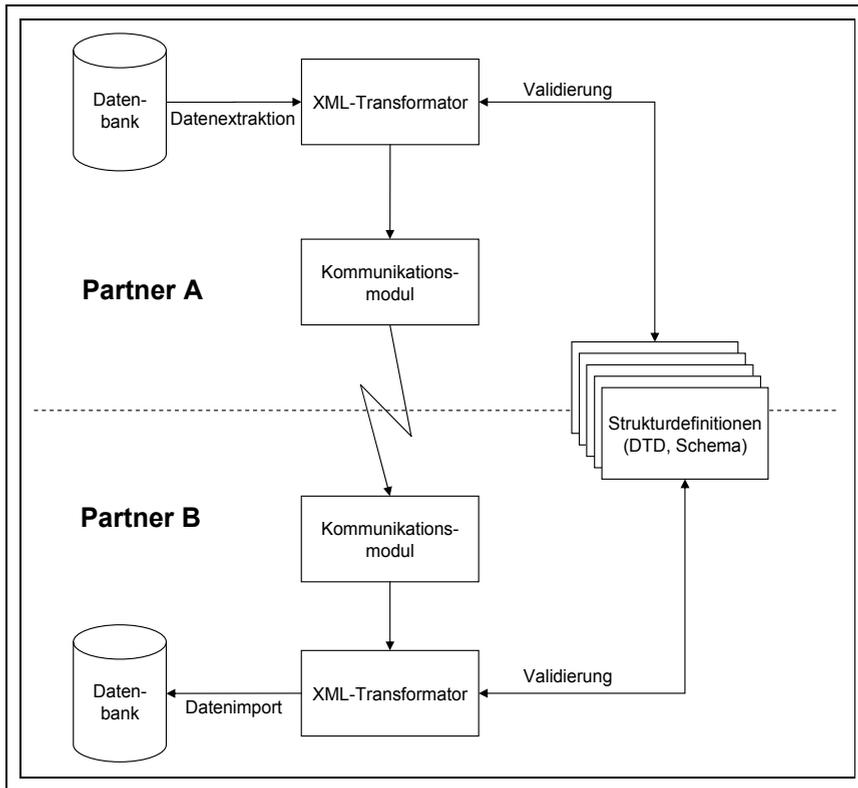


Abbildung 26: Ablauf einer Datenübertragung

5.1.3 Beurteilung

Um elektronischen Datenaustausch auf Basis von XML durchzuführen, müssen sich die beteiligten Parteien zuerst einmal auf einen Pool von Strukturdefinitionen einigen. Dazu ist das im vorigen Abschnitt dargestellte Vorgehen zu durchlaufen. Jedoch ist nicht anzunehmen, dass sich ein fester Pool von Strukturdefinition für virtuelle Unternehmen etabliert, da diese wettbewerbsrelevante Informationen darüber enthalten könnten, in welcher Form die Aufgabenverteilung und die Nutzung der dezentralen Ressourcen abgewickelt wird.⁷⁶

Betrachtet man die Kosten der vorgestellten Architektur, so ist der XML-Parser, der im dargestellten XML-Transformator zum Einsatz kommt, ein Open Source Produkt und damit kostenfrei erhältlich. Auch Programmpakete, die standardisierte Application Programming Interfaces (APIs) für die Verarbeitung von XML-Dokumenten zur Verfügung stellen, sind mittlerweile in alle relevanten Programmiersprachen (C++, Java) integriert. Somit skalieren die Kosten für eine

⁷⁶ Vgl. Hess (2002a), S. 326ff.

solche Architektur in Abhängigkeit mit der Größe des Pools der Strukturdefinitionen und der Komplexität derselbigen. Jedoch sind die Kosten deutlich unterhalb traditioneller EDI-Systeme anzusetzen, da einzelne Module frei verfügbar sind und dies niedrigere Entwicklungskosten für das System nach sich zieht.⁷⁷

Auf der Nutzenseite sind die üblichen Vorteile des elektronischen Datenaustauschs auszumachen. Diese lassen sich anhand ihres Charakters in drei Gruppen unterteilen. Die erste Gruppe umfasst die direkt monetär quantifizierbaren Nutzenpotentiale, die sich in Form von Kostensenkungen bemerkbar machen. Die zweite Gruppe sind die indirekt monetär quantifizierbaren Nutzenpotentiale. Unter diese Gruppe fällt zum Beispiel die Zeiteinsparung, die sich direkt bestimmen lässt, deren monetäre Quantifizierung jedoch indirekt festgestellt werden muss. Die dritte Gruppe sind die monetär nicht quantifizierbaren Nutzenpotentiale, bei denen Qualitätssteigerungen im Bezug auf die Datenbestände berücksichtigt werden und deren monetäre Bewertung nur schwer möglich ist.⁷⁸ Diese skizzierten Vorteile treten besonders beim Datenaustausch in Erscheinung, der einer hohen Häufigkeit unterliegt oder bei welchem zeitkritische Informationen zwischen den beteiligten Parteien übertragen werden.

Anders als bei festen Beziehungen mit standardisierten Auftragsabläufen, etwa in Supply-Chain-Netzwerken, bilden sich in virtuellen Unternehmen flexible, auftragsabhängige Teams.

Es ist daher aus Kosten/Nutzensicht zu prüfen, ob sich die Etablierung eines XM-Standards rechnet. Im konkreten Anwendungsfall beim virtuellen Unternehmen Techno-Pool, das als Entwicklungspartner diente, war der zu erwartende Zusatznutzen sehr gering, sodass auf eine XML-Entwicklung verzichtet wurde.

Zusammenfassend lässt sich festhalten, dass die Nutzung von XML im zwischenbetrieblichen Datenaustausch im Vergleich zu anderen Ansätzen, wie zum Beispiel EDIFACT, vor allem zu einer Reduzierung der Entwicklungskosten führt und somit der Einsatz von XML als Beschreibungssprache für die Nachrichten selbst und für die zugrunde liegenden Strukturdefinitionen befürwortet werden kann. Jedoch sollte, wie bereits dargestellt, zur Ausschöpfung der Vorteile bei der Realisierung mit Vorgängen begonnen werden, bei denen eine große Anzahl von Nachrichten übermittelt werden muss, deren Daten eine zeitkritische Komponente haben.

⁷⁷ Vgl. Weitzel / Harder / Buxmann (2001), S. 7ff.

⁷⁸ Vgl. Brehm (1997), S. 53ff.; Sedran (1991), S. 16ff.

5.2 Web Services

In diesem Abschnitt werden Web Services als eine neue Schnittstellentechnologie auf ihre Einsatzfähigkeit in virtuellen Unternehmen untersucht. Dazu werden einleitend die Grundlagen von Web Services skizziert sowie darauf aufbauend Einsatzmöglichkeiten von Web Services in virtuellen Unternehmen aufgezeigt. Den Abschluss bildet eine Beurteilung von Web Services hinsichtlich ihres Einsatzes in virtuellen Unternehmen.

5.2.1 Grundlagen

Nachfolgend sind die Grundlagen von Web Services niedergelegt. Aufbauend auf eine erarbeitete Begriffsdefinition und daraus abgeleiteten Eigenschaften von Web Services wird auf der einen Seite die Service Oriented Architecture (SOA) als Basisarchitektur von Web Services vorgestellt. Auf der anderen Seite werden die zur Realisierung genutzten und etablierten Basistechnologien (SOAP, WSDL, UDDI) erläutert.

5.2.1.1 Begriffsdefinition

Betrachtet man die Literatur und Fachzeitschriften im Bereich der Web Service Technologie genauer, so fällt auf, dass es keinen Mangel an Definitionen für den Begriff „Web Service“ gibt und nahezu jeder Fachautor seine eigene Begriffsdefinition verwendet. Man ist somit von einer einheitlichen, konsistenten und standardisierten Begriffsdefinition weit entfernt. Daher sollen nun zwei Definitionen vorgestellt und daraus eine Arbeitsdefinition dieses Begriffs abgeleitet werden.⁷⁹ Dabei fiel die Auswahl auf die Definition des W3C, das ein wesentlicher Treiber dieser Innovation ist, sowie auf eines der ersten veröffentlichten Fachbücher von Ethan Cerami zu dieser Thematik.

Auf der einen Seite definiert das World Wide Web Consortium (W3C) den Begriff des Web Services wie folgt:

“A Web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML based messages exchanged via internet-based protocols.”⁸⁰

Auf der anderen Seite schlägt der Cerami folgende Definition vor:

„A web service is any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any one operating system or programming language. Although they are not required, a web service may also have two additional

⁷⁹ Vgl. für einen Überblick von verfügbaren Begriffsdefinitionen Seely (2002).

⁸⁰ Vgl. Austin / Barbir / Ferris / Garg (2002).

*properties: A web service should be selfdescribing and a web service should be discoverable.*⁸¹

Aus diesen Definitionen können Web Services als Softwarekomponenten oder unabhängige, modulare Anwendungen aufgefasst werden. Diese Softwarekomponenten kapseln in sich eine wohl definierte Funktionalität und stellen diese über standardisierte Schnittstellen anderen Softwarekomponenten oder Anwendungen zur Verfügung.⁸² Durch diese Standardisierung der Schnittstellen ergibt sich die Möglichkeit der losen Kopplung bei Web Services. Die Nutzer können die Dienstleistungsanbieter ohne Probleme austauschen, wenn sich die Schnittstellensignatur von den zu substituierenden Diensten nicht unterscheidet. Darüber hinaus ermöglicht die Kapselung von Web Services eine plattform- und programmiersprachenunabhängige Realisierung. Dabei kann ein Web Service immer eindeutig über einen Uniform Resource Identifier (URI) identifiziert werden.⁸³

Web Services sind darüber hinaus selbstbeschreibend, da die bereitgestellten Funktionalitäten und die zur Nutzung notwendige Schnittstellensignatur in Form von XML-Dokumenten zur Verfügung gestellt werden. Auf Basis dieser Dokumente ist es einem Nutzer möglich, einen Web Service in das entsprechende Anwendungsumfeld einzubinden bzw. zu integrieren.⁸⁴ Auch basiert die Suche nach Web Services und letztendlich die Nutzung der Web Services selbst jeweils auf Basisstandards, die auf der Metasprache XML und etablierten Internettechnologien wie beispielsweise TCP/IP und HTTP aufsetzen. Damit übernehmen Web Services die bereits im vorigen Abschnitt aufgeführten Vorteile der Metasprache XML und basieren nicht auf proprietären sondern etablierten Protokollen. Somit können Web Services plattform- und betriebssystemunabhängig eingesetzt werden.

Im Gegensatz zu dieser eher technischen Sichtweise auf diese Innovation können Web Services aus einer ökonomischen Sicht auch als gekapselte und modulare Geschäftsanwendungen (Business Units) angesehen werden, die eine fest definierte Geschäftsfunktionalität abbilden. Somit stellen Web Services eine neue Möglichkeit dar, so genannte digitale Dienstleistungen über Netzwerke und standardisierte Schnittstellen zu vertreiben. Die angebotenen digitalen Dienstleistungen bzw. Geschäftsprozessfunktionalitäten können dabei entweder einzeln verwendet oder zu komplexen Geschäftsabläufen zusammengefügt werden.⁸⁵ Dabei wird dann von so genannten Super-Services gesprochen.

⁸¹ Vgl. Cerami (2002), S. 3f.

⁸² Vgl. Bettag (2001), S. 302ff.

⁸³ Vgl. Austin / Barbir / Ferris / Garg (2002).

⁸⁴ Vgl. Hoidn / Jungclaus (2002), S. 32.

⁸⁵ Vgl. Kreger (2001), S. 6.

5.2.1.2 Architekturmodell

Bei dem Web Service Architekturmodell handelt es sich um die konkrete Umsetzung der bereits seit Anfang der 70er Jahre diskutierten Service Oriented Architecture (SOA). Innerhalb dieser Architektur werden drei verschiedene Rollen unterschieden, die untereinander interagieren und Nachrichten austauschen. Zentrale Bestandteile bei der Interaktion der drei beteiligten Parteien sind auf der einen Seite der Dienst (Service) und auf der anderen Seite die Dienstbeschreibung.⁸⁶ Abbildung 27 veranschaulicht die drei Rollen, die Interaktionen zwischen den Rollen sowie die zentralen Bestandteile dieser Architektur graphisch. Nachfolgend sollen die Elemente aufgegriffen und erläutert werden.

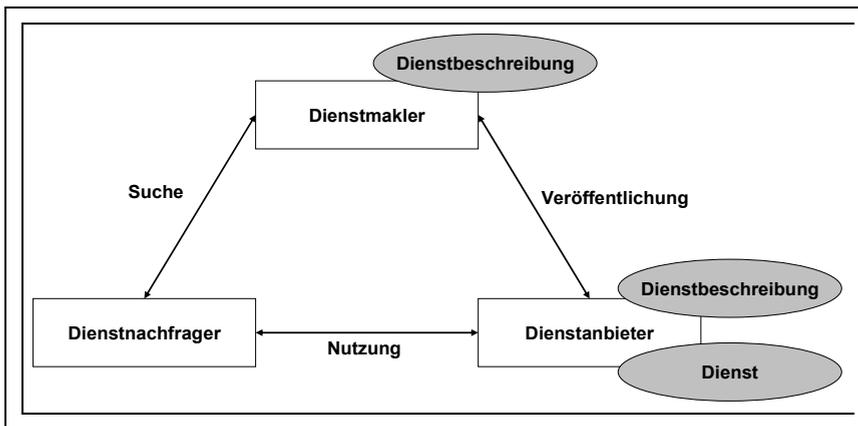


Abbildung 27: Architekturmodell für Web Services

Betrachtet man zunächst die drei Rollen in diesem Architekturkonzept, so kann man diesen drei Rollen die folgenden Aufgaben zuordnen, wobei bei jeder Rolle eine fachliche, eine technische und eine geschäftsorientierte Betrachtungsperspektive dargestellt wird:⁸⁷

- **Dienstanbieter**

Der Dienstanbieter (Service Provider) stellt einen bestimmten Dienst zur Verfügung und beschreibt und annonciert diesen. Der Dienstanbieter übernimmt somit aus der fachlichen Perspektive die Konzeption des Dienstes selbst. Darüber hinaus ist der Dienstanbieter aus technischer Sicht für die technische Umsetzung des Moduls sowie den Betrieb und die Wartung. Aus einem geschäftsorientierten Betrachtungspunkt ist er somit der Eigentümer des Moduls, der eine bestimmte Funktionalität zur Verfügung stellt, und der Betreiber der Plattform und des Dienstes.

⁸⁶ Vgl. Stevens (2002a); Stevens (2002b).

⁸⁷ Vgl. Snell / Tidwell / Kulchenko (2002), S: 5ff.; Cerami (2002), S. 9ff.

- **Dienstinachfrager**

Der Dienstinachfrager (Service Requestor) nutzt den Dienst, indem er ihn in seine Geschäftsprozesse und seine Anwendungen integriert. Aus der technischen Sicht handelt es sich um eine Anwendung, die den Dienst aufruft oder mit ihm interagiert. Dabei ist es unerheblich, ob die Anwendung durch eine Person bedient wird oder nicht. Im Rahmen der geschäftsorientierten Perspektive stellt der Nachfrager also den Konsumenten des Dienstes dar.

- **Dienstmakler**

Der Dienstmakler (Service Registry) ermöglicht aus der fachlichen Sicht die Speicherung der Dienstbeschreibungen in einem zentralen Verzeichnis und ermöglicht dem Dienstanbieter die Aktualisierung der hinterlegten Informationen sowie dem Dienstinutzer die Suche nach entsprechenden Diensten und den Bezug der Dienstbeschreibungen. Um die Suche nach diesen Beschreibungen möglichst performant und flexibel gestalten zu können, können optional anhand von Metadaten kategorisiert werden. Durch diese Kategorisierung werden verschiedene Sichten auf die Dienste und deren Beschreibungen ermöglicht bzw. die Suche nach bestimmten Diensten vereinfacht. Ein Verzeichnis (Yellow Pages) ermöglicht die Kategorisierung der Dienste anhand von Branchen. Auf Basis eines Anbieterverzeichnisses (White Pages) wird eine Kategorisierung anhand der Dienstanbieter angeboten. Schließlich gibt es noch ein technisches Verzeichnis (Green Pages), in welchem Verweise auf Adressen abgelegt sind, an denen sich die Dienstbeschreibungen befinden. Diese abgelegten Dienstbeschreibungen können dann durch den Dienstinachfrager bei der Suche inspiziert werden.⁸⁸ Aus der technischen Perspektive handelt es sich um eine Anwendung, welche die für den Dienst zugrunde liegenden fachlichen Anforderungen umsetzt. Aus der geschäftsorientierten Perspektive handelt es sich um ein Unternehmen, welches diese Suchdienste und die Ablagefunktionalitäten zur Verfügung stellt. Somit wird dadurch eine Art Vermittleraufgabe zwischen dem Dienstanbieter und dem Dienstinachfrager übernommen.

Die Interaktion zwischen den drei aufgezeigten Rollen kann dabei unter den Schlagwörtern der Veröffentlichung, der Suche und der Nutzung zusammengefasst werden, die nachfolgend näher erläutert werden sollen:⁸⁹

- **Veröffentlichung**

Damit eine Nutzung des Dienstes möglich wird, muss der Anbieter die Dienstbeschreibung dem Dienstinutzer zukommen lassen. Dazu veröffentlicht er diese Beschreibung beim Dienstmakler, über den der

⁸⁸ Vgl. Seely (2002), S. 197ff.; Snell / Tidwell / Kulchenko (2002), S. 105ff.; Snell / Tidwell / Kulchenko (2002), S. 5.

⁸⁹ Vgl. Beimborn / Mintert / Weitzel (2002), S. 277f.; Bettag (2001), S. 302ff.

Anbieter dann diese Informationen bezieht. Neben diesem indirekten Weg kann der Dienstanbieter aber auch direkt den Dienstanbieter über die Dienstbeschreibung informieren, falls dieser bekannt ist. In diesem Zusammenhang wird auch die Komplementärfunktion der Löschung der Veröffentlichung eingegliedert, die anfällt, wenn ein Dienst zurückgezogen werden soll.

- **Suche**

Bei dieser Interaktion durchsucht der Dienstanbieter die beim Dienstmakler abgelegten Dienstbeschreibungen. Dabei wird die Suche durch die Kategorisierung der Beschreibungen erleichtert. Bei einer erfolgreichen Suche bezieht der Dienstanbieter dann die Dienstbeschreibung bzw. einen Verweis auf die selbige, die als Basis für die Dienstnutzung ausreicht.

- **Nutzung**

Durch die vorhandene Dienstbeschreibung ist der Dienstanbieter nun über die vorhandenen Schnittstellen des Dienstes informiert und kann den Dienst in Anspruch nehmen.

Wesentlicher Bestandteil dieser Architektur sind auf der einen Seite der Dienst selbst, der wie bereits angeführt ein Softwaremodul darstellt, welches veröffentlicht und über das Netzwerk durch den Dienstanbieter angesprochen werden kann. Auf der anderen Seite bildet die Dienstbeschreibung einen zentralen Bestandteil, auf deren Basis Informationen über den Dienst wie die Schnittstellenspezifikation, genutzte Datentypen und Bindungsinformationen sowie die Netzwerkadresse übermittelt werden. Darüber hinaus kann die Dienstbeschreibung Zusatzinformationen für die Kategorisierung enthalten.

Aus diesen Ausführungen lässt sich nun ein Lebenszyklus für einen Web Service ableiten. Zu Beginn wird der Dienst durch den Dienstanbieter konzipiert und realisiert. Daran anschließend wird er beim Dienstmakler veröffentlicht. Der Dienstanbieter informiert sich beim Dienstmakler über angebotene Dienste, bezieht die Dienstbeschreibung und kann auf Basis dieser Dienstbeschreibung den Dienst einbinden und nutzen. Am Ende kann der Dienstanbieter den Dienst zurückziehen, in dem er die Veröffentlichung rückgängig macht und optional die Dienstanbieter über diesen Schritt informiert. Dieser Lebenszyklus ist in Abbildung 28 dargestellt.⁹⁰

⁹⁰ Vgl. Burghardt / Gehrke / Schumann (2003), S. 72f.; Rawolle / Burghardt (2002), S. 40ff.

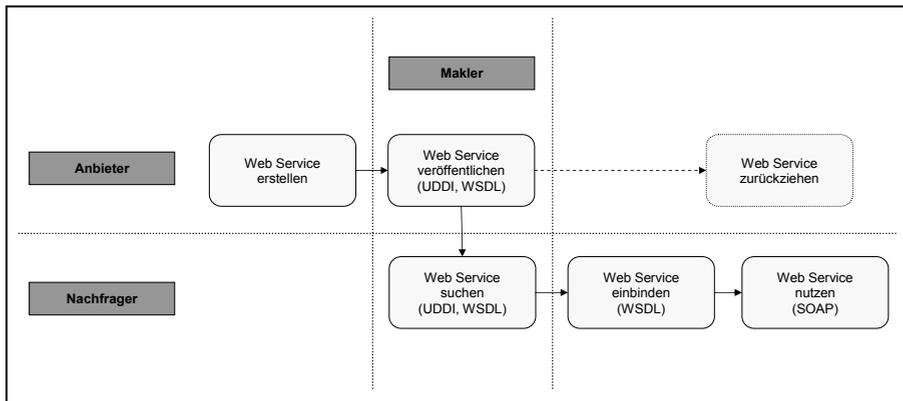


Abbildung 28: Lebenszyklus eines Web Services

Die gesamte Interaktion zwischen den drei beteiligten Rollen sowie die Nutzung des Dienstes erfolgen dabei über standardisierte Schnittstellen, die über ein Netzwerk angesprochen, dynamisch gefunden und eingebunden werden können.⁹¹ Somit kann dieses Architekturkonzept als eine Weiterentwicklung des traditionellen Client-Server-Konzepts gesehen werden.⁹²

5.2.1.3 Basistechnologien

Die bei der Umsetzung verwendeten Basistechnologien und Standards, welche die obigen Ausführungen bzgl. der Architektur realisieren, können anhand eines Zwiebschalenmodells bzw. Schichtenmodells in eine Rangfolge gebracht werden. Dabei werden mit jeder neuen Schicht neue Funktionsbereiche realisiert, die fachliche Komplexität nimmt vom Kern der Zwiebel nach außen hin immer weiter zu. Im Kern der Zwiebel ist die eigentliche Dienstleistung angesiedelt, deren Funktionalität durch entsprechende Schnittstellen dem Dienstinachfrager angeboten werden soll. Dabei werden die für die Durchführung der Dienstleistung notwendigen Informationen in Dokumenten auf Basis der Metasprache XML gekapselt. Die darüber angesiedelten einzelnen Funktionsbereiche können dann mit den Begrifflichkeiten der Dienstkommunikation, der Dienstbeschreibung und des Dienstverzeichnisses für die Suche beschrieben werden.

Die in diesen Bereichen etablierten Basistechnologien ab der Dienstkommunikationsschicht sollen nachfolgend beschrieben werden. Abbildung 29 veranschaulicht das Zwiebschalenmodell graphisch und kennzeichnet die Funktionsbereiche.

⁹¹ Vgl. Löwer / Picot (2002), S. 21ff.

⁹² Vgl. Sleeper / Robins (2002).

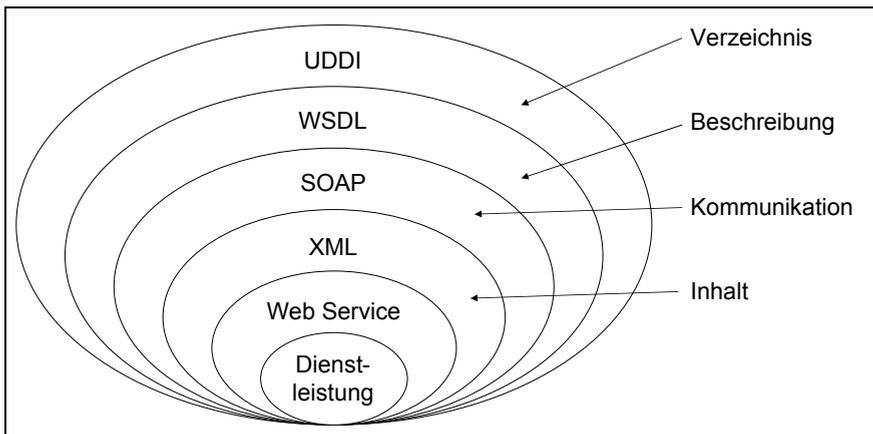


Abbildung 29: Zwiebelschalenmodell zur Anordnung der Basistechnologien bei Web Services

Im Bereich der Dienstkommunikation hat sich SOAP (Simple Object Access Protokoll) als Standard für die Nachrichtenkapselung durchgesetzt. Hierbei handelt es sich um ein einfaches, sprach-, plattform- und betriebssystemunabhängiges Protokoll für den entfernten Prozeduraufruf. Sowohl der Aufruf als auch die Antwort erfolgt in Form eines wohlgeformten XML-Dokuments. Dabei lässt sich die SOAP-Nachricht in einen Umschlag (SOAP:Envelope) und einen Innenteil zerlegen, wobei der Innenteil wiederum in einen optionalen Header (SOAP:Header) und einen Hauptteil (SOAP:Body) zerfällt. Im Hauptteil sind dann alle notwendigen Informationen wie der Methodename und die Parameter im Fall des Methodenaufrufs beziehungsweise die Rückgabewerte oder Fehlermeldungen im Falle der Antwortnachricht eingebettet.⁹³ Des Weiteren zeichnet sich SOAP durch seine Flexibilität in Bezug auf das verwendete Anwendungsprotokoll (eine Vielzahl von Bindungen an Internetprotokolle wie z.B. HTTP, RMI/IIOP, SMTP, FTP und MQ ist möglich), auf die verwendete Struktur der Nachricht (Transport von beliebigen XML-Daten durch die Verwendung von Namensräumen) und auf das Kommunikationsmodell (Unterstützung von Messaging und Weiterleiten der Nachrichten über das RPC-Modell hinaus) aus.⁹⁴ Derzeit sind in der Spezifikation SOAP 1.2 nur Bindungen an HTTP GET/POST und SMTP beschrieben (Stand: Mai 2004).⁹⁵

Im Bereich der Dienstbeschreibung hat sich die XML-basierte Auszeichnungssprache WSDL (Web Service Description Language) als IDL (Interface Definition Language) etabliert. Durch WSDL werden alle notwendigen technischen Informationen gekapselt, die zum Aufruf und zur Nutzung eines Web Services nötig sind. Als Beispiele für die hinterlegten Informationen seien an dieser Stelle

⁹³ Vgl. Scribner / Stiver (2000), S. 107ff.

⁹⁴ Vgl. Vasudevan (2001).

⁹⁵ Vgl. Christensen / Curbera / Meredith / Weerawarana (2002).

die abstrakten Nachrichtenformate (Anfrage oder Antwort), die Art der Bindung (Anfrage, Anfrage-Antwort, Antwort) und die Bindung an das genutzte Internetprotokoll selbst (HTTP, SMTP) genannt.

Bei der Dienstauffindung hat sich UDDI (Universal Description, Discovery and Integration) als ein möglicher Verzeichnisdienst durchgesetzt. Die UDDI Spezifikation bietet einen Mechanismus zur Beschreibung des Anbieters und zur Veröffentlichung von Web Services. Zentraler Bestandteil von UDDI ist ein Verzeichnis, in denen die Anbieter von Web Services Informationen über sich selbst und die angebotenen Dienste sowie deren Zugriffsmöglichkeit hinterlegen. Dieses Verzeichnis besteht aus weißen, gelben und grünen Seiten. Die weißen Seiten enthalten allgemeine Informationen über den Anbieter wie Name, Adresse, Kontaktperson, etc. Die grünen Seiten ermöglichen die Hinterlegung der technischen Beschreibung unter anderem in Form eines Verweises auf ein WSDL-Dokument. Um eine leichtere Suche innerhalb des Verzeichnisdienstes zu ermöglichen, können sich die Unternehmen beispielsweise in den gelben Seiten nach geographischen Parametern kategorisieren. Für die Aktualität der hinterlegten Informationen ist der Web Service Anbieter selbst verantwortlich.

5.2.2 Einsatzmöglichkeiten in virtuellen Unternehmen

Aus den Ausführungen lassen sich zwei wesentliche Einsatzmöglichkeiten für Web Services ableiten, die beide als netzwerkinterne Dienstleistungen angesehen werden können und nachfolgend diskutiert sowie erläutert werden sollen.

Auf der einen Seite können die Partnerunternehmen den bereits im vorigen Abschnitt diskutierten zwischenbetrieblichen Nachrichtenaustausch durch Web Services unterstützen bzw. auf deren Basis realisieren. Dazu könnte das in diesem Abschnitt aufgezeigte Kommunikationsmodul mit einer Web Service Schnittstelle versehen werden. Dieses stellt dann als zentrale Komponente innerhalb eines jeden Partnerunternehmens umfangreiche Funktionalitäten für den Datenaustausch bereit. Im Idealfall wäre die komplette Kommunikation innerhalb solcher virtuellen Unternehmen dann über diese Komponente abgedeckt.

Beispielsweise könnte bei jedem Partner eine Schnittstelle zur Abfrage von Fortschrittinformationen realisiert werden, die dann vom Auftragskoordinator aktiviert wird und als Übergabe- und Rückgabedokumente die im vorigen Abschnitt diskutierten XML-Dokumente erwartet. Da die Kommunikation bei Web Services auf Basis des Kommunikationsprotokolls SOAP abläuft und bei der Verwendung von SOAP beliebige XML-Dokumente als Nutzlast (Payload) der Nachricht übermittelt werden können, ergänzen sich hier die beiden vorgestellten Konzepte des Einsatzes von XML als Basis für die Nachrichtentypen und Web Services als Basis für die Realisierung.

Abbildung 30 veranschaulicht dieses graphisch, die Schnittstellen auf Basis von Web Services sind entsprechend hervorgehoben.

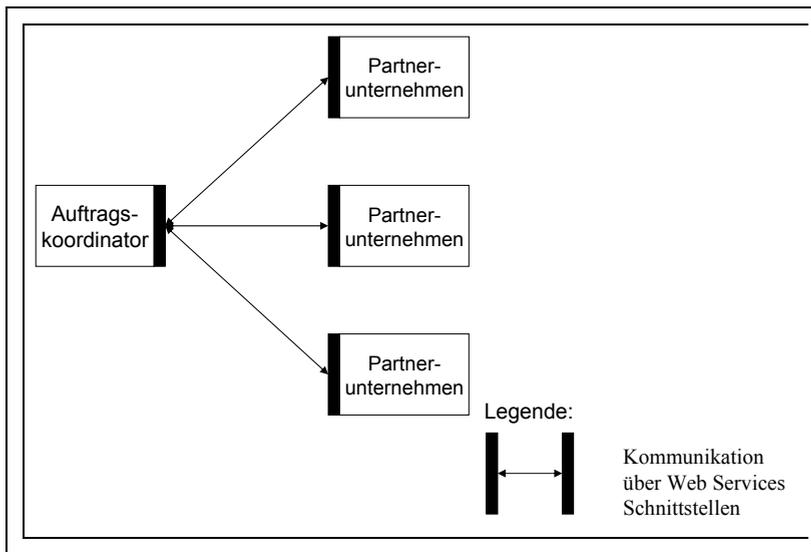


Abbildung 30: Kommunikation der Partnerunternehmen über Web Service Schnittstellen

Auf der anderen Seite könnte auch ein zentral betriebenes Informationssystem realisiert werden, welches alle fachlichen Anforderungen in einem solchen virtuellen Unternehmen abdeckt. Diese fachlichen Anforderungen können auf zwei verschiedenen Ebenen dargestellt werden.

Auf der auftragsbezogenen Ebene soll der komplette Lebenszyklus eines Auftrags im virtuellen Unternehmen abgebildet werden. Dazu zählen neben der Anfrageerfassung die Auftragsaufteilung und Ausschreibung, die Teilangebotserstellung, die Angebotsaggregation und bei Zustimmung des Auftragsgebers dann die Auftragsabwicklung und die Auftragsabrechnung. Diese Aufgaben lassen sich, wie bereits in Kapitel 3 geschehen, der Informationsphase, der Vereinbarungsphase, der Durchführungsphase sowie der Auflösungsphase zuordnen.

Auf der auftragsübergreifenden Ebene kann dann durch ein Netzwerk-Controlling, das alle aus einem Partnerpool gebildeten Auftragskonfiguration zur Durchführung der Aufträge umfasst, eine Messung des Erfolgs des virtuellen Unternehmens stattfinden. Die relevanten Erfolgsgrößen werden von den Partnerunternehmen festgelegt und ähneln Erfolgsgrößen „klassischer“ Unternehmen. Allerdings wurde im vorliegenden Projekt auf das Netzwerkcontrolling kein Schwerpunkt gelegt, es besitzt jedoch einen hohen Stellenwert im Rahmen der auftragsübergreifenden Bewertung von virtuellen Unternehmen.

Diese fachlichen Anforderungen können dann so umgesetzt werden, dass alle Funktionalitäten über Schnittstellen auf Basis von Web Services realisiert werden. Auf diese Art hat man die Möglichkeit, die Funktionalitäten aus einer Webanwendung heraus zu nutzen und somit eine plattformunabhängige Applikation

auf Basis eines Browsers zu realisieren. Je nach Ausprägung dieser Webapplikation könnte diese dann durch die Partnerunternehmen als zentrales IV-System für das virtuelle Unternehmen oder aber durch die Kunden des virtuellen Unternehmens als Informationsplattform genutzt werden. Darüber hinaus bietet diese Architektur auch die Möglichkeit, die Schnittstellen direkt aus den Anwendungen der Partnerunternehmen bzw. über das entsprechende Kommunikationsmodul zu nutzen, so dass auf diese Art eine hohe Flexibilität in Bezug auf die Nutzung dieser Schnittstellen verwirklicht wird.

Ein entsprechendes Informationssystem wurde bereits am Institut für Wirtschaftsinformatik Abteilung Wirtschaftsinformatik II der Universität Göttingen prototypisch umgesetzt. Es unterstützt dabei alle notwendigen Funktionalitäten auf der auftragsbezogenen Ebene. Dazu zählen unter anderem die Stammdatenverwaltung der Partnerunternehmen und Kunden sowie beispielsweise die Verwaltung des kompletten Lebenszyklus eines Auftrags von der Erfassung bis hin zur Abrechnung.

Abbildung 31 veranschaulicht auch diese Architektur graphisch.

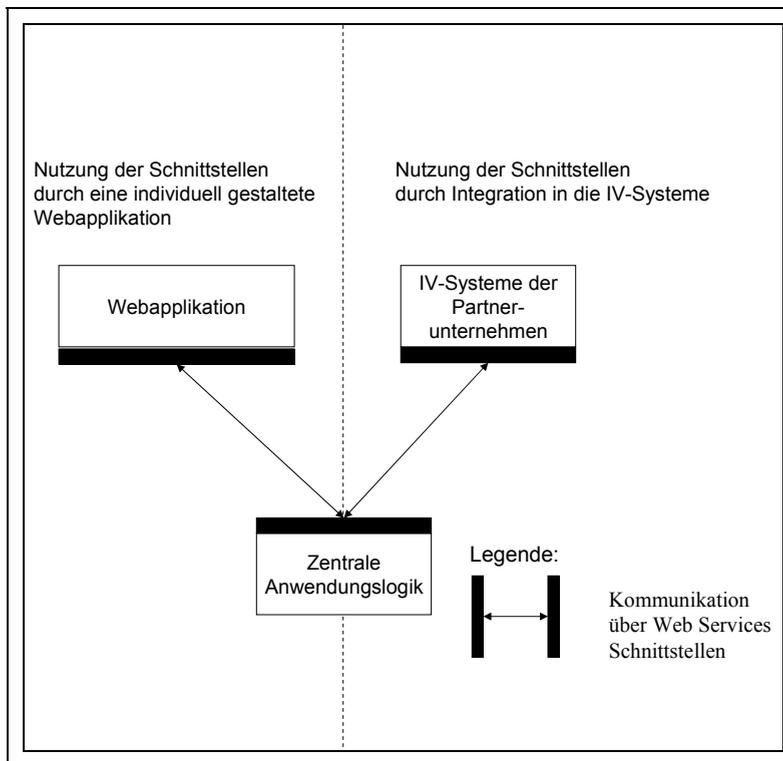


Abbildung 31: Gekapselte zentrale Anwendungslogik auf Basis von Web Services

5.2.3 Beurteilung

Die zugrunde liegende Architektur bei Web Services zeichnet sich durch eine hohe Interoperabilität aus. Diese wird durch die Kerntechnologien, die auf der Metasprache XML basieren, und eine breite Unterstützung durch verschiedene Konsortien gewährleistet. Die Verwendung dieser Kerntechnologien ermöglicht darüber hinaus die Möglichkeit, den Zugriff sowohl für den Menschen (durch Nutzung geeigneter Desktop-Anwendungen) als auch für Maschinen (durch Nutzung von geeigneter APIs) zu ermöglichen. Diese beiden Möglichkeiten des Zugriffs wurden auch bei den vorgestellten Einsatzmöglichkeiten aufgegriffen und sind von zentraler Bedeutung für den hohen Nutzen.

Durch die Kapselung der Funktionalität und durch den möglichen modularen Aufbau von komplexen Web Services selbst wird auch eine hohe Mehrfachverwendbarkeit von Web Services erreicht. Darüber hinaus wird durch die Modularität die Möglichkeit geschaffen, Web Services zu verteilen und somit dezentral zu betreiben. Dies kommt vor allem der dezentralen Struktur in solchen virtuellen Unternehmen entgegen.

Als wesentlicher Vorteil ist jedoch die lose Kopplung von Web Services zu sehen, die durch die Existenz fest definierter Schnittstellen erreicht wird. Daher ist es möglich, Web Services ohne aufwändige Integrationsleistungen zu nutzen bzw. Web Services bei gleicher Schnittstellensignatur gegeneinander austauschen. Somit können die Partnerunternehmen, je nach Auftragsstyp, der durch ein speziell gebildetes virtuelles Unternehmen abgewickelt werden soll, die angebotenen Dienste beliebig genutzt und kombiniert werden. Dabei treten die Partnerunternehmen selbst als Dienstanbieter in Erscheinung.

Die Praktikabilität des Einsatzes von Web Services wurde in einem extra hierfür eingerichteten zweiten Testprototyp untersucht und als hoch relevant eingestuft. Im lauffähigen Prototypen, der im Kapitel fünf beschrieben ist, wurde die Web Services Technologie aufgrund einer vom Praxispartner gewünschten, bewährten Architektur und des Projektfortschritts bei Abschluss dieser Untersuchung nicht aufgegriffen. Für die Weiterentwicklung des Systems bzw. die Entwicklung nachfolgender Software sollte der Einsatz von Web Services jedoch unbedingt in Erwägung gezogen werden.

5.3 Online Analytical Processing (OLAP)

Nach diesem eben diskutierten technisch geprägten Konzepten folgt in diesem Abschnitt die Diskussion eines Vorgehens für die Integration und Analyse von Datenbeständen innerhalb virtueller Unternehmen. Zielsetzung ist es, zu untersuchen, inwieweit sich OLAP als ein Konzept für die Analyse der Daten in dem hier diskutierten Problemkreis virtueller Unternehmen eignet und in welcher Form eine Unterstützung im Bereich des Controllings erreicht werden kann.

Spezielles Augenmerk wird dabei auf die speziellen Möglichkeiten der Datenanalyse gelegt, die durch die speziellen Funktionalitäten von OLAP-Systemen erreicht wird. Dazu werden einleitend die Grundlagen von OLAP erläutert. Daran anschließend werden Einsatzmöglichkeiten aufgezeigt. Der Abschnitt schließt mit einer kurzen Eignungsbewertung von OLAP.

5.3.1 Grundlagen

Um die Grundlagen von OLAP strukturiert zu erläutern, wurde der Abschnitt wie folgt untergliedert. Zu Beginn wird das grundlegende Konzept von OLAP-Systemen erläutert. Darauf aufbauend wird eine Abgrenzung von verwandten Konzepten zur Datenanalyse vorgenommen sowie mögliche Architekturen von OLAP-Systemen diskutiert.

5.3.1.1 *Konzept*

Im Bereich der Datenspeicherung von operativen Anwendungssystemen haben sich relationale Datenbanken als Speichersysteme durchgesetzt. Sie bilden die Basis nahezu jeder Standardanwendung.⁹⁶ Relationale Datenbanken zeichnen sich durch eine starke Transaktionsorientierung aus und werden daher als OLTP-Systeme (OnLine Transaction Processing) bezeichnet. Jedoch sind OLTP-Systeme aufgrund des Fokus auf operative Transaktionen und der damit einhergehenden Realisierung des Echtzeitbetriebs nur bedingt in der Lage, komplexe Analysen und Auswertungen auf dem operativen Datenbestand auszuführen. Sie sind daher im Bereich der Analyse von Daten nur beschränkt einsetzbar.⁹⁷

Dieser Schwachpunkt von OLTP-Systemen wird durch OLAP-Systeme aufgegriffen, deren Aufgabe nicht die Datenspeicherung im operativen Betrieb, sondern die Analyse der vorhandenen Unternehmensdaten ist. Charakteristisch für OLAP-Systeme ist die mehrdimensionale Darstellung und Analyse der Daten, die insbesondere auf der besonderen Ausgestaltung betriebswirtschaftlicher Fragestellungen beruht. Dabei wird die Mehrdimensionalität durch die Anordnung betriebswirtschaftlicher Kennzahlen (Gewinn, Kosten, etc.) entlang unterschiedlicher Dimensionen (Zeit, Produktart, etc.) erzeugt. Die Bildung der Dimensionen führt zu einer Verdichtung der Daten entlang eines so genannten Konsolidierungspfades. Als Beispiel kann der Konsolidierungspfad bei der Dimension „Zeit“ über den Tag, den Monat, das Quartal und das Jahr dienen.⁹⁸

Bei der optischen Darstellung der analysierten Daten bzw. der Ergebnisse werden in der Theorie so genannte Hyperwürfel (Data Cubes) verwendet, die durch die verwendeten Dimensionen aufgespannt werden und sich durch eine unbegrenzte Anzahl von möglichen Dimensionen auszeichnen. In der Praxis

⁹⁶ Vgl. Totok (2000), S. 41.

⁹⁷ Vgl. Haberstock (2000), S. 63.

⁹⁸ Vgl. Totok (2000), S. 41ff.

kommen jedoch nur Hyperwürfel mit drei Dimensionen zu Einsatz, um die menschliche Interpretation zu gewährleisten. Die jeweiligen Ausprägungen der Dimensionen kommen dann durch eine entsprechende Textindizierung der Würfelkanten zum Ausdruck.⁹⁹

Um eine Navigation innerhalb eines solchen Würfels zu erreichen, sind durch das OLAP-Systeme verschiedene Funktionalitäten vorzuhalten. Auf der einen Seite ermöglichen so genannte „*Slice-and-Dice*“ Funktionalitäten, nicht relevante Ausprägungen einer Dimension herauszuschneiden bzw. durch Vertauschen von Würfelkanten und damit durch Vertauschen von vorhandenen Dimensionen bzw. durch Drehen des Würfels eine alternative Datensicht zu erzeugen. Auf der anderen Seite ermöglichen so genannte „*Drill down*“ und „*Drill up*“ Funktionalitäten dem Anwender, das Aggregationsniveau der einzelnen Dimensionen zu verändern, um somit eine Navigation durch die verschiedenen Informationshierarchien entlang des Konsolidierungspfads zu ermöglichen.¹⁰⁰ Somit hält das OLAP-System Datenverdichtungsvorschriften in Abhängigkeit vom Konsolidierungspfad und der konkreten Ausprägung einer Dimension vor.

OLAP-Systeme sollten die in einem Standardwerk von CODD spezifizierten Regeln für eine multidimensionale Datenauswertung erfüllen. Auf der anderen Seite bietet der FASMI-Test des OLAP-Reports eine Möglichkeit, zu überprüfen, ob ein Anwendungssystem als OLAP-fähig im Sinne der mehrdimensionalen Datenauswertung bezeichnet werden kann.¹⁰¹

5.3.1.2 *Einordnung in verwandte Konzepte*

Im Zusammenhang mit OLAP-Systemen werden häufig die Begriffe Data Warehouse, Data Report und Data Mining genannt, so dass eine kurze Abgrenzung zum OLAP-Konzept und zu entsprechenden Systemen erfolgen muss.

Unter dem Begriff des Data Warehouse versteht man die logische Konzeption einer unternehmensweiten, analyseorientierten Datenbasis. Im Gegensatz dazu liegt der Fokus von OLAP, wie bereits dargestellt, in der Analyse der Unternehmensdaten. Somit kann ein Data Warehouse durchaus als Basis für eine solche Analyse der Unternehmensdaten ausgenutzt werden, ist jedoch keine zwingende Voraussetzung für die Nutzung von OLAP-Systemen. Beide Konzepte schließen sich somit nicht gegenseitig aus, sondern ergänzen sich in nahezu idealer Weise.

Unter Data Report werden gerichtete, nutzergesteuerte Datenanalysen verstanden, bei den der Anwender präzise und eigenständig die Vorschrift der Datenanalyse beispielsweise in Form eines SQL-Befehls formuliert und die

⁹⁹ Vgl. Merz (2002), S. 96.

¹⁰⁰ Vgl. Totok (2000), S. 63ff.; Oehler (2000), S. 176ff.

¹⁰¹ Vgl. Totok (2000), S. 61f.; Holthuis (1999), S. 55.; Mucksch (1998), S. 410.

Visualisierung der Ergebnisse selbst übernimmt. OLAP-Systeme unterstützen den Anwender ebenfalls bei einer gerichteten, nutzergesteuerten Analyse von Datenbeständen, indem diese Abfragen in unterschiedlicher Form ermöglichen, entsprechende Unterstützung bei der Erstellung solcher Abfragen leisten und die Ergebnisse in Würfelform anwenderfreundlich visualisieren. Somit können OLAP-Systeme als eine Weiterentwicklung von traditionellen Data Reports gesehen werden, bei denen die Komplexität der Datenanalyse durch die geeignete Unterstützung durch OLAP-Systeme zunimmt.

Beim Data Mining hingegen findet im Gegensatz zum OLAP-Konzept eine ungerichtete, datengesteuerte Selektion und Analyse der Datenbestände statt. Aufgrund der datengesteuerten Auslösung einer solchen Analyse nimmt die Komplexität der Datenanalyse beim Data Mining weiter zu. Die drei vorgestellten Konzepte der Datenanalyse (Data Report, OLAP und Data Mining) stehen somit nicht in Konkurrenz zueinander, sondern stellen verschiedene Formen der Datenanalyse dar und ergänzen sich somit.

In der Literatur findet daher eine Unterscheidung anhand der Auslöser (Anwender beziehungsweise Daten) in ein Verifikationsmodell (Data Report, OLAP) und ein Entdeckungsmodell (Data Mining) statt.¹⁰²

5.3.1.3 *Architekturen*

Betrachtet man die möglichen Architekturen von OLAP-Systemen, so lassen sich die drei grundsätzlichen Architekturen des virtuellen OLAP, des ROLAP und des MOLAP bei OLAP-Systemen unterscheiden, die nachfolgend kurz dargestellt werden sollen.

Bei allen drei Ansätzen dienen geeignete Werkzeuge wie beispielsweise Browser, Tabellenkalkulationen oder Business Intelligence Tools (BIT) als Front-End für die Erstellung der Analyseabfragen und für die Visualisierung der Ergebnisse als Würfel, die auf den OLAP-Systemen aufsetzen und selbst um OLAP-Funktionalitäten angereichert wurden.¹⁰³ Lediglich die Art der Datenbank, auf die die Werkzeuge bei der Analyse der Datenbestände zugreifen, unterscheidet sich bei den drei möglichen Architekturvarianten.

Beim virtuellen OLAP werden durch die Werkzeuge erzeugten Analyseanfragen direkt an die relationale Datenbank weitergeleitet, eine Aufbereitung der Unternehmensdaten für Auswertungszwecke findet nicht statt. Es werden somit direkt die operativen Unternehmensdatenbestände genutzt.

Beim relationalen OLAP (ROLAP) werden die Analyseanfragen ebenfalls an eine relationale Datenbank weitergeleitet. Jedoch wurden die Unternehmensdatenbestände vorher um eine so genannte Dimensionstabelle angereichert, die benötigte Meta- und Aggregationsinformationen für die multidimensionale

¹⁰² Vgl. Merz (2002), S. 97.

¹⁰³ Vgl. Schinzer / Bange / Mertens (1999), S. 17.; Totok (2000), S. 67ff.

Auswertung enthält. Bei diesen beiden Varianten werden also die in einer relationalen Datenbank gespeicherten bestehenden Datenbestände, teilweise um eine Dimensionstabelle angereichert, genutzt. Die multidimensionale Datensicht wird im Werkzeug erzeugt. Die multidimensionale Datenbank besteht somit also nur virtuell.¹⁰⁴

Beim multidimensionalen OLAP (MOLAP) wird für jeden potentiell möglichen Wert, der sich aus den Schnittpunkten der Ausprägungen der Dimensionen ergibt, ein Speicherplatz vorgehalten und wenn möglich mit einem gültigen Wert belegt.¹⁰⁵ Diese Belegung mit einem gültigen Wert ist jedoch nicht immer möglich. Die multidimensionale Datenbank setzt somit die auf konzeptioneller Ebene entwickelten Hyperwürfel unmittelbar in eine eigene Speicherstruktur um. Die vorhandenen Unternehmensdatenbestände werden also in einer aggregierten Form in der multidimensionalen Datenbank abgelegt. Jedoch handelt es sich bei multidimensionalen Datenbanken in der Regel um proprietäre Lösungen.¹⁰⁶

Abbildung 32 veranschaulicht die drei dargestellten Architekturvarianten.

¹⁰⁴ Vgl. Totok (2000), S. 66ff.; Haberstock (2000), S. 65ff.

¹⁰⁵ Beim multidimensionalen OLAP können die benötigten Speicherplätze als Produkt aller Ausprägungen jeder möglichen Dimension im Hyperwürfel berechnet werden.

¹⁰⁶ Vgl. Haberstock (2000), S. 66.

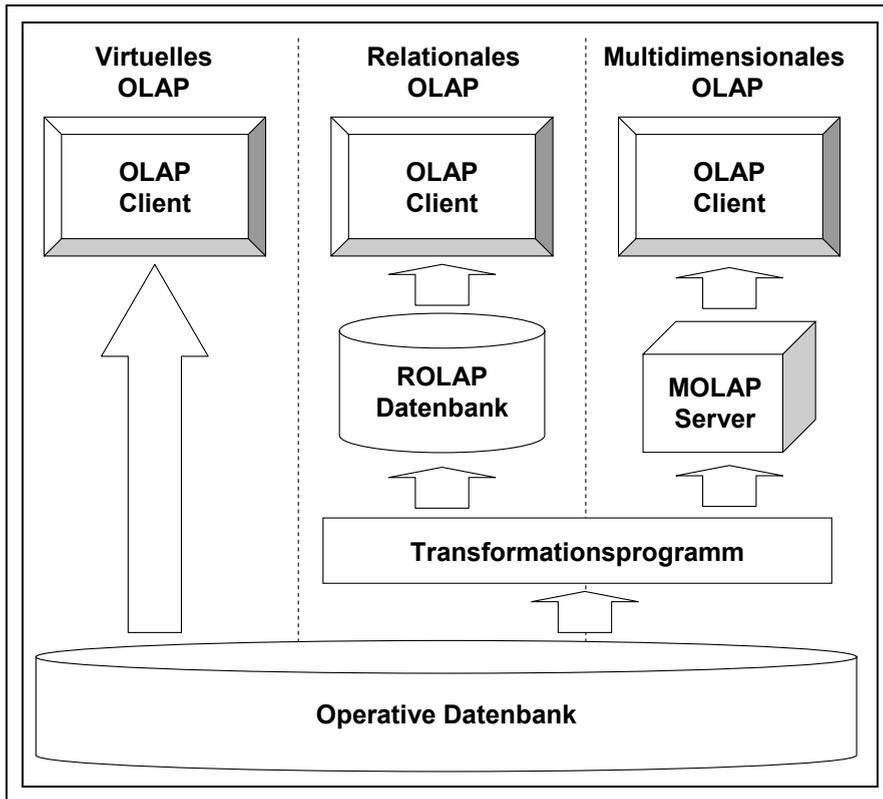


Abbildung 32: Architekturvarianten von OLAP-Systemen

Vergleicht man die aufgezeigten Varianten hinsichtlich des Antwortzeitverhaltens und des Speicherplatzbedarfs für die Datenbank, der zusätzlich zur Haltung der Unternehmensdaten anfällt, so ergibt sich folgender Zusammenhang.

Betrachtet man zuerst das Antwortzeitverhalten, so werden beim virtuellen OLAP und beim relationalen OLAP so genannte Binärbaumverfahren für den Datenzugriff in relationalen Datenbanken verwendet. Das Antwortzeitverhalten ist jedoch bei relationalen OLAP aufgrund der ergänzten Metainformationen besser als beim virtuellen OLAP. Beim multidimensionalen OLAP hingegen werden alle möglichen Dimensionselemente in den Schnittpunkten der Dimensionen vorgehalten, so dass aufgrund dieser Speicherungsform der Datenzugriff über Hashing Verfahren realisiert werden kann, deren Laufzeit deutlich besser als bei Binärsuchverfahren ist. Somit wird beim multidimensionalen OLAP das beste Antwortzeitverhalten erreicht.¹⁰⁷

In Bezug auf den zusätzlich benötigten Speicherplatz lässt sich eine gegensätzliche Rangfolge der Architekturen festlegen. Beim virtuellen OLAP wird

¹⁰⁷ Vgl. Oehler (2000), S. 126f.

kein zusätzlicher Speicherplatz benötigt, da die Daten um keine zusätzlichen Metainformationen angereichert werden. Beim relationalen OLAP wird zusätzlicher Speicherplatz für eben diese Metainformationen benötigt, so dass sich ein höherer Speicherplatzbedarf als beim virtuellen OLAP ergibt. Beim reinen multidimensionalen OLAP ist der zusätzliche Speicherplatzbedarf aufgrund des Vorhaltens sämtlich möglicher Werte in den Schnittstellen der Dimensionsausprägungen am größten. Darüber hinaus liegt dort ebenfalls eine ineffiziente Datenspeicherung vor, da Speicherzellen vorgehalten werden, die niemals mit einem gültigen Wert belegt werden können.¹⁰⁸

Zusammenfassend lässt sich festhalten, dass sich das Antwortzeitverhalten mit Inanspruchnahme von zusätzlichem Speicherplatz verbessert, die Speichernutzung jedoch beim multidimensionalen OLAP ineffizient ausfallen kann. Daher haben sich am Markt so genannte Hybrid Produkte (HOLAP) durchgesetzt, bei denen nur hochaggregierte Daten in multidimensionalen Datenbanken abgelegt werden und Daten niedriger Granularität in relationalen Datenbanken verbleiben.¹⁰⁹

5.3.2 Einsatzmöglichkeiten in virtuellen Unternehmen

Als Haupteinsatzgebiet von OLAP-Systemen lässt sich das Controlling von virtuellen Unternehmen ausmachen, da in diesem Fall eine Aggregation der vorliegenden Daten sinnvoll erscheint. Um jedoch den Einsatz innerhalb des Controllings zu analysieren, müssen zuerst einmal die anfallenden Controllingaufgaben aufgezeigt werden, um daran anschließend eine Architektur für ein OLAP-gestütztes Controlling zu diskutieren und für jeden Aufgabenbereich exemplarisch einzelne Konsolidierungspfade zu definieren, die durch OLAP-Systeme unterstützt werden können.

5.3.2.1 Controllingaufgaben in virtuellen Unternehmen

Legt man das planungs- und kontrollorientierte Controllingverständnis von Horváth zugrunde, so besteht die Aufgabe des Controllings in Netzwerken darin, neben der Sicherstellung einer Informationskongruenz als zentrale Aufgabe auch Koordinations- und Planungsfunktionen zur Unterstützung des Managements zu übernehmen.¹¹⁰ Dieses Controllingverständnis wird den weiteren Ausführungen wegen der hohen Akzeptanz in der Praxis zugrunde gelegt.

Als direktes Ziel des Controllings wird die Unterstützung des Managements durch Bereitstellung relevanter Informationen verfolgt. Bei diesen Managementaufgaben können bei virtuellen Unternehmen die auftragsbezogene Ebene und die auftragsübergreifende Ebene unterschieden werden. Hierbei

¹⁰⁸ Vgl. Gentsch (2001), S. 530.

¹⁰⁹ Vgl. Mucksch (1998), S. 432.; Schinzer / Bange / Mertens (1999), S. 53f.

¹¹⁰ Vgl. Horváth (1998).

werden auf der auftragsübergreifenden Ebene die Voraussetzungen für die Übernahme konkreter Aufträge auf der auftragsbezogenen Ebene geschaffen.

Betrachtet man die auftragsbezogene Ebene, so ist dort das Auftragscontrolling innerhalb von virtuellen Unternehmen anzusiedeln. Dabei ist die komplette Auftragsabwicklung, die bereits in Abschnitt zwei beschrieben wurde, vom Zeitpunkt der Auftragspezifikation über die Zwischenschritte Teilauftrag vergeben, Auftragspreis festlegen, Abwicklung abbilden bis hin zur Kontrolle der Abwicklung zu unterstützen und in geeigneter Weise zu kontrollieren. Dies kann beispielsweise bei der Auftragsabwicklung in Form von Fortschrittskontrollen, Soll-Ist-Vergleichen oder durch Analyse von geeigneten Kennzahlen erfolgen. Im Bereich der Preisfestsetzung kann die Preisentwicklung für einen Teilauftrag analysiert werden, wenn mehrere Aufträge vom selben Auftragsstyp durch das Netzwerk abgewickelt werden.¹¹¹

Auf der auftragsübergreifenden Ebene steht die Vorbereitung des auftragsbezogenen Handelns im Vordergrund der Controllingaufgaben. Hierbei lassen sich mit dem Erfolgs-, Aktions- und Partnercontrolling drei verschiedene Controllingbereiche voneinander abgrenzen.

Die Partnerunternehmen verfolgen durch das Mitwirken in einem virtuellen Unternehmen individuelle Ziele. Im Rahmen des Erfolgscontrollings stehen die Evaluierung der Partnerunternehmen hinsichtlich der individuellen Zielerreichung auf der einen Seite und die kollektive Zielerreichung auf der anderen Seite im Mittelpunkt der Betrachtung. In diesem Bereich beschränkt sich die derzeitige Unterstützung durch IV-Systeme lediglich auf die Dokumentation der aufgestellten Ziele und der entsprechenden Zielerreichungsgrade.¹¹²

Im Partnercontrolling werden die Entscheidungen des Managements bzgl. der Akquirierung neuer Partner für das virtuelle Unternehmen vorbereitet. Dafür muss innerhalb des Partnercontrollings geprüft werden, ob die Aufnahme eines neuen Partners einen Zusatznutzen für das Netzwerk stiftet. Dieser wird vor allem durch die Beurteilung der Kernkompetenzen und der Kapazitäten des neuen Partners bestimmt. Dabei sollen durch die Akquirierung neuer Partner entweder die Gesamtzahl der verfügbaren Kompetenzen im Netzwerk erweitert oder aber bestehende Kapazitätsengpässe beseitigt werden. In diesem Bereich beschränkt sich die Werkzeugunterstützung auf die Dokumentation und Auswertung partnerbezogener Daten.¹¹³

Im Aktionscontrolling wird das Management bei der strategischen Planung des Netzwerks im engeren Sinne unterstützt. Dabei liegt die Hauptaufgabe in der Identifikation und Analyse von Kooperationsfeldern sowie im Vorbereiten von Aktionen in einem ausgewählten Kooperationsfeld. In diesem Bereich beschränkt

¹¹¹ Vgl. Hess / Schumann (2000a), S. 411ff.

¹¹² Vgl. Hess / Zieger (2000), S. 4ff.

¹¹³ Vgl. Hess / Schumann (2000b), S. 68ff.

sich die Werkzeugunterstützung auf die Dokumentation partnerbezogener Aktivitäten innerhalb einzelner Kooperationsfelder.

Abbildung 33 stellt die Controllingaufgaben auf auftragsbezogener und auftragsübergreifender Ebene dar. Dabei durchlaufen alle vier Controllingbereiche den typischen Planungs-Abbildungs-Kontroll-Zyklus. Auch wird die Vernetzung der einzelnen Controllingbereiche durch die angedeuteten Verbindungen zum Ausdruck gebracht.¹¹⁴

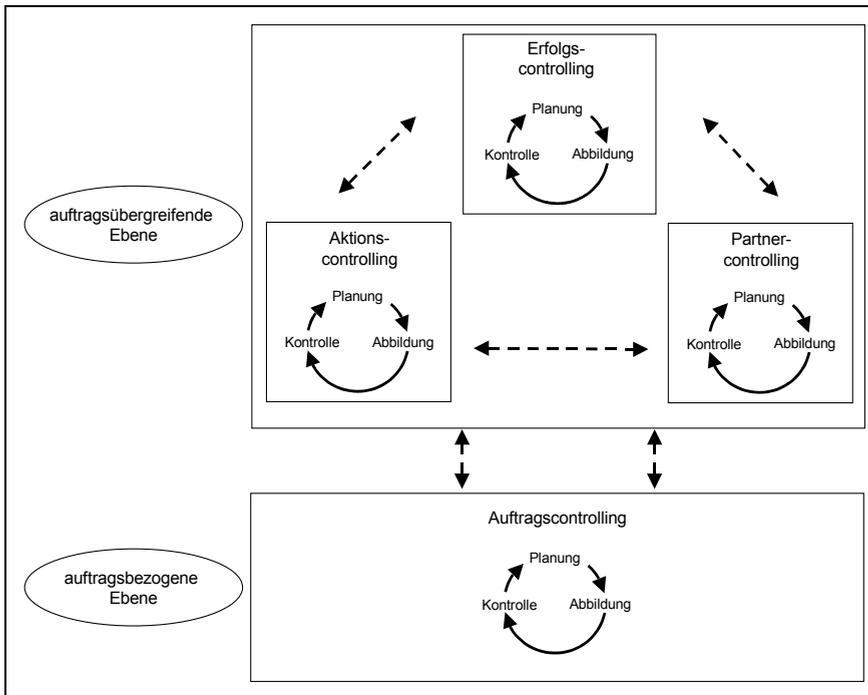


Abbildung 33: Controllingaufgaben in virtuellen Unternehmen

5.3.2.2 Architektur eines OLAP-gestützten Controlling-systems

Für den Einsatz von OLAP-Systemen muss in einem ersten Schritt die zugrunde liegende Datenbasis geschaffen werden. Geht man davon aus, dass innerhalb eines virtuellen Unternehmens keine Konsistenzprobleme vorliegen und auch die Daten innerhalb des Netzwerks semantisch eindeutig sind, kann auf die Zwischenschaltung eines Data Warehouse verzichtet werden.

Sollten diese beiden Annahmen nicht zutreffen, so muss die semantische Eindeutigkeit und die Konsistenz durch ein zwischengeschaltetes Data Warehouse erzeugt werden, indem die Datenbasis der Partnerunternehmen auf diese zentrale Datenbasis innerhalb des Data Warehouse durch geeignete Transformationsregeln

¹¹⁴ Vgl. Hess (2002b), S. 26.

überführt wird. Auch könnte diese semantische Eindeutigkeit und Konsistenz der Daten durch ein zentral betriebenes IV-System gewährleistet werden, welches die benötigten Funktionalitäten wie u. a. Auftragsvergabe und Auftragsüberwachung auf auftragsbezogener Ebene bereitstellt und auf einer zentralen relationalen Datenbank die netzwerkspezifischen Daten für die auftragsbezogene und auftragsübergreifende Ebene persistent speichert.

Aus diesem relationalen Vorsystemen würde ein Transformationsprogramm anhand von definierten Transformationsregeln alle benötigten Daten extrahieren und in einer geeigneten Form bei Bedarf aggregieren, um eine zügige Auswertung durch die OLAP-Systeme zu ermöglichen. Bei Bedarf können die Daten optional um entsprechende Metainformationen angereichert werden, die die Laufzeit von multidimensionalen Auswertungen verbessern. Dieser Datenexport in die Basisdatenbank für die Auswertungen unter Nutzung von OLAP könnte dabei periodisch oder aber bei wichtigen Veränderungen der Netzwerkstruktur durchgeführt werden.

Dieser hier aufgezeigte Lösungsvorschlag stellt eine Mischung der dargestellten Architekturkonzepte zwischen virtuellen und relationalen OLAP dar, wobei der Gedanke des relationalen OLAP nur eine optionale Komponente einnimmt. Auch wird durch die Architektur das operative System durch die Auswertungsanfragen nicht beeinträchtigt. Im Gegensatz zum ebenfalls im vorigen Abschnitt diskutierten Lösungsansatz unter Verwendung eines Data Warehouse muss nur eine Schnittstelle zwischen dem relationalen Vorsystem und der Basisdatenbank realisiert werden, wodurch der Pflegeaufwand der Schnittstellen minimiert wird.

Abbildung 34 veranschaulicht diese Architektur graphisch.

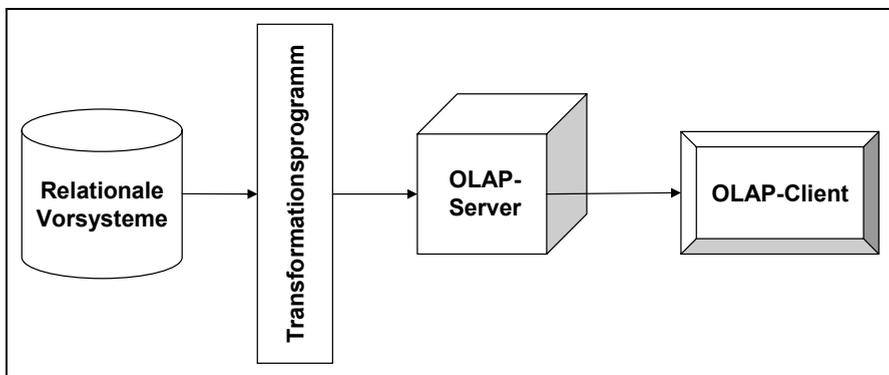


Abbildung 34: OLAP-Architektur für das Controlling in virtuellen Unternehmen

5.3.2.3 Unterstützungslleistung innerhalb der Controllingaufgaben

Da durch die Verwendung von OLAP-Systemen eine multidimensionale Sicht auf die Daten bereitgestellt wird, die durch Aggregation der operativen Datenbestände

entlang von Konsolidierungspfaden erzeugt wird, bietet sich die Datenanalyse in dieser Form lediglich auf der auftragsübergreifenden Ebene an. Auf der auftragsbezogenen Ebene im Rahmen des Auftragscontrollings ist kein sinnvoller Einsatz von OLAP-Systemen erkennbar.

Betrachtet man daher die auftragsübergreifende Ebene, so bietet sich nun eine Darstellung der Unterstützungsleistung anhand der drei identifizierten Controllingbereiche des Erfolgs-, des Partner- und des Aktionscontrollings an. Hier müssen beispielhaft mögliche Konsolidierungspfade verschiedener Dimensionen aufgezeigt werden, die dann ihrerseits den Ergebniswürfel einer Auswertung aufspannen.

Im Rahmen des Erfolgscontrollings werden, wie bereits dargestellt, die Zielerreichungsgrade der individuellen und der kollektiven Ziele analysiert. Um den Erfolg von Unternehmensnetzwerken zu analysieren, bietet es sich an, die partnerspezifischen Ziele zu gemeinsamen Oberzielen innerhalb des Netzwerks zu aggregieren und um spezifische Ziele des Netzwerks anzureichern. Für diese gemeinsame Zielmenge kann dann anhand des Zielerreichungsgrades der Erfolg des Netzwerks gemessen werden. Bei der Erfolgsmessung werden in klassischen Unternehmen in der Regel monetäre Kennzahlen gebildet, die aus den entsprechenden internen Daten u. a. des Rechnungswesens abgeleitet werden. Dabei wird dann beispielsweise die Differenz aus Kosten und Erlösen als Kennzahl für die Bewertung des Netzwerkerfolgs herangezogen. Jedoch tritt bei der Erfassung der Kosten eine besondere Problematik in Netzwerken auf, da die für die Teilaufträge veranschlagten Kosten nicht immer den tatsächlichen Kosten des Partnerunternehmens entsprechen, sondern um eine Risikoaufschlag angereichert sind. Aus dieser Sicht bietet sich diese Kennzahl der Differenz aus Kosten und Erlösen nur bedingt als Maßstab für die Erfolgsbewertung an. Jedoch reicht eine Aufstellung lediglich der Erlöse in aggregierter Form bereits zur Bewertung des Netzwerkerfolgs aus und kann für Informationszwecke bei der Akquisition von neuen Partnerunternehmen eingesetzt werden.

Man erkennt aus diesen Ausführungen, dass innerhalb des Erfolgscontrollings spezielle Probleme auftreten, die auf die Netzwerkstruktur zurückzuführen sind und daher an dieser Stelle nicht weiter diskutiert werden sollen. Es lassen sich jedoch standardisierte Kennzahlen wie u. a. „Anteil am Netzwerkerlösen“, „Umsatzsatzwachstum“ und „absolute Netzwerkerlöse“ für virtuelle Unternehmen erstellen, durch die beispielsweise die Partnerunternehmen, die Kooperationsfelder oder die Auftragstypen in eine Rangfolge gebracht werden können. Durch die Nutzung dieser Kennzahlen lassen sich dann eine Vielzahl von Fragestellungen wie beispielsweise „Wie groß waren die Erlöse im Kooperationsfeld A im Jahr 2000?“ beantworten. Bei diesen hoch aggregierten Kennzahlen wird der Einsatz von OLAP-Systemen sinnvoll.

Nachfolgendes Beispiel soll eine multidimensionale Sicht auf die absoluten Netzwerkerlöse verdeutlichen. Dabei wird die Sicht durch die Dimensionen

„Netzwerkhierarchie“, „Zeit“ und „Szenario“ aufgespannt. Der Dimension „Netzwerkhierarchie“ kann ein Konsolidierungspfad aus Sicht der Aufträge über den Teilauftrag, den Auftrag, den Auftragsstyp, den Kooperationsfeld und als höchste Hierarchiestufe das gesamte Netzwerk oder aber ein partnerbezogener Konsolidierungspfad über den Teilauftrag, die Partnerunternehmen und das gesamte Netzwerk zugrunde liegen. Bei der Dimension „Zeit“ kann der gewählte Pfad über Tag, Monat, Quartal und Jahr verlaufen. Die Dimension „Szenario“ besitzt keinen Konsolidierungspfad, sondern hält nur die beiden Ausprägungen „Soll“ und „Ist“ vor.

Abbildung 35 veranschaulicht diese multidimensionale Sicht auf die absoluten Netzwerkerlöse graphisch.

Netzwerkhierarchie	Zeit			
	1999	2000	2001	2002
Kooperationsfeld A	802	920	865	999
Kooperationsfeld B	802	1021	1234	1423
Kooperationsfeld C	342	123	50	0
Kooperationsfeld D	123	345	454	800

Abbildung 35: Beispiel einer multidimensionalen Sichtweise auf die Netzwerkerlöse

Im Rahmen des Partnercontrollings werden Informationen bereitgestellt, welche die Auswahl bzw. Gewinnung neuer Partner für das Netzwerk unterstützen. Dabei ist der Nutzenszuwachs durch die Aufnahme des neuen Partners zu beurteilen, der sich hauptsächlich durch die hinzugefügten Kernkompetenzen und Kapazitäten des potentiellen Partners ergibt. Es werden Partner hinzugenommen, wenn durch diese Kapazitätsengpässe beseitigt bzw. neue Kernkompetenzen dem

Partnerpool hinzugefügt werden können. In diesem Bereich lassen sich dann auch Fragestellungen wie „Liegt ein Monopol für eine Kernkompetenz durch ein Partnerunternehmen vor?“ generieren, die durch entsprechende Kennzahlen wie „Anzahl der Partnerunternehmen mit bestimmter Kernkompetenz“ beantwortet werden können. In diesem Bereich bieten sich dann für die multidimensionale Sicht Dimensionen wie „Kernkompetenz“, „Zeit“ oder „Kapazität“ an. Dabei könnte die Dimension „Kernkompetenz“ den Konsolidierungspfad „Teilkompetenz“ zu „Hauptkompetenz“ aufweisen und die Dimension „Zeit“ den bereits dargestellten Verdichtungspfads beibehalten. Durch die Aufspaltung der Kompetenzen in Teil- und Hauptkompetenzen ist eine differenziertere Betrachtung der Kompetenzen möglich. Die Dimension „Kapazität“ könnte analog zur Dimension „Szenario“ die beiden Ausprägungen „Kapazitätsnachfrage“ und „Kapazitätsangebot“ besitzen. Aus diesen Dimensionen lässt sich dann wieder ein Würfel aufbauen, der mit entsprechenden Ausprägungen der Kennzahlen besetzt ist.

Im letzten Controllingbereich des Aktionscontrollings wird die strategische Planung des Netzwerks im engeren Sinne vorbereitet. Die Hauptaufgabe liegt in der Planung und späteren Erfolgsbewertung der einzelnen Kooperationsfelder.¹¹⁵ Im klassischen Unternehmen werden dazu Portfoliomodelle verwendet, um die Entwicklung von Geschäftsfeldern, die den Kooperationsfeldern entsprechen, zu planen. Dabei könnten beispielsweise die generierten Erlöse auf den einzelnen Kooperationsfeldern im Zeitablauf gegenübergestellt werden, um die Fragestellung nach der zeitlichen Entwicklung der Erlöse auf den einzelnen Kooperationsfeldern zu beurteilen. Auch können durch eine Analyse der Auftragsarten, die das Marktangebot des Netzwerks determinieren, weitere Kooperationsfelder aufgedeckt werden. Die notwendigen multidimensionalen Sichten können aus den beiden bereits dargestellten Controllingbereichen durch Kombination aufgebaut werden, so dass eine konkrete Darstellung einer multidimensionalen Sicht entfallen kann. Neben der hier aufgeführten möglichen Dimensionen sind natürlich noch eine Vielzahl weiterer Definition möglich, die sich aus der Art der entsprechenden Fragestellungen und der zu ihrer Beantwortung benötigten Kennzahlen ergeben.

5.3.3 Beurteilung

Durch den Einsatz von OLAP-Systemen können die Entscheidungen des virtuellen Unternehmens zumindest bezüglich der Datenbasis und bezüglich der Analyseseite der Daten auf ein gesichertes Fundament gestellt werden. Die Einsatzgebiete lassen sich hierbei in die drei Bereiche der „Kennzahlenberechnung“, der „Ursachenanalyse“ und der „Dokumentation der Zielerreichungsgrade“ einteilen.

¹¹⁵ Vgl. Hess / Schumann (2000a), S. 12ff.

Die Ausführungen haben gezeigt, dass vor allem die Berechnung von hoch aggregierten Kennzahlen durch die Verwendung von OLAP-Systemen unterstützt und vereinfacht wird. Die Kennzahlen werden anhand vorgegebener Konsolidierungspfade zur Laufzeit der Auswertung berechnet, wobei die Berechnung entweder auf den operativen Originaldaten aufsetzt, oder aber auf einem Datenbestand, der bereits Daten in voraggregierter Form besitzt, die durch den Transformationsprozess erzeugt wurden.

Auch die Ursachenanalyse wird durch OLAP-Systeme durch so genannte „Drill down“ und „Drill up“ Funktionalitäten unterstützt, die entlang der Konsolidierungspfade die Navigation im Ergebniswürfel ermöglichen. Diese Möglichkeit der Navigation wird standardmäßig durch OLAP-Systeme unterstützt und erfordert keine Anpassung der Systeme. Somit liegt hier ein Zusatznutzen von OLAP-Systemen vor.

Auch die Dokumentation der Zielerreichungsgrade, die bisher lediglich über Werkzeuge unterstützt wird, wird durch OLAP-Systeme realisiert. Die Bestimmung der Zielerreichungsgrade kann jedoch jederzeit erneut erfolgen, sollte sich die zugrunde liegende Datenbasis verändert haben.

Insgesamt lässt sich der Einsatz von OLAP-Systemen in allen drei Controllingbereichen befürworten, jedoch muss durch eine differenzierte Kosten/Nutzen-Betrachtung eine endgültige Entscheidung herbeigeführt werden.

Im konkreten Entwicklungsprojekt wurde trotz einer grundsätzlichen Eignung eines OLAP-Systems auf die Implementierung verzichtet, da die für das Controlling zu erwartenden Nutzenpotenziale weit hinter den Kosten für eine Konzeption und Umsetzung eines OLAP-Systems zurückblieben.

6 Zusammenfassung und Ausblick

Das Ziel dieser Veröffentlichung bestand darin, die fachlichen Anforderungen an ein Werkzeug zur Unterstützung des Managements virtueller Unternehmen herzu-leiten und aufbauend auf diese Anforderungen das Werkzeug prototypisch umzu-setzen. Ergänzend zur eigentlichen Werkzeugentwicklung war aus technischer Sicht weiterhin zu analysieren, ob virtuelle Unternehmen durch innovative Tech-nologien, wie XML, Web Services und OLAP, sinnvoll unterstützt werden können.

Um diese Zielsetzung zu erfüllen wurden im zweiten Kapitel zunächst virtuelle Unternehmen vorgestellt und die grundlegende Bedeutung der Informations-technologie für virtuelle Unternehmen herausgearbeitet.

Im dritten Kapitel wurden die wesentlichen Anforderungen an eine Werkzeug-unterstützung für virtuelle Unternehmen untersucht. Hierzu wurden nach einer Systematisierung von Aufgaben, Instrumenten und Werkzeugen sowie der Analyse der Basisanforderungen an Informationstechnologie in virtuellen Unter-nehmen die Anforderungen an die Werkzeugunterstützung anhand eines Phasen-schemas hergeleitet. Zusätzlich wurden bereits verfügbare Werkzeuge zur Abdeckung der Anforderungen beschrieben.

Das vierte Kapitel stellte die Entwicklung des prototypischen Werkzeugs "VICOPLAN" ausführlich dar. Zunächst wurden zu diesem Zweck die Rahmenbedingungen und der Entwicklungshintergrund beschrieben. Ausgehend von der Auswahl eines geeigneten Vorgehensmodells zur Softwareentwicklung wurde schließlich die fachliche Konzeption und technische Umsetzung des Prototyps dargelegt. Eine Übersicht über die Systemfunktionen von VICOPLAN bildete den Abschluss des vierten Kapitels.

Im fünften Kapitel wurden ausgewählte innovative Technologien hinsichtlich ihrer Einsatzmöglichkeiten in virtuellen Unternehmen untersucht. Als Ergebnis der Analyse von XML, Web Services und OLAP kann festgehalten werden, dass

die Technologien grundsätzlich große Unterstützungspotenziale bieten, ihr Einsatz jedoch aus einer Kosten-Nutzen-Sicht im Einzelfall kritisch hinterfragt werden sollte.

Diese Veröffentlichung soll Praktiker und Wissenschaftler bei der Umsetzung von Werkzeugen für virtuelle Unternehmen unterstützen. Um diese Unterstützung zu gewährleisten wird das Projekt „VICOPLAN“ auch nach dem offiziellen Projektende weiter verfolgt.

VICOPLAN steht so zum Beispiel als Prototyp allen interessierten virtuellen Unternehmen kostenlos zur Verfügung. Über die Website www.vicoplan.de werden eine Online-Testversion, ein Kurzhandbuch sowie Literatur zum Thema bereitgestellt.

VICOPLAN wird auch über die Projektlaufzeit hinweg in der Forschungsgruppe „NetWorX“ am Institut für Wirtschaftsinformatik und Neue Medien der Universität München fachlich weiterentwickelt. Über ein Engagement des Münchner Instituts auf der Webplattform www.vforum.de (spezielle Community zu virtuellen Unternehmen) im Bereich Kooperations-Tools wird auch die wissenschaftliche Community über die entstandenen Publikationen hinaus über VICOPLAN informiert.

Über das Spin-Off „Netvolution GmbH“ (www.netvolution.de) des Instituts für Wirtschaftsinformatik und Neue Medien der Universität München wird das mit VICOPLAN aufgebaute Know-How an andere Unternehmen weitergegeben. VICOPLAN wird dabei grundsätzlich kostenlos abgegeben und es werden lediglich die unter Umständen anfallenden Einführungs- bzw. Customizingleistungen in Rechnung gestellt.

Literaturverzeichnis

- Adler, S. / Berglund, A. / Caruso, J. / Deach, S. / Graham, T. / Grosso, P. / Gutentag, E. / Milowski, A. / Parnell, S. / Richman, J. / Zilles, S. (2001): Extensible Stylesheet Language (XSL) Version 1.0, <http://www.w3.org/TR/xsl/>, Abruf am 30.04.2003.
- Austin, D. / Barbir, A. / Ferris, C. / Garg, S. (2002): Web Services Architecture Requirements, <http://www.w3.org/TR/2002/WD-wsa-reqs-20021011>, Abruf am 21.05.2003.
- Bach, M. (2000): XSL und XPath - verständlich und praxisnah: Transformationen und Ausgabe von XML-Dokumentationen mit XSL, Boston 2000.
- Balzert, H. (2000): Lehrbuch der Software-Technik, 2. Auflage, Heidelberg.
- Balzert, H. (1996): Objektorientierte Systemanalyse, Heidelberg.
- Behme, W. (1995): Virtuelle Unternehmen. In: Zeitschrift für Planung 6 (1995) 6, S. 297-230.
- Behme, H. / Mintert, S. (2001): XML in der Praxis: professionelles Web-Publishing mit der Extensible Markup Language, 2, München [u.a.] 2001.
- Beimborn, D. / Mintert, S. / Weitzel, T. (2002): WI - Schlagwort - Web Services und ebXML. In: Wirtschaftsinformatik 44 (2002) 3, S. 277-280.
- Bettag, U. (2001): Web Services. In: Informatik Spektrum 5 (2001) 24, S. 302-304.
- Biethahn, J. / Muksch H. / Ruf, W. (2000): Ganzheitliches Informationsmanagement, 3. unwes. veränd. Auflage, München.
- Boag, S. / Chamberlin, D. / Fernandez, M. F. / Florescu, D. / Robie, J. / Siméon, J. (2002): XQuery 1.0: An XML Query Language, <http://www.w3.org/TR/xquery/>, Abruf am 30.04.2003.
- Boehm, B. W. (1986): Wirtschaftliche Software-Produktion, Wiesbaden.
- Boehm, B.W. (1988): A Spiral Model for Software Development and Enhancement, in: IEEE Computer Volume 21 (Mai 1988), S. 61-72.

- Box, D. / Skonnard, A. / Lam, J. F. (2000): Essential XML: beyond markup, Boston [u.a.] 2000.
- Bray, T. / Hollander, D. / Layman, A. (1999): Namespaces in XML, <http://www.w3.org/TR/REC-xml-names/>, Abruf am 30.04.03.
- Bray, T. / Paoli, J. / Sperberg-McQueen, C. M. / Maler, E. (2000): Extensible Markup Language (XML) 1.0, <http://www.w3.org/TR/REC-xml>, Abruf am 30.04.2003.
- Brehm, B. (1997): Strategisches Management von Electronic Data Interchange (EDI), Göttingen 1997.
- Bremer, G.(1998): Genealogie von Entwicklungsschemata, in: Kneuper, R. / Müller-Luschnat, G. / Oberweis A. (Hrsg.), Vorgehensmodelle für die betriebliche Anwendungsentwicklung, Stuttgart.
- Budde, R. / Kautz, K. /Kuhlenkamp, K. /Züllighoven, H. (1992): Prototyping – An approach to evolutionary System Development, Berlin.
- Burghardt, M. / Gehrke, N. / Schumann, M. (2003): Implikationen kommerzieller Web Services, In: Eckstein, R. / Tolksdorf, R.: XMIDX 2003 - XML-Technologien für Middleware - Middleware für XML-Anwendungen, Köln 2003, S. 71-82.
- Buse, H. P. (1997): Wandelbarkeit von Produktionsnetzen - Auswirkungen auf die Gestaltung des interorganisatorischen Logistiksystems, In: Dangelmeier, W.: Vision Logistik: Logistik wandelbarer Produktionsnetze, Paderborn 1997, S. 71-139.
- Cerami, E. (2002): Web services essentials, Beijing [u.a.] 2002.
- Christensen, E. / Curbera, F. / Meredith, G. / Weerawarana, S. (2002): Web Services Description Language (WSDL) 1.1..., <http://www.w3.org/TR/wsdl>, Abruf am 25.06.2002.
- Clark, J. / DeRose, S. (1999): XML Path Language (XPath) Version 1.0, <http://www.w3.org/TR/xpath>, Abruf am 30.04.2003.
- Coad, P. / Yourdon, E. (1991): Object-Oriented Analysis, 2. Auflage, New Jersey.
- Coldewey, J. (2003): Agile Entwicklung – ein Überblick, in HMD Heft 231, 2003, S. 46-54.
- Coldewey, J. (2002): Agile Entwicklung Webbasierter Systeme: Einführung und Überblick, in: Wirtschaftsinformatik 44/2002, S. 237-248.
- Ebers, M. / Hermsch, M. / Matzke, M. / Mehlhorn, A. (2000): Strukturwandel und Steuerungsformen von Netzwerken in der deutschen Bauindustrie, In: Sydow, J. / Windeler, A.: Steuerungsformen von Netzwerken, Opladen 2000, S. 251-279.
- Eddy, S. E. / Delong, B. (2001): XML in plain English, Foster City, Calif. [u.a.] 2001.
- Eicker, S. /Nietsch, M. (1999): Standards zum objektorientierten Paradigma, in: Wirtschaftsinformatik 41/1999, S. 358-370.

- End, W. / Gotthardt, H. / Winkelmann, R. (1990): Softwareentwicklung, Leitfaden für Planung, Realisierung und Einführung von DV-Verfahren, 7. wesentl. überarb. Auflage, Berlin.
- Fischer, T. / Biskup, H. / Müller-Luschnat, G. (1998): Begriffliche Grundlagen für Vorgehensmodelle, in: Kneuper, R. / Müller-Luschnat, G. / Oberweis A. (Hrsg.), Vorgehensmodelle für die betriebliche Anwendungsentwicklung, Stuttgart.
- Fischer, J. (1993): Unternehmensübergreifende Datenmodellierung - der nächste folgerichtige Schritt der zwischenbetrieblichen Datenverarbeitung. In: Wirtschaftsinformatik 35 (1993) 3, S. 241-254.
- Fülbier, D. / Binnig, C. / Schmidt, A. (2003): Turbine: Ein Java-Framework für die Entwicklung von Webanwendungen, in iX: Magazin für Professionelle Informationstechnik 11/2003, S. 118-121.
- Gentsch, P. (2001): Wissenserwerb in Innovationsprozessen: Methoden und Fallbeispiele für die informationstechnologische Unterstützung, Wiesbaden 2001.
- Genussa, P. (1999) Evolution and Use of Generic Markup Languages, In: Möhr, W.: SGML und XML: Anwendungen und Perspektiven, Berlin [u.a.] 1999, S. 27-50.
- Goldfarb, C. F. / Prescod, P. (2002): XML handbook, Upper Saddle River, NJ 2002.
- Haberstock, P. (2000): Executive Information Systems und Groupware im Controlling: Integration durch das prozessorientierte Team-Controllingsystem (ProTeCos), Wiesbaden 2000.
- Hallmann, M. (1990): Prototyping komplexer Softwaresysteme: Ansätze zum Prototyping und Vorschlag einer Vorgehensweise, Stuttgart.
- Harold, E. R. (2001): XML bible, New York, NY [u.a.] 2001.
- Hedberg, B. / Dahlgren, G./Hansson, J./Nils-Göran, O. (1997): Virtual Organizations and Beyond, New York u.a.
- Heib, R. (1998): Business Process Reengineering mit ARIS-Modellen, in: ARIS – Vom Geschäftsprozess zum Anwendungssystem, 3. völlig neubearbeitete und erw. Auflage, Berlin.
- Hess, T. (1999): ZP-Stichwort: Unternehmensnetzwerke. In: Zeitschrift für Planung 10 (1999) 2, S. 225-230.
- Hess, T. (2001): Controlling eines virtuellen Unternehmens. In: Kostenrechnungspraxis, Sonderheft E-Business & Controlling (2001) S. 92-100.
- Hess, T. (2002a): Netzwerkcontrolling: Instrumente und ihre Werkzeugunterstützung, Wiesbaden 2002. (a)
- Hess, T. (2002b): Planning and Control of Virtual Corporations in the Service Industry: The Prototype VICOPLAN, 35th Annual Hawaii International Conference on System Sciences, 2002, S. 24-33. (b)

- Hess, T. / Schumann, M. (2000a): Auftragscontrolling in Unternehmensnetzwerken. In: Zeitschrift für Planung 11 (2000) S. 411-432. (a)
- Hess, T. / Schumann, M. (2000b): Werkzeuggestütztes Controlling in Unternehmensnetzwerken: Stand und Entwicklungsperspektiven, In: Verbundtagung Wirtschaftsinformatik 2000: Aachen 2000, S. 68-85. (b)
- Hess, T. / Zieger, M. (2000): Werkzeugunterstützung für das Controlling virtueller Unternehmen: Das System VICOPLAN, In: Engelen, M. / Neumann, D.: Virtuelle Organisation und Neue Medien, Köln 2000, S. 188-198.
- Hesse, W. (1997): Improving the software process guided by the EOS model, in: Proc. SPI '97 European Conference on Software Process Improvement, Barcelona.
- Hesse, W. (1998): Vorgehensmodelle für objektorientierte Software-Entwicklung, in: Kneuper, R. / Müller-Luschnat, G. / Oberweis A. (Hrsg.), Vorgehensmodelle für die betriebliche Anwendungsentwicklung, Stuttgart.
- Hoidn, H. P. / Jungclaus, R. (2002): Web Services aus Sicht der Unternehmensarchitektur. In: Information Management & Consulting 17 (2002) 3, S. 31-35.
- Holthuis, J. (1999): Der Aufbau von Data Warehouse-Systemen: Konzeption, Datenmodellierung, Vorgehen, 2, Wiesbaden 1999.
- Holzner, S. (2002): XSLT - Anwendung und Referenz: XML-Transformationen, XPath, Einsatz mit Java, JSP und ASP, München 2002.
- Horváth, P. (1998): Controlling, 7, München 1998.
- Köhler, H.-D. (2000): Netzwerksteuerung und/ oder Konzernkontrolle: Die Automobilkonzerne im Internationalisierungsprozeß, in: Sydow, J./Windeler, A. H. (Hrsg., 2000): Steuerung von Netzwerken, Wiesbaden, S. 281-300.
- Kreger, M. (2001): Web Services Conceptual Architecture, <http://www-4.ibm.com/software/solutions/webservices/WSCA.pdf>, Abruf am 20.05.2001.
- Jacobson I. /Booch, G. /Rumbaugh, J. (2000): The Unified Software Development Process, 3. Auflage, Boston.
- Kocher, H.: (1998): Management objektorientierter Projekte, in: OBJEKTspektrum 3/1998, S. 65-70.
- Lang, K. W. (2001): Virtuelle Unternehmen, In: Noack, U. / Spindler, G.: Unternehmensrecht und Internet, München 2001, S. 169-198.
- Langner, P. / Schneider C. / Wehler, J. (1997) Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen, in: Wirtschaftsinformatik 39/1997, S. 479-489.
- Laurent, S. S. (2001): XML: a primer, New York, NY [u.a.] 2001.
- Light, R. (1997): Presenting XML, Indianapolis, Ind. 1997.

- Linn, N. (1989): Die Implementierung vertikaler Kooperationen: theoretische Konzeption und erste empirische Ergebnisse zum Prozess der Ausgliederung logistischer Teilaufgaben, Frankfurt am Main [u.a.] 1989.
- Löwer, U. M. / Picot, A. (2002): Web Services - Technologie - Hype oder Strategie-Faktor. In: Information Management & Consulting 17 (2002) 3, S. 20-25.
- McLaughlin, B. / Kersken, S. / Key, J. (2002): Java XML, Dt. Ausg. der 2. Aufl, Beijing [u.a.] 2002.
- Mertens, P. /Bodendorf, F. /König, W. /Picot, A. /Schumann, M. (2001): Grundzüge der Wirtschaftsinformatik, 7. neu bearbeitete Auflage, Berlin.
- Mertens, P. / Faisst, W. (1996): Virtuelle Unternehmen: eine Organisationsstruktur für die Zukunft?. In: Wirtschaftswissenschaftliches Studium 25 (1996) 6, S. 280-285.
- Mertens, P. / Griese, J. / Ehrenberg, D. (1998): Virtuelle Unternehmen und Informationsverarbeitung: mit 8 Tabellen, Berlin [u.a.] 1998.
- Merz, M. (2002): E-Commerce und E-Business: Marktmodelle, Anwendungen und Technologien, 2, Heidelberg 2002.
- Michel, T. (1999): XML kompakt - Struktur, Layout, Verweise: eine praktische Einführung, Wien 1999.
- Miles, R. E. / Snow, C. C. (1986): Organization: New Concepts for New Forms. In: California Management Review 28 (1986) 3, S. 62-73.
- Morgenthal, J. P. / Forge, B. L. (2001): Enterprise application integration with XML and Java, Upper Saddle River, NJ [u.a.] 2001.
- Mucksch, H. (1998): Das Data Warehouse-Konzept: Architektur, Datenmodelle, Anwendungen, 3, Wiesbaden 1998.
- Müller, D. (2002): IT-Werkzeuge für das Management virtueller Unternehmen, Kiel.
- Netzer, F. (1999): Strategische Allianzen im Luftverkehr: Nachfrageorientierte Problemfelder ihrer Gestaltung, Frankfurt am Main.
- Neumann, H. (2002): Analyse und Entwurf von Softwaresystemen mit der UML, München.
- Neuburger, R. (1994): Electronic Data Interchange: Einsatzmöglichkeiten und ökonomische Auswirkungen, Wiesbaden 1994.
- Noack, J. (2002): Eine Werkbank für den Zuschnitt von objektorientierten Softwareprozessen, in: Wirtschaftsinformatik 44/2002, S. 315-324.
- o.A. (2003): Principles behind the Agile Manifesto, <http://www.agilemanifesto.org/principles.html>, Abruf am 2003-10-08.
- o.A. (2003a): Introduction To OMG's Unified Modeling Language (UML), http://www.omg.org/gettingstarted/what_is_uml.htm, Abruf am 2003-10-27.

- o.A. (2003b): XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition), A Reformulation of HTML 4 in XML 1.0, W3C Recommendation 26 January 2000, re-vised 1 August 2002, <http://www.w3.org/TR/xhtml1/>, Abruf am 2003-12-03.
- o.A. (2003c): Internet-Standard RFC2616 – „Hypertext Transfer Protocol -- HTTP/1.1“, <ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt>, Abruf am 2003-12-03.
- o.A. (2003d): Internet-Standard RFC1321 – „The MD5 Message-Digest Algorithm“, <ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt>, Abruf am 2003-12-22.
- Oehler, K. (2000): OLAP: Grundlagen, Modellierung und betriebswirtschaftliche Lösungen, München [u.a.] 2000.
- Oellermann, W. L. (2001): Architecting Web Services, Berkeley, California 2001.
- Oestereich, B. (1998): Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified modeling language, 4. aktualisierte Auflage, München.
- Oum, T. H./Park, J.-H./Zhang, A. (2001): Globalization and Strategic Alliances: The Case of the Airline Industry, Amsterdam.
- Phillips, L. A. (2000): Using XML, Special edition, Indianapolis 2000.
- Picot, A. / Reichwald R. / Wigand, R.: (2001): Die grenzenlose Unternehmung, 4. Auflage, Wiesbaden.
- Rawolle, J. (2002): Content management integrierter Medienprodukte: ein XML-basierter Ansatz, Wiesbaden 2002.
- Rawolle, J. / Burghardt, M. (2002): Web Services - eine Alternative für die zwischenbetriebliche Integration. In: IS Report 6 (2002) 2, S. 40-46.
- Reinhold, M. / Versteegen, G. (2002): Geordnetes Chaos: V-Modell versus Rational Unified Process, in: i'X 5/2002, S. 135-138.
- Royce, W. W. (1970): Managing the Development of Large Software Systems: Concepts and Techniques, in: Proceedings IEEE WESCON 1970, S. 1-9, Los Angeles.
- Reinhart, G. / Schliffenbacher, K. (2000): Unternehmensnetzwerke, In: Wildemann, H.: Virtuelle Fabrik: Wandelungsfähigkeit durch dynamische Unternehmenskooperationen, München 2000, S. 11-18.
- Schach, S. (2002): Object-Oriented and Classical Software Engineering, 5. Auflage, Boston.
- Schader, M. /Rundshagen, M. (1996): Objektorientierte Systemanalyse: eine Einführung, 2. Auflage, Berlin.
- Scheer, A.-W. (1998): ARIS – Vom Geschäftsprozeß zum Anwendungssystem, 3. völlig neu-bearb. und erw. Auflage, Berlin.
- Schinker, H. D. / Bange, C. / Mertens, H. (1999): Data warehouse und Data mining: marktführende Produkte im Vergleich, 2, München 1999.

- Schopp, B./Stanoevska-Slabeva, K.: Electronic Contracting in elektronischen Märkten, in: Weiber, R. H. (Hrsg, 2000): Handbuch Electronic Business, Wiesbaden, S. 755-770.
- Scribner, K. / Stiver, M. C. (2000): Understanding SOAP, Indianapolis 2000.
- Sedran, T. (1991): Wettbewerbsvorteile durch EDI ?. In: Information Management 6 (1991) 2, S. 16-21.
- Seely, S. (2002): SOAP: cross platform Web service development using XML, Upper Saddle River, NJ 2002.
- Siebert, H. (2001): Ökonomische Analyse von Unternehmensnetzwerken, in: Sydow, J. (Hrsg., 2001): Management von Netzwerkorganisationen, Wiesbaden, S. 8-27.
- Sleeper, B. / Robins, B. (2002): The Law of Evolution: A Pragmatic Analysis of the Emerging Web Services Market, San Francisco 2002.
- Snell, J. / Tidwell, D. / Kulchenko, P. (2002): Programming web services with SOAP, Beijing [u.a.] 2002.
- Snow, C. C. / Miles, R. E. / Coleman, H. J. (1992): Management 21st Century Network Organizations. In: Organizational Dynamics 20 (1992) 3, S. 5-19.
- Stahlknecht, P. /Hasenkamp, U. (2002): Einführung in die Wirtschaftsinformatik, 10. Auflage (überarbeitet und aktualisiert), Berlin 2002.
- Stevens, M. (2002a): Service Oriented Architecture Part 1, http://www.developer.com/db/print.php/10920_1010451_2, Abruf am 24.05.2003. (a)
- Stevens, M. (2002b): Service Oriented Architecture Part II, <http://www.developer.com/db/print.php/1014371>, Abruf am 24.05.2003. (b)
- Sydow, J. (1999): Management von Netzwerkunternehmen: Zum Stand der Forschung, in: Sydow, J. H. (Hrsg., 1999): Management von Netzwerkorganisationen, Wiesbaden, S. 279-314.
- Sydow, J. (2002): Strategische Netzwerke: Evolution und Organisation, 1. Aufl, Wiesbaden 2002.
- Szallies, C. (1998): Software-Spezifikation mit ereignisgesteuerten Prozeßketten (EPK), in: Softwaretechnik Trends 5/1998.
- Thompson, H. (2002), The Extensible Stylesheet Language (XSL), <http://www.w3.org/Style/XSL/>, Abruf am 30.04.2003.
- Thompson, H. S. / Beech, D. / Maloney, M. / Mendelsohn, N. (2001): XML Schema Part 1: Structures, <http://www.w3.org/TR/xmlschema-1/>, Abruf am 30.04.2003.
- Totok, A. (2000): Modellierung von OLAP- und Data-Warehouse-Systemen, Wiesbaden 2000.
- Upton, D. / McAfee, A. (1996): The Real Virtual Factory. In: Harvard Business Review 74 (1996) 7/8, S. 123-133.

- Vasudevan, V. (2001): A Web Service Primer,
<http://www.xml.com/pub/a/2001/04/04/webservices/>, Abruf am
25.06.2002.
- Volkmer, M. (1997): Entwicklung objektorientierter Analysemodelle für
Informationssysteme auf Grundlage von Prozessmodellen, Aachen.
- Weber, J. / Franken, M. / Göbel, V. (1998): Wandelbare Produktionsnetze - Kleine
und mittlere Unternehmen: Netzwerkfähigkeit im Management durch
Kennzahlen herstellen. In: *Industrie Management* 14 (1998) 6, S. 25-28,
insges. 4 S..
- Weitzel, T. / Harder, T. / Buxmann, P. (2001): *Electronic Business und EDI mit
XML*, Heidelberg 2001.
- Wildemann, H. (1997): Koordination in Unternehmensnetzwerken. In: *Zeitschrift
für Betriebswirtschaft* 67 (1997) 4, S. 417-439.
- Winand, U. / Nathusius, K. (1998): *Unternehmensnetzwerke und virtuelle
Organisationen*, Stuttgart 1998.
- Windeler, A./Lutz, A./Wirth, C. (2000): Netzwerksteuerung durch Selektion. Die
Produktion von Fernsehserien in Projektnetzwerken, in: Sydow, J./Windeler,
A. H. (Hrsg., 2000): *Steuerung von Netzwerken*, Wiesbaden, S. 178-205.
- Wittenberg, S. / Hess, T. (2002): Reputation als Steuerungsinstrument in
netzwerkinternen Märkten, In: Engelen, M. / Homann, J.: *Virtuelle
Organisation und Neue Medien*, 2002, S. 107-131.
- Wohlgemuth, O. (2002): *Management netzwerkartiger Kooperationen:
Instrumente für die unternehmensübergreifende Steuerung*, Wiesbaden 2002.
- Wohlgemuth, O. / Hess, T. (2002): Evaluating the Success of Virtual
Corporations with CONECT: Basic Procedural Indications and Practical
Application, In: Franke, U. J.: *Managing virtual web organizations in the 21st
century*, London 2002, S. 292-318.
- Zuser, W. / Biffel, S. / Grechenig, T. / Köhle, M. (2001): *Software-Engineering
mit UML und dem Unified Process*, München.

Anhang

Pflichtenheft von VICOPLAN in der Kurzfassung

Zielbestimmung:

Die Software soll das Management von Unternehmensnetzwerken unterstützen. Hierbei sind auftragsbezogene Funktionen und auftragsübergreifende Funktionen zu unterscheiden. Zu näheren Zielbestimmung werden Musskriterien (Produktkern), Wunschkriterien und Abgrenzungskriterien festgelegt.

Musskriterien

- *Grundfunktionalitäten*
 - Verwalten von Koordinatoren-, Partner- und Kundendaten
 - Verwalten von „Kompetenzen“, die im Netzwerk verfügbar sind
- *Auftragsbezogene Ebene*
 - Verwalten von Kundenanfragen
 - Verwalten von Angebots- und Auftragsdaten
 - Ablage von Dateianhängen zu Anfragen, Angeboten, Aufträgen
 - Aufteilung von Anfragen in Arbeitspakete, denen benötigte Kompetenzen zugeordnet werden können
 - „Ausschreibung“ von Arbeitspaketen durch Benachrichtigung der Partner und Bereitstellung von Erfassungsmasken für entsprechende Angebote von Partnern
 - Verwaltung der Ausschreibung (Zeitlimits, Vollständigkeit)
 - Kompositionsmodul für Gesamt(Kunden-)Angebote aus einzelnen Partnerangeboten
 - Ablauf- und Budgetplanung für Aufträge

- Aufteilung von Arbeitspaketen in zeitbezogene Aktivitäten
- Monitoring-System für Kosten je Arbeitspaket
- Dokumentationssystem für Ereignisse nach Auftragsvergabe
- Beginn/Abschluss von Arbeitspaketen
- Erfassung von Kosten
- Eingang von (Kunden-) Zahlungen
- Ausgang von Zahlungen (an Partner)
- Leistungsnachweise („digitaler Stundenzettel“)
- *Auftragsübergreifende Ebene*
 - Analyse von Kundenumsätzen
 - Analyse von Partnerumsätzen und -bewertungen
 - Analyse von Kompetenzauslastung, -anforderung

Wunschkriterien

- *Auftragsbezogene Ebene*
 - Möglichkeit Arbeitspakete zurückzurufen und Ausschreibungen nach der Publikation zu modifizieren
 - Möglichkeit bei laufenden Aufträgen Arbeitspakete zu ergänzen
- *Auftragsübergreifende Ebene*
 - Controlling- und CRM-Funktionen
 - Customer-Lifetime-Value-Analyse
 - Kennzahlenmodul

Abgrenzungskriterien

- keine buchhalterischen Funktionen (Kontierung o.ä.)
- keine Projektmanagementsoftware

Produkteinsatz

Die Software soll den verteilten Wertschöpfungsprozess in Unternehmensnetzwerken administrativ unterstützen. Diese Unterstützungsfunktion beginnt mit dem Erfassen einer Kundenanfrage über das Internet und endet mit der

Dokumentation und Abschluss des Auftrags. Der Schwerpunkt liegt hierbei auf Transparenz und umfassender Dokumentation der unterstützten Vorgänge.

Anwendungsbereiche

- kaufmännisch / administrativer Bereich (auftragsbezogener Teil)
- Management-Informationssystem (auftragsübergreifender Teil)

Zielgruppen

Am System sollen sowohl die Kunden des Unternehmensnetzwerks, dessen Koordinator als auch die Partnerunternehmen arbeiten können.

Betriebsbedingungen

Software läuft unbeaufsichtigt im Serverbetrieb und bedient verschiedene Web-Clients in Büroumgebung.

Produktübersicht

- *Produktfunktionen (siehe hierzu Tabelle 5)*
- *Produktdaten (siehe hierzu Tabelle 6)*
- *Stadien eines Auftrags*

Ein Auftrag durchläuft ab der Erfassung der Anfrage durch den Kunden und dem Abschluss der Auftragsleistung aus Sicht des Systems verschiedene Stadien, deren zeitlicher Umfang ebenfalls langfristig zu speichern ist.

Stadiumsnummer	Stadiumsbeschreibung
1	Anfrage (neu)
2	Anfrage (bearbeitet)
3	Interne Ausschreibung
4	Angebotserstellung
5	Auftrag

Qualitätsanforderungen

Der Fokus des Entwicklungsprozesses soll auf der betriebswirtschaftlich relevanten Funktionalität liegen. Es soll eine lauffähige Softwareversion entwickelt werden, die jedoch nicht die Performanz und Sicherheit eines marktreifen Produkts aufweisen muss.

Bei der Entwicklung soll darauf geachtet werden, dass – wo sinnvoll - offene Schnittstellen (wie z.B. XML) verwendet werden und spätere Änderungen an der Software möglichst ein-fach durchzuführen sind.

Nicht-Funktionale Anforderungen

Der Zugang ist passwordgeschützt. Passwörter sollen nicht im Klartext über das Netzwerk übertragen werden, da das System evtl. im Internet eingesetzt werden soll.

Ein Großteil der Datensatzänderungen ist gesondert zu protokollieren. Bei wichtigen Feldern (noch zu ermitteln) soll angegeben werden, wann welcher Benutzer den Datensatz zuletzt geändert hat.

Wenn die Freigabe und Genehmigung von Aufträgen ohne handschriftliche Unterzeichnung abgewickelt wird, dann ist eine digitale Signatur (z.B. PGP) zu verwenden und der gesamte Datenverkehr zu verschlüsseln (z.B. durch SSL).

Technische Produktumgebung

Das Produkt ist eine geschlossene Menge von PHP-Skripten und wird von einem Webserver mit PHP-Interpreter ausgeführt. Damit kann es im Intra- und Internet eingesetzt werden.

Aufgrund des Pilot-Charakters konzentriert sich die Entwicklungsarbeit auf inhaltliche Aspekte, weshalb das Produkt sicherheitstechnisch nicht voll ausgereift ist. Aus diesem Grund werden weitere Zugangsrestriktionen benötigt (z.B. Firewall).

- *Software*
Serverseitig kann mit jeder Webserversoftware gearbeitet werden, die einen PHP-Interpreter einbinden kann (z.B. Apache, MS Internet Information Server). Weiterhin wird eine SQL-Datenbank benötigt. Entwicklungsdatenbank ist mySQL, Unterstützung für Microsoft SQLServer, Oracle und PostgreSQL soll einfach ergänzt werden können. Damit ist das Produkt auf allen gängigen Betriebssystemen lauffähig (Windows, Linux, MacOS X, ...).

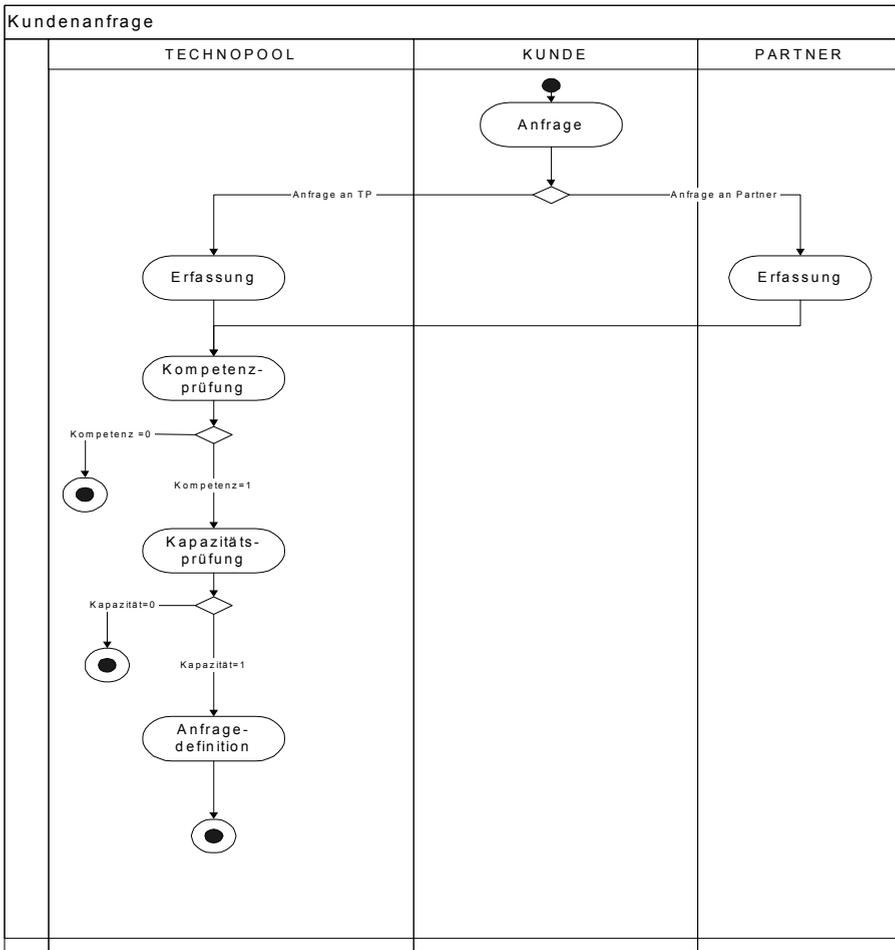
Clientseitig wird der Microsoft Internet Explorer ab Version 5.5 benötigt.

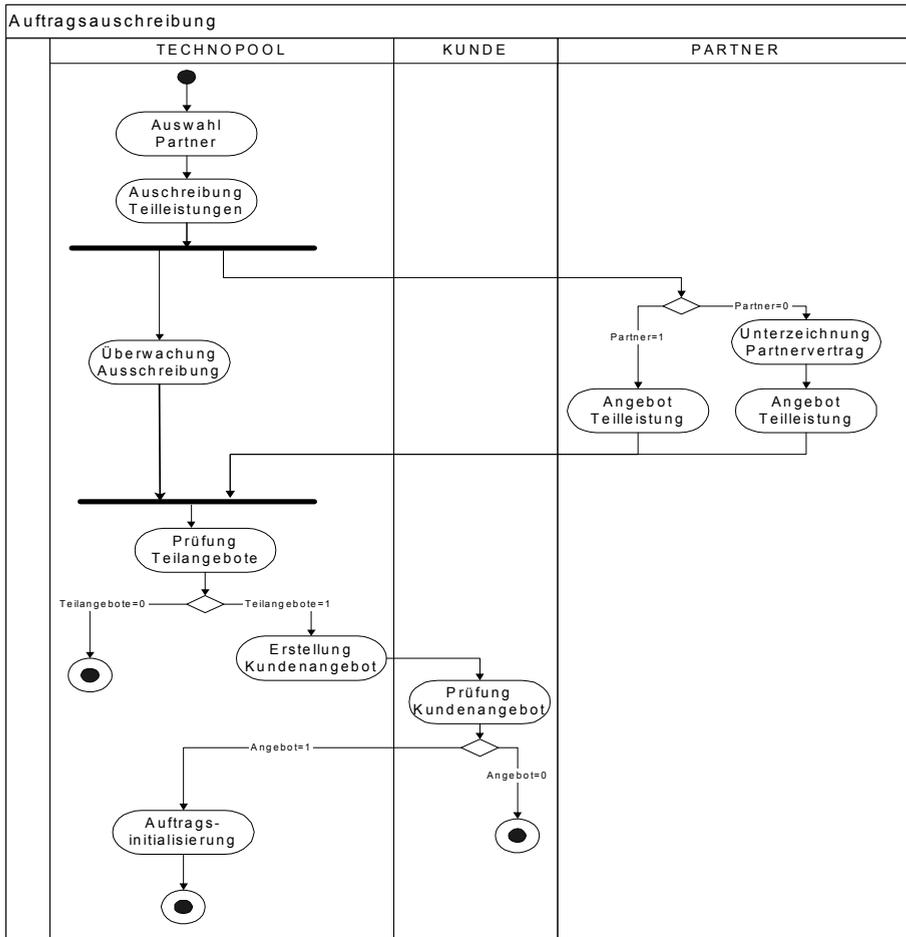
- *Hardware*
Serverseitig wird eine Plattform benötigt, die die o.a. Software betreiben kann. Beispielsweise PC, DEC Alpha, SUN sparc, HP PA-RISC u.v.m.
Clientseitig wird ein Farbdisplay mit einer Grafikauflösung von mindestens 1024x768 Bildpunkten dringend empfohlen.

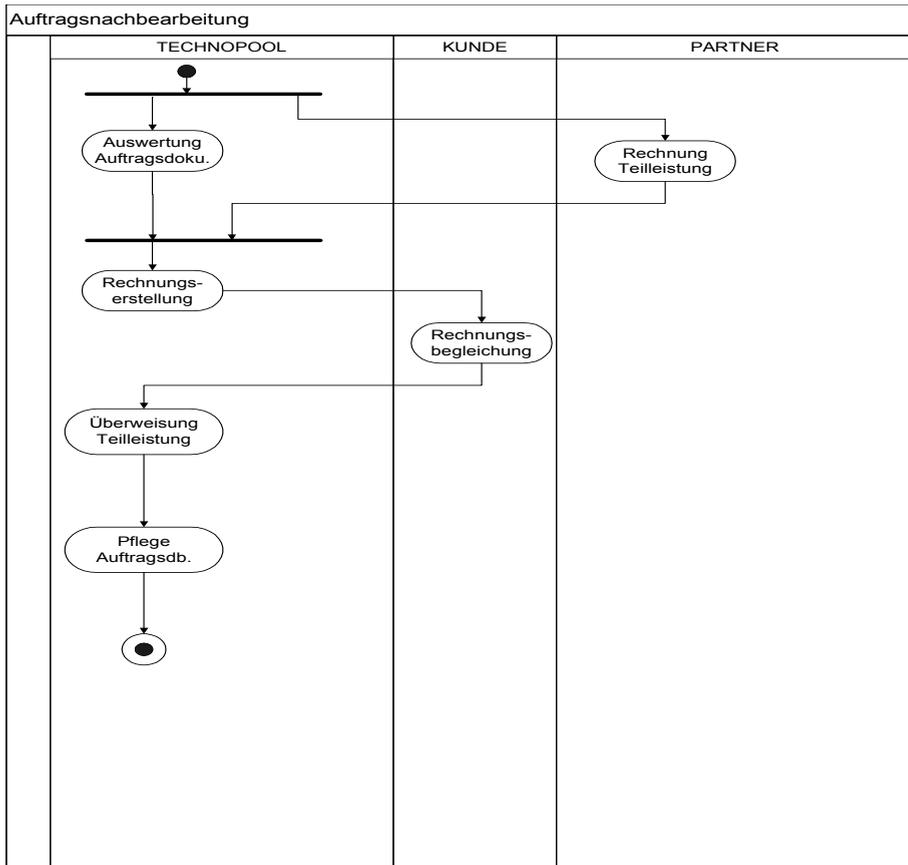
Schnittstellen

Vom System generierte Dokumente werden im PDF-Format in der Datenbank abgelegt und stehen anderen Anwendungen zur Verfügung. Weitere Schnittstellen sind denkbar.

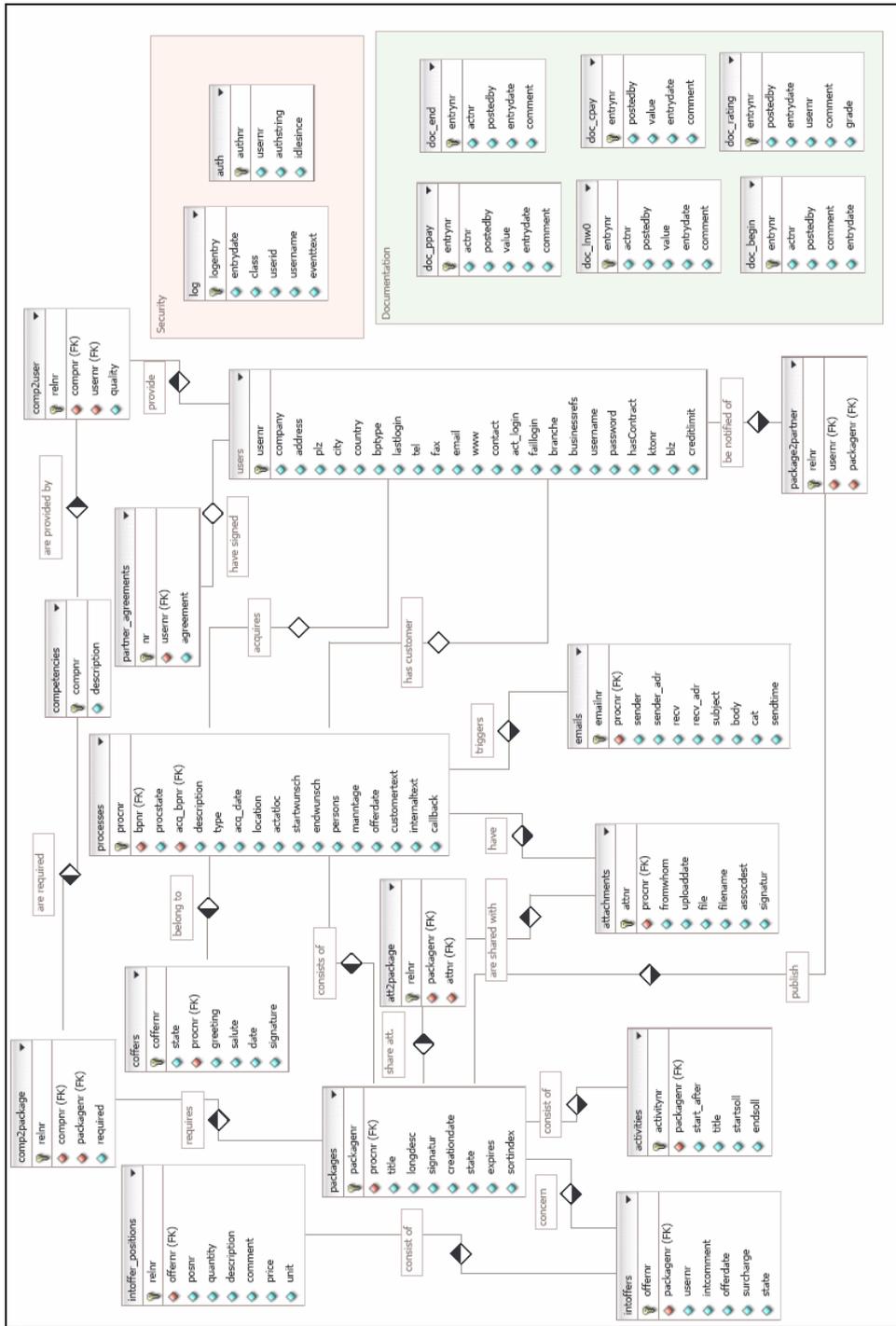
Aktivitätsdiagramme







Datenbankmodell



Ausgewählte relevante Veröffentlichungen, durchgeführte Workshops, Messeteilnahmen sowie aktuelle Informationen

Publikationen

- Burghardt, M. (2003): Innovative Technologien in virtuellen Netzwerken - Einsatzmöglichkeiten von XML, Web Services und OLAP, Arbeitsberichte der Abt. Wirtschaftsinformatik II, Universität Göttingen, Nr. 10, Göttingen, 2003.
- Burghardt, M., Diekmann, T., Hagenhoff, S., Schumann, M. (2003): Structure of a Web-Based IT-System for Virtual Networks. In: G. Kotsis, S. Bressan, B. Catania, I. K. Ibrahim (Hrsg.): Fifth International Conference on Information Integration and Web-based Applications and Services (IIWAS2003), Jakarta, Indonesien, 2003, S. 87 – 96.
- Burghardt, M., Gehrke, N., Hagenhoff, S., Schumann, M. (2003): Spezifikation und Abwicklung von Workflows auf Basis von Web-Services. In: HMD - Praxis der Wirtschaftsinformatik 234 (2003), S. 61 – 69.
- Burghardt, M.; Gehrke, N.; Schumann, M. (2003): Eine Architektur zur Abrechnung von Web Services. In: Eckstein, R.; Tolksdorf, R. (Hrsg.): XMIDX 2003 XML-Technologien für Middleware – Middleware für XML-Technologien, Bonn, S. 45 – 56.
- Burghardt, M.; Gehrke, N.; Schumann, M. (2003): Implikationen kommerzieller Web Services. In: Eckstein, R.; Tolksdorf, R. (Hrsg.): XMIDX 2003 XML-Technologien für Middleware – Middleware für XML-Technologien, Bonn, S. 71 – 82.
- Hess, T. (2002): Planning and Control of Virtual Corporations in the Service Industry: The Prototype VICOPLAN, in: Proceedings of the 35th Hawaii International Conference on System Sciences - 2002, Hawaii, 2002.
- Hess, T.; Wittenberg, S. (2003): Interne Märkte in Dienstleistungsnetzwerken. In: Bruhn, M.; Stauss, B. (Hrsg.): Dienstleistungsnetzwerke. Jahrbuch Dienstleistungsmanagement 2003, Wiesbaden, S. 161 – 184.
- Hess, T.; Katzy, B.; Wittenberg, S (2003): Kooperationstools- Werkzeug oder Spielzeug? In: new management 11/2003, S. 16 – 21.
- Wittenberg, S.; Hess, T. (2002): Reputation als Steuerungsinstrument in netzwerkinternen Märkten. In: Engelen, M./ Homann, J. (Hrsg.): Virtuelle Organisation und Neue Medien, Köln u.a., S. 107 – 131.

Messeteilnahmen

- CeBIT 2003, eigener Messestand im Ausstellungsreich von „Bayern Innovativ“, 2003.
- Dienstleistungstagung des BMBF, Berlin, 2003.
- Fakultätstag des Departments für BWL der Uni München, 2004.
- Agenturgipfel der IHK München, 2004.

Durchgeführte Workshops

- Göttingen, Universitätsbibliothek, 24.2.04: offizieller BMBF-Abschlussworkshop, ca. 50 Teilnehmer.
- Rostock, Industrie- und Handelskammer, 15.4.04, Kooperationsworkshop, ca. 20 Teilnehmer.
- Darmstadt, TÜV Managementsysteme, Kooperationsworkshop, 23.6.04, ca. 20 Teilnehmer.
- Gera, Wirtschaftsförderung (Marriot Hotel), 19.7.04, Kooperationsworkshop, ca. 40 Teilnehmer.

Aktuelle Informationen

- Internetpräsenz zum vorgestellten Prototypen VICOPLAN unter <http://www.vicoplan.de>
- Informationen zum abgeschlossenen BMBF-Vorhaben sind unter der Adresse <http://www.dl2100.de/projectdetail.php?projectid=115> zu finden.
- Eine Diskussion über verschiedenen Kooperationstools ist unter <http://www.ve-Forum.de> verfügbar.