



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Wirtschaftswissenschaften
Fachgebiet Wirtschaftsinformatik

Föderierte Portale zur Überwindung inner- und
zwischenbetrieblicher Portalproliferation

– Referenzrahmen, Konzepte, Modelle und Realisierung –

Dissertation
zur Erlangung des akademischen Grads
des Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)
der Universität Paderborn

vorgelegt von
Dipl.-Wirt.-Inf. Olaf Hahn
Am Hang 29
34369 Hofgeismar

Hofgeismar, Oktober 2004

Inhaltsübersicht

1	Einleitung	1
1.1	Szenario und Zielsetzung der Arbeit	1
1.2	Wissenschaftstheoretische Ausrichtung der Arbeit	3
1.3	Aufbau der Arbeit	4
2	Abgrenzung des Forschungsumfelds	6
2.1	Aktuelle Entwicklungen im Unternehmensumfeld	6
2.2	Unternehmensportale	17
2.3	Forschungsziel	39
3	Föderationen	46
3.1	Ursprung des Begriffs Föderation	46
3.2	Der Begriff Föderation in der Betriebswirtschaftslehre	47
3.3	Der Begriff Föderation in der Informationstechnologie	48
3.4	Zusammenfassung	78
4	Konzeption einer Architektur zur Kopplung von Portalen	80
4.1	Entwurfsmethoden	80
4.2	Basiskomponenten und -dienste eines Portals	84
4.3	Operationalisierung gekoppelter Portale	92
4.4	Anforderungen	97
4.5	Kommunikations- und Synchronisationsarchitektur gekoppelter Portale	110
4.6	Daten- und Funktionsmodell gekoppelter Portale	152
4.7	Zusammenfassung	157
5	Umsetzung des Konzepts gekoppelter Portale am Beispiel des Workplace Portals G8	158
5.1	Basistechnologien	158
5.2	Kopplung von G8-Portalen	169
5.3	Zusammenfassung	190
6	Praktische Nutzung gekoppelter Portale – Anwendungsszenario PAVONE AG	191
6.1	Vorbemerkungen	191
6.2	Rahmenbedingungen	191
6.3	Aufbau und Inhalt des gekoppelten Platzes	193
6.4	Erfahrungen	194
6.5	Limitationen	198
6.6	Zusammenfassung	199
7	Schlussbetrachtung und Ausblick	200
7.1	Ausblick	200
7.2	Ergebnis und kritische Würdigung der Arbeit	208
8	Literaturverzeichnis	213
9	Anhang	232
9.1	WSDL-Typen	232
9.2	WSDL-Nachrichten	236
9.3	WSDL-Operationen	242

Inhaltsverzeichnis

1	Einleitung	1
1.1	Szenario und Zielsetzung der Arbeit	1
1.2	Wissenschaftstheoretische Ausrichtung der Arbeit	3
1.3	Aufbau der Arbeit	4
2	Abgrenzung des Forschungsumfelds	6
2.1	Aktuelle Entwicklungen im Unternehmensumfeld	6
2.1.1	Unternehmensformen zwischen Hierarchie und Markt	7
2.1.1.1	Organisationskontinuum	8
2.1.1.2	Rolle von Information und Informationsüberflutung	10
2.1.1.3	Informationstechnologie als „Enabler“	11
2.1.1.4	Vertrauen	13
2.1.1.5	Zusammenfassung	13
2.1.2	Wissensintensive Unternehmensformen	15
2.1.2.1	Definition und Abgrenzung	15
2.1.2.2	Teams als elementare Organisationsstruktur	16
2.1.2.3	Zusammenfassung	17
2.2	Unternehmensportale	17
2.2.1	Grundlagen	18
2.2.1.1	Historie	18
2.2.1.2	Klassifizierung	19
2.2.1.3	Funktionen und Eigenschaften	22
2.2.1.4	Begriffsdefinition	23
2.2.1.5	Abgrenzung zu anderen Technologien	24
2.2.1.5.1	Electronic Data Interchange	24
2.2.1.5.2	Applikationsintegration	26
2.2.2	Ausgewählte Aspekte von Portalumgebungen	30
2.2.2.1	Unterstützungsfunktion teambasierter Kooperation	30
2.2.2.2	Interorganisationale Nutzung von Portalen	32
2.2.2.3	Das Multi-Portal-Problem	33
2.2.2.3.1	Situationsbeschreibung	34
2.2.2.3.2	Existierende Lösungsvorschläge	35
2.2.3	Forschungs- und Praxislücke	38
2.3	Forschungsziel	39
2.3.1	Zieldefinition	39
2.3.2	Andere Forschungsansätze	41
2.3.2.1	Puschmann und Alt: „Process Portals inter-organizational networking of businesses“	41
2.3.2.2	ANSA: „Establishing Co-operation in Federated Systems“	42
2.3.2.3	FOSTER: „Information and Communication Technology support infrastructures and interoperability for Virtual Organisations“	43
2.3.2.4	Zhang et al.: „Enterprise Virtualisation: Concept, Methodology, and Implementation“	44

3	Föderationen	46
3.1	Ursprung des Begriffs Föderation	46
3.2	Der Begriff Föderation in der Betriebswirtschaftslehre	47
3.3	Der Begriff Föderation in der Informationstechnologie	48
3.3.1	Föderierte Datenbanksysteme	49
3.3.1.1	Begriffsdefinition und Basisarchitektur	50
3.3.1.2	Charakteristika föderierter Datenbanksysteme	53
3.3.1.3	Taxonomie von Multi- und föderierten DBMS	57
3.3.1.4	Aspekte der Verteilung	59
3.3.1.5	Aspekte der Heterogenität und Autonomie	63
3.3.1.6	Sicherheit in föderierten Datenbanksystemen	67
3.3.2	Föderierte Informationssysteme	73
3.3.3	Föderierte Benutzeridentitäten	76
3.4	Zusammenfassung	78
4	Konzeption einer Architektur zur Kopplung von Portalen	80
4.1	Entwurfsmethoden	80
4.1.1	Bottom-Up-, Top-Down- und Jojo-Entwurf	80
4.1.2	Vorgehen zum Entwurf einer Architektur gekoppelter Portale	83
4.2	Basiskomponenten und -dienste eines Portals	84
4.2.1	Informations- und Anwendungssysteme	84
4.2.2	Portlets	85
4.2.3	Seiten	86
4.2.4	Plätze	87
4.2.5	Weitere Dienste	89
4.2.6	Zusammenfassung	91
4.3	Operationalisierung gekoppelter Portale	92
4.3.1	Begriffsbestimmung	92
4.3.2	Kontinuum der Portalkopplung	93
4.3.3	Taxonomie gekoppelter Portale	95
4.3.4	Zusammenfassung	97
4.4	Anforderungen	97
4.4.1	Betriebswirtschaftliche Anforderungen	97
4.4.1.1	Spezifität und Flexibilität der Bindung	98
4.4.1.2	Sichtbarkeit der internen Organisation	99
4.4.2	Technische Anforderungen	100
4.4.2.1	Aspekte der Verteilung	100
4.4.2.2	Aspekte der Heterogenität und Autonomie	102
4.4.2.3	Kopplungsarchitektur	104
4.4.2.4	Sicherheit in gekoppelten Portalsystemen	105
4.4.2.4.1	Anbieten von Portlets	106
4.4.2.4.2	Anbieten von Seiten und Plätzen und gemeinsame Nutzung von Plätzen	109
4.5	Kommunikations- und Synchronisationsarchitektur gekoppelter Portale	110
4.5.1	Topologie	110
4.5.2	Verteilung	112
4.5.2.1	Synchrone vs. asynchrone Replikation	112
4.5.2.2	Identifikation und Verteilung von Aktualisierungen	113

4.5.2.3	Vollständig dezentraler vs. dezentraler Ansatz mit Koordinationsinstanz	115
4.5.3	Rollen	117
4.5.4	Synchronisationsmodell	121
4.5.4.1	Definition von Zeitstempeln als Basis der Synchronisation	122
4.5.4.2	Synchronisation auf den verschiedenen Stufen der Portalkopplung.....	123
4.5.4.2.1	Anbieten von Portlets.....	123
4.5.4.2.2	Anbieten von Seiten oder Plätzen.....	126
4.5.4.2.3	Gemeinsame Nutzung von Plätzen	127
4.5.4.3	Konfliktmanagement.....	137
4.5.4.3.1	Seitenregister.....	138
4.5.4.3.2	Layout von Seiten	140
4.5.4.3.3	Portlets.....	142
4.5.4.4	Zeitstempel zur Minimierung der Synchronisationsdaten.....	144
4.5.4.5	Transparenz zwischen lokalen und Remote-Portlet-Instanzen ...	146
4.5.4.5.1	Standardverarbeitung	147
4.5.4.5.2	Portlet-Ersetzung mittels Portlet-Typen	150
4.5.4.5.3	Nicht verfügbare Portlets	151
4.6	Daten- und Funktionsmodell gekoppelter Portale	152
4.6.1	Kommunikationsbeziehung zwischen Portalen	153
4.6.2	Anbieten von Portlets	154
4.6.3	Anbieten von Seiten	155
4.6.4	Gemeinsame Nutzung von Plätzen.....	156
4.7	Zusammenfassung	157
5	Umsetzung des Konzepts gekoppelter Portale am Beispiel des Workplace Portals G8	158
5.1	Basistechnologien.....	158
5.1.1	Workplace Portal G8	158
5.1.1.1	Architektur	158
5.1.1.2	Basiselemente	160
5.1.2	Web Services	161
5.1.2.1	Simple Object Access Protocol	163
5.1.2.2	Web Service Description Language	164
5.1.2.3	Universal Description, Discovery and Integration.....	165
5.1.2.4	Web Services als Middleware für die Kopplung von Portalen	166
5.1.2.5	Web-Services-Frameworks und UDDI-Implementierungen	167
5.2	Kopplung von G8-Portalen.....	169
5.2.1	Basisdienste für die Portalkopplung.....	170
5.2.2	Kommunikationsbeziehung zwischen Portalen	173
5.2.2.1	Portalregistrierung	173
5.2.2.2	Portalverbindung	174
5.2.3	Anbieten von Portlets	176
5.2.3.1	Erweiterungen zum Anbieten von Portlets	176
5.2.3.1.1	Konfiguration.....	176
5.2.3.1.2	Erweiterung des lokalen Objektmodells.....	177
5.2.3.1.3	Integration bestehender Content-Adaptoren.....	178

5.2.3.2	Erweiterung zur Integration von Remote-Portlets.....	178
5.2.3.2.1	Einbettung in das Portlet Definition Repository und Konfiguration.....	178
5.2.3.2.2	Remote SOAP Adapter.....	181
5.2.3.2.3	Portlet-Routing.....	182
5.2.4	Anbieten von Seiten.....	183
5.2.4.1	Vorlagen als Seiten beim Anbieter.....	184
5.2.4.2	Vorlagen als Remote-Seiten beim Nachfrager.....	185
5.2.5	Gemeinsame Nutzung von Plätzen.....	187
5.2.5.1	Erweiterungen des internen Objektmodells.....	187
5.2.5.2	Einbettung in die Benutzerschnittstelle.....	188
5.3	Zusammenfassung.....	190
6	Praktische Nutzung gekoppelter Portale – Anwendungsszenario PAVONE AG.....	191
6.1	Vorbemerkungen.....	191
6.2	Rahmenbedingungen.....	191
6.3	Aufbau und Inhalt des gekoppelten Platzes.....	193
6.4	Erfahrungen.....	194
6.5	Limitationen.....	198
6.6	Zusammenfassung.....	199
7	Schlussbetrachtung und Ausblick.....	200
7.1	Ausblick.....	200
7.1.1	Soziologische und betriebswirtschaftliche Aspekte.....	201
7.1.2	Berücksichtigung von Standardisierungen.....	201
7.1.3	Erweiterung des Synchronisationsmodells.....	202
7.1.4	Integration der Inter-Portlet-Kommunikation.....	203
7.1.5	Föderierung weiterer Portaldienste.....	204
7.1.6	Nutzungs- und Abrechnungsverfahren.....	206
7.1.7	Weiterentwicklung der Implementierung.....	206
7.1.8	Implementierung der Kopplung in einem anderen Portal-Framework.....	207
7.2	Ergebnis und kritische Würdigung der Arbeit.....	208
8	Literaturverzeichnis.....	213
9	Anhang.....	232
9.1	WSDL-Typen.....	232
9.2	WSDL-Nachrichten.....	236
9.3	WSDL-Operationen.....	242

Abbildungsverzeichnis

Abbildung 2-1:	Faktoren für Veränderungsprozesse von Unternehmen und Märkten	7
Abbildung 2-2:	Kontinuum organisatorischer Gestaltungsformen	9
Abbildung 2-3:	Klassifizierung von Unternehmensportalen	20
Abbildung 2-4:	Portalevolution	21
Abbildung 2-5:	Einsatzszenarien für Portale vs. EDI	25
Abbildung 2-6:	Portal als Shared Information Workspace	31
Abbildung 2-7:	Hype Cycle for Portal Ecosystems, 2003.....	36
Abbildung 2-8:	Klassen von VO und zugehörige IuK-Technologien	43
Abbildung 3-1:	Beispiel einer Topologie eines föderierten Datenbanksystems	51
Abbildung 3-2:	Allgemeine Architektur föderierter Datenbanksysteme.....	52
Abbildung 3-3:	Alternativen zur physischen Realisierung des Föderationsdienstes.....	53
Abbildung 3-4:	Taxonomie von DBS unter Berücksichtigung der Autonomie der CDBS	57
Abbildung 3-5:	Taxonomie von DBS unter Berücksichtigung der Dimensionen Verteilung, Heterogenität und Autonomie	59
Abbildung 3-6:	Drei-Ebenen ANSI/SPARC DBMS Schemaarchitektur.....	64
Abbildung 3-7:	Komponenten der Drei-Ebenen-Schemaarchitektur	64
Abbildung 3-8:	Fünf-Ebenen-Schemaarchitektur eines FDBS	65
Abbildung 3-9:	Komponenten der Fünf-Ebenen-Schemaarchitektur	66
Abbildung 3-10:	Drei-Ebenen-Architektur eines FIS	74
Abbildung 3-11:	Architektur eines Mediator-basierten Informationssystems	75
Abbildung 3-12:	Klassifikation föderierter Informationssysteme	75
Abbildung 4-1:	Top-Down-Ansatz zur Sichtenintegration	82
Abbildung 4-2:	Gemeinsame Strukturelemente verschiedener Portal- implementierungen	84
Abbildung 4-3:	Strukturelle Darstellung der Beziehungen der Portalelemente	89
Abbildung 4-4:	Kontinuum der Portalkopplung.....	93
Abbildung 4-5:	Taxonomie von Portalkopplung unter Berücksichtigung der Dimensionen Verteilung, Heterogenität und Autonomie	96

Abbildung 4-6:	Komponenten der logischen Architektur zur Portalkopplung	103
Abbildung 4-7:	Architektur der Föderierungsdienste gekoppelter Portale.....	105
Abbildung 4-8:	Netzwerktopologien	111
Abbildung 4-9:	Rollen beim Anbieten von Portlets.....	118
Abbildung 4-10:	Rollen beim Anbieten von Seiten und Plätzen.....	119
Abbildung 4-11:	Rollen bei gemeinsamer Nutzung eines Platzes	119
Abbildung 4-12:	Lebenszyklus eines gemeinsam genutzten Platzes	120
Abbildung 4-13:	Haupt- und Subphasen des Synchronisationszyklus.....	127
Abbildung 4-14:	Synchroner Ausgangszustand und abgeleitete asynchrone Seitenregister.....	131
Abbildung 4-15:	Seitenregister des Coordinator-Place nach Abschluss der Phase 1 der Synchronisation	132
Abbildung 4-16:	Seitenregister nach Abschluss der zweiphasigen Synchronisation	133
Abbildung 4-17:	Ausgangszustand des Layouts einer Seite vor der Synchronisation	134
Abbildung 4-18:	Visualisierung der Phase 1 der Synchronisation der Seitenstruktur	135
Abbildung 4-19:	Struktur der Seiten nach Abschluss der zweiphasigen Synchronisation	137
Abbildung 4-20:	Synchronisation des Seitenregisters mit Konflikten	139
Abbildung 4-21:	Ausgangssituation bei Konflikten in der Struktur einer Seite	141
Abbildung 4-22:	Layout der Seite nach Auflösung der Konflikte beim Coordinator- Place	142
Abbildung 4-23:	Repräsentation der Seite nach Auflösung der Konflikte beim Coordinator-Place	144
Abbildung 4-24:	Beispielszenario für Remote-Portlets in verteilten gemeinsam genutzten Plätzen	148
Abbildung 4-25:	Schichten des Daten- und Funktionsmodells zur Kopplung von Portalen.....	153
Abbildung 4-26:	Operationen zum Management der Kommunikationsbeziehung	154
Abbildung 4-27:	Operationen zum Abruf von Portlet-Definitionen und zur Nutzung von Portlet-Instanzen.....	155
Abbildung 4-28:	Operation zum Abruf angebotener Seiten	156
Abbildung 4-29:	Operationen zur verteilten gemeinsamen Nutzung von Plätzen.....	157
Abbildung 5-1:	Konzeptionelle Portalarchitektur des G8-Portals	159

Abbildung 5-2:	Rollen und Aufgaben in einer serviceorientierten Architektur	162
Abbildung 5-3:	Web Service Stack.....	162
Abbildung 5-4:	Struktur einer SOAP-Nachricht.....	163
Abbildung 5-5:	Struktur eines WSDL-Dokuments	164
Abbildung 5-6:	ER-Diagramm des UDDI-Datenmodells	166
Abbildung 5-7:	Direkte Kommunikationsbeziehung zwischen dem Portal Soap Gateway und dem Portal Soap Gateway Proxy.....	171
Abbildung 5-8:	Indirekte Kommunikationsbeziehung über einen Intermediär-Portlet-Provider mittels des Portal Router Service	172
Abbildung 5-9:	Portal-Connection-Konfigurationsdokument	175
Abbildung 5-10:	Bereich zur Zugriffssteuerung eines gekoppelten Portals auf ein Portlet in einem Portlet-Konfigurationsdokument.....	176
Abbildung 5-11:	Bereich zur Zugriffssteuerung auf ein Remote-Portlet in einem Portlet-Konfigurationsdokument.....	180
Abbildung 5-12:	Integration des Remote Soap Adapters.....	181
Abbildung 5-13:	Ausschnitt einer SOAP-Nachricht zum Abrufen des Markups eines Remote-Portlets.....	182
Abbildung 5-14:	Beispiel des SOAP-Headers zum Routing von Remote-Portlets.....	183
Abbildung 5-15:	Erweiterung von Vorlagen zum Anbieten von Seiten.....	184
Abbildung 5-16:	Integration von extern zur Verfügung gestellten Vorlagen in den Dialog zum Editieren des Seitenregisters	186
Abbildung 5-17:	Ausschnitt aus dem Konfigurationsdialog des Coordinator-Place zur Definition von Consumer-Portalen	188
Abbildung 6-1:	Ansicht des gekoppelten Platzes des Anwendungsszenarios	193
Abbildung 7-1:	Hype Cycle for the Portal Ecosystem, 2004	212

Tabellenverzeichnis

Tabelle 4-1:	Phase 1 der Synchronisation des Seitenregisters	131
Tabelle 4-2:	Phase 2 der Synchronisation des Seitenregisters	132
Tabelle 4-3:	Phase 1 der Synchronisation der Struktur einer Seite	136
Tabelle 4-4:	Phase 2 der Synchronisation der Struktur einer Seite	136

Abkürzungsverzeichnis

AG	Aktiengesellschaft
AI	Application Integration
ANSA	Advanced Networked Systems Architecture
ANSI	American National Standards Institute
API	Application Programming Interface
ASF	Apache Software Foundation
AXIS	Apache Extensible Interaction System
B2B	Business to Business
B2C	Business to Consumer
B2E	Business to Employee
BPEL4WS	Business Process Execution Language for Web Services
CDBMS	Component Database Management System
CDBS	Component Database System
CDM	Canonical/Common Data Model
CIM	Computer Integrated Manufacturing
CMS	Content Management System
CRM	Customer Relationship Management
CSCW	Computer Supported Cooperative Work
DAC	Discretionary Access Control
DB	Database
DBMS	Database Management System
DBS	Database System
DCMI	The Dublin Core Metadata Initiative
DDL	Data Definition Language
DML	Data Manipulation Language
DOM	Document Object Model
EAI	Enterprise Application Integration
EDI	Electronic Data Interchange
EFIS	Engineering Federated Information Systems
EIP	Enterprise Information Portal
EP	Enterprise Portal
ERP	Enterprise Resource Planning
FDBMS	Federate Database Management System
FDBS	Federate Database System

FIS	Federate Information Systems
FTP	File Transfer Protocol
GCC	Groupware Competence Center
HP	Hewlett Packard
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPR	Reliable Hypertext Transfer Protocol
IBM	Industrial Business Machines
ICT	Information and Communication Technology
ID	Identification
IETF	The Internet Engineering Task Force
IT	Information Technology
IuK	Information und Kommunikation
KBO	Knowledge-based organization
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control
MBIS	Mediator-based Information Systems
MDBS	Multi Database System
MLS	Multilevel Security
MQ	Message Queuing
OASIS	Organization for the Advancement of Structured Information Standards
PDA	Personal Digital Assistant
PEDL	Portal Element Definition Language
PEML	Portal Element Manipulation Language
POAI	Portal-Oriented Application Integration
RDBMS	Relational Database Management System
RPC	Remote Procedure Call
SAX	Simple API for XML
SES	Smart Enterprise Suite
SIMPLE	SIP for Instant Messaging and Presence Leveraging
SIP	Session Initiation Protocol
SMB	Small and Midsize Business
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SPR	Software Problem Report

SSO	Single Sign On
tModel	Technology Model
UBR	Universal Business Registry
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
VO	Virtuelle Organisation / Virtual Organization
VS	Virtual Space
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WS	Web Service
WSCl	Web Services Choreography Interface
WSDL	Web Services Definition Language
WSFL	Web Services Flow Language
WSIL	Web Service Inspection Language
WSRP	Web Service for Remote Portlets
WWW	World Wide Web
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
ZKL	Zugriffskontrollliste

1 Einleitung

1.1 Szenario und Zielsetzung der Arbeit

Die Rahmenbedingungen, in denen Organisationen agieren, haben sich in den letzten Jahren stark verändert. Aus den vielfältigen Gründen hierfür werden zwei besonders relevante Trends herausgegriffen.

Der erste globale Trend ist die immer weiter zunehmende Bedeutung von Informationen und Wissen sowie deren effiziente Nutzung im Postindustriezeitalter. Dies spiegelt sich auch auf gesellschaftlicher Ebene durch die Referenzierung der Gesellschaftsform als „Informations-“ oder „Wissengesellschaft“ wider (vgl. [Probst/Raub/Romhardt 2003], S. 3). Damit geht die Entwicklung einher, dass aufgrund der verbesserten Möglichkeiten zur elektronischen Speicherung und Verfügbarkeit von Informationen sowie deren Vernetzung die Menge an Informationen stark angewachsen ist. Häufig besteht das Problem nicht mehr darin, dass benötigte Informationen nicht verfügbar sind, vielmehr können diese in der unüberschaubaren Flut an Informationen nicht mehr identifiziert und deshalb nicht genutzt werden (vgl. [Feather 1998], S. 118). Durch die starke Abhängigkeit der Unternehmen von der Verfügbarkeit und dem Zugriff auf benötigte Informationen und der daraus resultierenden Anwendung des speziellen Wissens stellt dies eine erfolgsrelevante Situation dar (vgl. [Edmunds/Morris 2000], S. 18 ff.).

Ein weiterer Trend ist die Modularisierung, Dynamisierung und Virtualisierung von Unternehmen, der für diese häufig mit tief greifenden Veränderungen einhergeht. Begriffe wie modulare Unternehmen, Unternehmensnetzwerke und -kooperationen sind heutzutage nicht mehr nur Schlagworte in der betriebswirtschaftlichen Literatur, sondern bereits gelebte Realität (vgl. z. B. [Picot/Reichwald/Wigand 2001], S. 2 ff.). Das Potenzial und die Bedeutung dieser Veränderungen stellen die Unternehmen vor neue Herausforderungen. Durch die Schaffung neuer Formen der teambasierten Zusammenarbeit und Gruppenorganisation wurde diesen auf nichttechnischer Ebene begegnet. Grundlage ist die flexible und dynamische Zusammenarbeit der Mitarbeiter in sowohl technischen wie nichttechnischen Netzwerken (vgl. [Picot/Reichwald/Wigand 2001], S. 11). Ansatzweise wird diese Form der Zusammenarbeit bereits innerhalb von Unternehmen praktiziert, für unternehmensübergreifende Teams kommt ihr jedoch ein wesentlicher Neuheitsgrad zu. Die Etablierung intra- wie interorganisationaler Teamstrukturen macht eine grundsätzliche Neuausrichtung der Unternehmen notwendig. Der Fokus liegt nicht mehr nur auf einzelnen Mitarbeitern, sondern auf Teams. Die Komplexität der Aufgaben wächst bei unternehmensübergreifender Zusammenarbeit durch die notwendige Überbrückung von Unternehmensgrenzen weiter an.

Unternehmensportale, als eine Ausprägung moderner Informations- und Kommunikationstechnologien (IuK-Technologien), sind in den letzten sechs Jahren mit dem Ziel entwickelt worden, einen wesentlichen Beitrag zur Lösung des skizzierten Problems der Informationsüberflutung zu leisten (vgl. [Dias 2001], S. 273). Sie sind als zentrales Werkzeug zur Navigation innerhalb der unternehmensinternen und -externen Informationsbestände sowie zum Zugriff auf die Anwendungen des Unternehmens konzipiert worden. Sie bieten den Mitarbeitern den zentralen und personalisierten Zugriffspunkt (Single Point of Access) auf genau die Informationen, Prozesse, Anwendungen und Personen, die sie zur Erbringung ihrer Aufgaben benötigen. Stand zu Beginn der Entwicklung von Portalen der einzelne Mitarbeiter im Zentrum der Betrachtung, so wurden die Portalarchitekturen zusehends um Elemente zur Unterstützung der Kommunikation, Koordination und Kollaboration zwischen Mitarbeitern erweitert (vgl. [Davydov 2001], S. 157). Portale sind folglich eine geeignete IuK-Technologie, um die Benutzer bei der, im Rahmen der Darstellung des zweiten Trends, skizzierten zunehmenden teambasierten Zusammenarbeit zu unterstützen.

In jüngerer Vergangenheit ist zu beobachten, dass sich innerhalb von Unternehmen verstärkt verschiedene Initiativen zur Einführung von Portalen gebildet haben. Zum einen werden unternehmensinterne Portale für die eigenen Mitarbeiter aufgebaut, zum anderen Portale für externe Partner und Kunden. Erste Erfahrungen zeigen, dass sich die Gesamtsituation der Informationsversorgung und Zusammenarbeit durch die Nutzung von Portalen deutlich verbessert hat (vgl. z. B. [Phifer 2004], [Kyte 2003]). In Erwartung dieser Effizienzsteigerung ist in Teilen bereits evident geworden, in Teilen erst zu prognostizieren, dass immer mehr unternehmensinterne und -externe Portale entstehen, die zusammen genutzt werden müssen, um eine vollständige Aufgabenerfüllung zu erreichen (vgl. [Gootzit/Phifer 2003]). Diese Entwicklung widerspricht jedoch der eigentlichen Zielsetzung und dem Grund für die durch Einführung von Portalen erreichten Effizienzsteigerung, nämlich die Schaffung eines einzigen zentralen Zugriffspunkts auf alle Informationen, Prozesse, Anwendungen und Personen.

Aus der in vielen Unternehmen bereits Realität gewordenen Entwicklung, dass Mitarbeiter nicht mehr nur ein einziges, sondern mehrere Portale nutzen müssen, um ihre Aufgaben erfüllen zu können, lässt sich die erste Definition der Zielsetzung der vorliegenden Arbeit ableiten:

Mit welchen Ansätzen ist es möglich, das Potenzial, das Portalen sowohl in intras als auch interorganisationalen Anwendungsfeldern zukommt, effizient zu nutzen, ohne durch die zunehmende Proliferation von Portalen dieses Potenzial gleichzeitig zu zerstören?

Die Ergebnisse der Untersuchung sind in einem Modell abzubilden, welches zu dessen Überprüfung informationstechnisch umgesetzt und in Fallstudien oder Anwendungsszenarien auf seine Praxistauglichkeit hin validiert werden muss. Übergeordnetes Ziel bei der Konzeption eines Modells ist die Erarbeitung von Vorschlägen für zukünftige Standards bzw. Standardisierungsbemühungen. Durch eine Orientierung an diesem Ziel sind Ergebnisse mit größerer Allgemeingültigkeit und damit potenziell weiterer Ausstrahlung möglich als durch bereits von der Grundausrichtung her proprietäre Ansätze. Eine zweite, detailliertere Definition der Zielsetzung dieser Arbeit findet sich als Abschluss des zweiten Kapitels im Abschnitt 2.3.1.

1.2 Wissenschaftstheoretische Ausrichtung der Arbeit

Die in der Wirtschaftsinformatik positionierte Arbeit hat ihre Grundlagen sowohl im Bereich der Betriebswirtschaftslehre als auch im Bereich der Informatik. Sie zielt darauf ab, Fragestellungen aus beiden Bereichen zu beantworten. Als methodologische, wissenschaftstheoretische Grundlage finden die Prinzipien der Sekundärforschung (engl. Desk Research), der Modellentwicklung und der Aktionsforschung (engl. Action Research) Anwendung.

Die von den Prinzipien der *Sekundärforschung* (vgl. z. B. [Gabler 1997], S. 3390) geleitete systematische Auswahl, das Studium und die Analyse von Veröffentlichungen zu unterschiedlichen Forschungsgegenständen bilden das breit angelegte Fundament, auf dem eine gesicherte Bearbeitung der Zielsetzung stattfinden kann. Berücksichtigung finden sowohl wissenschaftliche als auch praxisorientierte Beiträge. Aufbauend darauf ist der Untersuchungsgegenstand genauer zu spezifizieren und im Rahmen der *Modellentwicklung* (vgl. z. B. [Heinrich 1993]) aus der realen Welt in einem Modell abzubilden. Komplexe Betrachtungsgegenstände werden erst durch diesen Schritt, in dem eine Abstraktion und Fokussierung auf deren wesentliche Aspekte erfolgt, operationalisierbar und öffnen sich der wissenschaftlichen Untersuchung. Im Sinne eines Dialogs zwischen Forschung und Praxis ist es einerseits wünschenswert, dass die Ergebnisse der Sekundärforschung und der Modellbildung in einer in der Praxis nutzbaren Form umgesetzt werden. Hierzu bietet sich die Realisierung in prototypischen Entwicklungen an. Andererseits sind bereits die Modelle an den Erfordernissen der Praxis auszurichten. Dieser Wirkungs- und Regelkreis aus Praxis und Forschung, der auch durch die Rückkopplung der beim praktischen Einsatz der prototypischen Entwicklung gesammelten Erfahrungen in die Modellbildung entsteht, ist Bestandteil der *Aktionsforschungs-Methodik* (vgl. z. B. [Whyte et al. 1991] und [Ulrich/Hill 1979]).

1.3 Aufbau der Arbeit

Im zweiten Kapitel werden die in Abschnitt 1.1 dargestellten globalen Trends aufgegriffen und im Sinne der Fundamentbildung für die weitere Arbeit detaillierter ausgeführt. Betrachtungsgegenstände sind aus mehrheitlich betriebswirtschaftlicher Sicht aktuelle Entwicklungen im Unternehmensumfeld (Abschnitt 2.1). Hierbei wird speziell auf Unternehmensformen zwischen Hierarchie und Markt, mit diesen in Zusammenhang stehende Fragestellungen sowie auf wissensintensive Unternehmensformen eingegangen. Als Paradigma eines mehrheitlich informationstechnischen Ansatzes für die im betriebswirtschaftlichen Bereich genannten Betrachtungsgegenstände werden Portale eingeführt (Abschnitt 2.2). Neben den Eigenschaften, die diese zur Lösung der Herausforderungen besonders geeignet erscheinen lassen, wird gleichermaßen herausgearbeitet, welche ebenfalls in Abschnitt 1.1 angedeuteten Probleme mit der zunehmenden Verbreitung von Portalen einhergehen. Hieraus ergibt sich die Darstellung der Forschungs- und Praxislücke, die Grundlage und Motivation für die vorliegende Arbeit ist. Das zweite Kapitel schließt aufbauend auf den gemachten Ausführungen mit der Konkretisierung des Forschungsziels – der Föderierung von Portalen – und der Betrachtung angrenzender Forschungsansätze (Abschnitt 2.3).

Im dritten Kapitel wird basierend auf der zuvor hergeleiteten Zielsetzung die im zweiten Kapitel begonnene Fundamentlegung fortgesetzt. Die Betrachtung der Ursprünge der Bedeutung des Begriffs der Föderation (Abschnitt 3.1) und dessen Verwendung in der Betriebswirtschaftslehre (Abschnitt 3.2) gibt erste Aufschlüsse über mögliche Aspekte der weiteren Untersuchungen. Die Beschäftigung mit der Verwendung von Föderationen im Gebiet der Informationstechnologie (vgl. Abschnitt 3.3) – hier speziell im Zusammenhang mit föderierten Datenbanken, Informationssystemen und Benutzeridentitäten – bereitet schlussendlich das Feld für die spätere umfassende Analyse der mit der Föderierung von Portalen verbundenen Fragestellungen.

Der Fokus des vierten Kapitels ist im Rahmen der Modellbildung die Konzeption einer Architektur zur Föderierung von Portalen. Diese erfolgt unter Einbeziehung des übergeordneten Ziels, Vorschläge für zukünftige Standards im Bereich föderierter Portale zu erarbeiten. Zu Beginn wird eine hierzu geeignete Entwurfsmethode identifiziert (Abschnitt 4.1) und die Basiskomponenten und Dienste eines Portals vorgestellt (Abschnitt 4.2). Aufbauend auf den Erkenntnissen der beiden vorgelagerten Hauptkapitel findet eine Operationalisierung des Verständnisses gekoppelter und föderierter Portale statt (Abschnitt 4.3). Orthogonal hierzu wird ein Kontinuum der Portalkopplung entwickelt, das verschiedene bei der Modellierung zu

berücksichtigende Stufen der Portalkopplung beschreibt. Die Ableitung von sowohl betriebswirtschaftlichen als auch technischen Anforderungen an die Portalkopplung schließt den die eigentliche Modellierung vorbereitenden Teil ab (Abschnitt 4.4). Diese Ableitung trägt der Forderung Rechnung, bereits bei der Modellbildung Anforderungen aus der Praxis zu berücksichtigen. Die Modellierung der Portalkopplung setzt sich zusammen aus der Darstellung einer detaillierten Kommunikations- und Synchronisationsarchitektur (vgl. Abschnitt 4.5) sowie einem Daten- und Funktionsmodell (vgl. Abschnitt 4.6). Wesentliche Aspekte des Kommunikations- und Synchronisationsmodells bilden Strategien, welche die Verteilung und die Synchronisation von Portalstrukturen zulassen. Sie beziehen alle Stufen des Kontinuums der Portalkopplung mit ein.

Die informationstechnische Umsetzung der entwickelten Architektur für gekoppelte Portale (Abschnitt 5.2) wird im fünften Kapitel anhand der Erweiterung eines bestehenden Portal-Frameworks unter Anwendung von Web-Service-Technologien vorgenommen (Abschnitt 5.1). Diese Umsetzung ist einerseits Grundlage dafür, das als Ergebnis der Forschung entwickelte Modell in seiner technischen Realisierbarkeit zu validieren. Andererseits steht die Implementierung anschließend für die Evaluierung in der Praxis zur Verfügung.

Das sechste Kapitel stellt ein praktisches Anwendungsszenario vor, in dem die zuvor dargestellte informationstechnische Umsetzung zusammen mit einem Praxispartner eingesetzt und getestet wurde. Die Darstellung der Rahmenbedingungen (Abschnitt 6.1 und 6.2), des Aufbaus und Inhalts der Portalkopplung (Abschnitt 6.3) sowie der ableitbaren Erkenntnisse (Abschnitt 6.4) werden begleitet von der kritischen Betrachtung der Limitationen und Aussagefähigkeit dieser ersten praktischen Anwendung (Abschnitt 6.5).

Das abschließende siebte Kapitel eröffnet durch einen umfangreichen Ausblick (Abschnitt 7.1) Perspektiven auf weitere, in der vorliegenden Arbeit nicht berücksichtigte oder erst durch den praktischen Einsatz aufgeworfene Fragestellungen, die Gegenstand weiterer Forschungen und Entwicklungen sein müssen. Anhand der zusammenfassenden Darstellung der Ergebnisse findet resümierend eine kritische Würdigung der Arbeit statt (Abschnitt 7.2).

2 Abgrenzung des Forschungsumfelds

Im Rahmen dieses einführenden Kapitels wird das in der Einleitung umrissene Forschungsumfeld konkretisiert, die Forschungs- und Praxislücke aufgezeigt sowie das Forschungsziel präzisiert.

Aufbauend auf den in Abschnitt 2.1 vorgestellten mehrheitlich betriebswirtschaftlichen Fragestellungen im Bereich des Wandels von Organisationsformen und der Zusammenarbeit von Unternehmen, gibt Abschnitt 2.2 zunächst eine allgemeine Einführung in die Konzepte von Unternehmensportalen, bevor diese im Hinblick auf die zuvor dargestellten Fragestellungen betrachtet werden. Basierend auf der Formulierung der Forschungs- und Praxislücke wird in Abschnitt 2.3 abschließend das Forschungsziel präzisiert. Dieses wird durch die Darstellung ausgewählter anderer Forschungsansätze im selben und in angrenzenden Forschungsbereichen vervollständigt.

2.1 Aktuelle Entwicklungen im Unternehmensumfeld

Das klassische Bild eines Unternehmens als abgeschlossenes, integriertes, sich an einem physischen Ort befindliches Gebilde ist in weiten Bereichen der Wirtschaft nicht mehr existent. Es herrscht weitgehende Einigkeit darüber, dass die klassischen Grenzen der Unternehmung unscharf zu werden beginnen, dass sie sich nach innen wie nach außen verändern und in Teilen auch vollständig auflösen. Für eine Vielzahl von Unternehmen haben sich in den vergangenen Jahren tief greifende Veränderungen der Wettbewerbsbedingungen ergeben, die von ihnen mehr Flexibilität und Innovationsfähigkeit anstelle von Produktivitätssteigerung durch starre Arbeitsteilung verlangen (vgl. [Picot/Reichwald/Wigand 2001], S. 2 und 9). Einige der Faktoren und ihr Zusammenwirken sind in Abbildung 2-1 dargestellt.

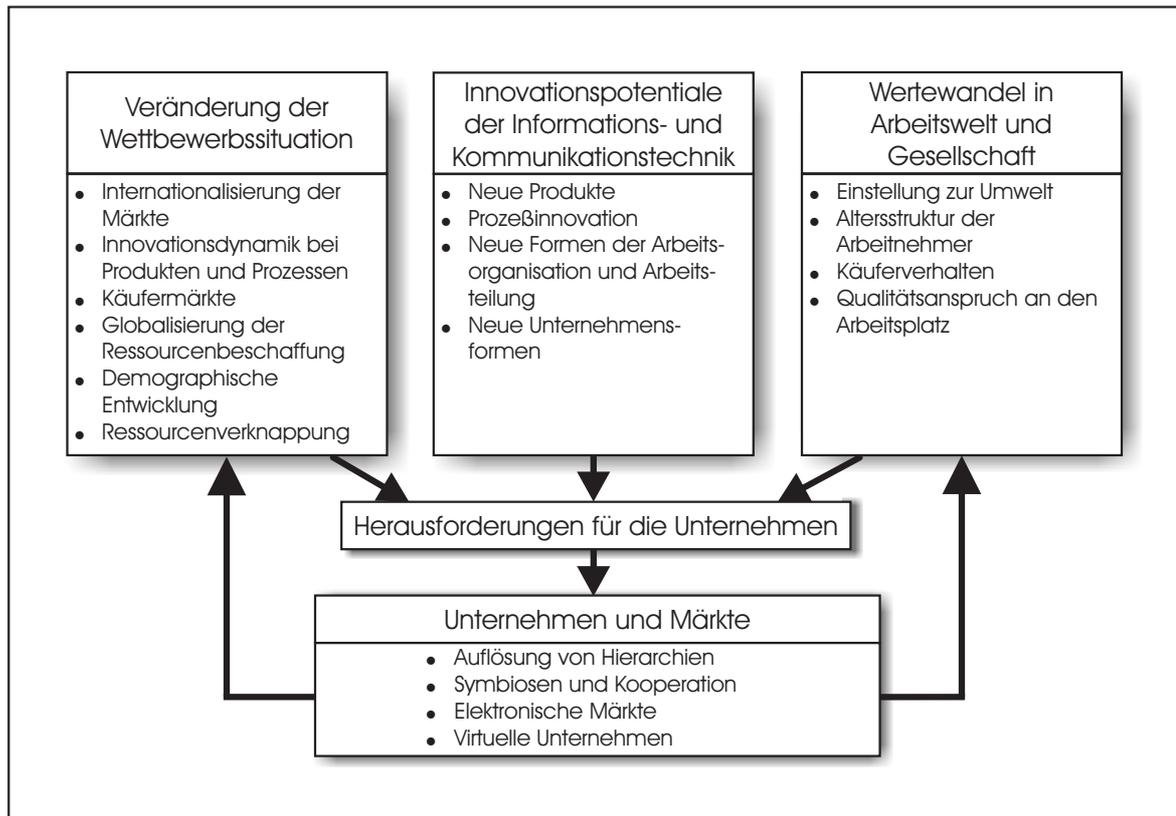


Abbildung 2-1: Faktoren für Veränderungsprozesse von Unternehmen und Märkten
(vgl. [Picot/Reichwald/Wigand 2001], S. 3)

Ein Fokus der Betrachtung liegt im Folgenden auf der durch den Wandel der Rahmenbedingungen ausgelösten Entwicklung von neuen Unternehmensformen, die zwischen den wirtschaftlichen Interaktionsmustern Hierarchie und Markt zu verorten sind. Eine weitere Pointierung aktueller Entwicklungen erfolgt mit der Betrachtung wissensintensiver Unternehmensformen als Konzeptionalisierung der zunehmend wissensgetriebenen Unternehmenswelt.

2.1.1 Unternehmensformen zwischen Hierarchie und Markt

Die Arbeit von Ronald Coase ([Coase 1937]), in der das Problem der ökonomischen Organisation erstmals ausdrücklich als Institutionenvergleich und Folge von Transaktionen und den mit ihnen verbundenen Kosten formuliert wurde, bildet die wesentliche Basis für die bis in die 80er Jahre vorherrschende Dichotomisierung wirtschaftlicher Institutionen in Hierarchie und Markt.

Die von Coase begründete und später von Williamson ([Williamson 1975]) wieder aufgegriffene Transaktionskostentheorie betrachtet die Leistungs- und Austauschbeziehung zwischen Individuen und fasst die damit einhergehende Übertragung von Verfügungsrechten als Transaktion auf. Die mit der Übertragung anfallenden Transaktionskosten, bei denen es sich

in erster Linie um Informations- und Kommunikationskosten handelt (vgl. [Neuburger 1994], S. 54), werden als Effizienzmaßstab zur Beurteilung und Auswahl unterschiedlicher institutioneller Arrangements herangezogen. Die Transaktionskostentheorie sieht die Kosten der Abwicklung durch die drei Merkmale Transaktionshäufigkeit, -unsicherheit und Spezifitätsgrad der Investitionen zur Durchführung der Transaktion determiniert.

Die *Transaktionshäufigkeit* bestimmt maßgeblich die Amortisationszeit und damit die ökonomische Vorteilhaftigkeit hierarchischer oder marktlicher Organisationformen. *Transaktionsunsicherheit* drückt die Möglichkeit unvorhersehbarer Aufgabenänderungen, damit einhergehender Vertragsveränderungen und daraus folgender erhöhter Transaktionskosten aus. Der *Spezifitätsgrad* einer Transaktion beschreibt schließlich, wie stark die Investitionen, die für die Aufgabenerfüllung getätigt wurden, spezifisch für diese Transaktion sind oder inwiefern sie auch für andere Transaktionen eingesetzt werden können. Je höher die Spezifität, desto größer ist die Abhängigkeit vom Transaktionspartner, weshalb die Transaktionskostentheorie in diesen Fällen längerfristige Bindungen empfiehlt (vgl. [Picot/Reichwald/Wigand 2001], S. 50 ff.).¹

Die Erklärungsansätze der Transaktionskostentheorie werden nicht von allen Seiten akzeptiert und verschiedentlich kritisiert (vgl. z. B. [Schneider 1985] oder [Ghoshal/Moran 1996]). Ein wesentlicher Kritikpunkt, der auch durch die Einführung einer dritten, hybriden Form als Strukturalternative zwischen Hierarchie und Markt (vgl. [Williamson 1985]) nicht entkräftet wurde, ist das Festhalten an den als diskret konzeptionalisierten Strukturalternativen.

2.1.1.1 Organisationskontinuum

Um das zuvor dargestellte Problem der diskreten Strukturalternativen zu überwinden, schlägt Ebers eine Konzeptionalisierung entlang der Dimension der Autonomie der Transaktionspartner vor. Dies führt seiner Ansicht nach dazu, dass „marktliche und hierarchische institutionelle Arrangements in der Tat [...] als die beiden Pole eines Kontinuums von institutionellen Gestaltungsformen angesehen werden können“ ([Ebers 1994], S. 31).

¹ Für vertiefende Darstellungen zur Transaktionskostentheorie sei neben den bereits angegebenen Quellen weiterhin z. B. auf Michaelis ([Michaelis 1985]) und Grote ([Grote 1990]) verwiesen.

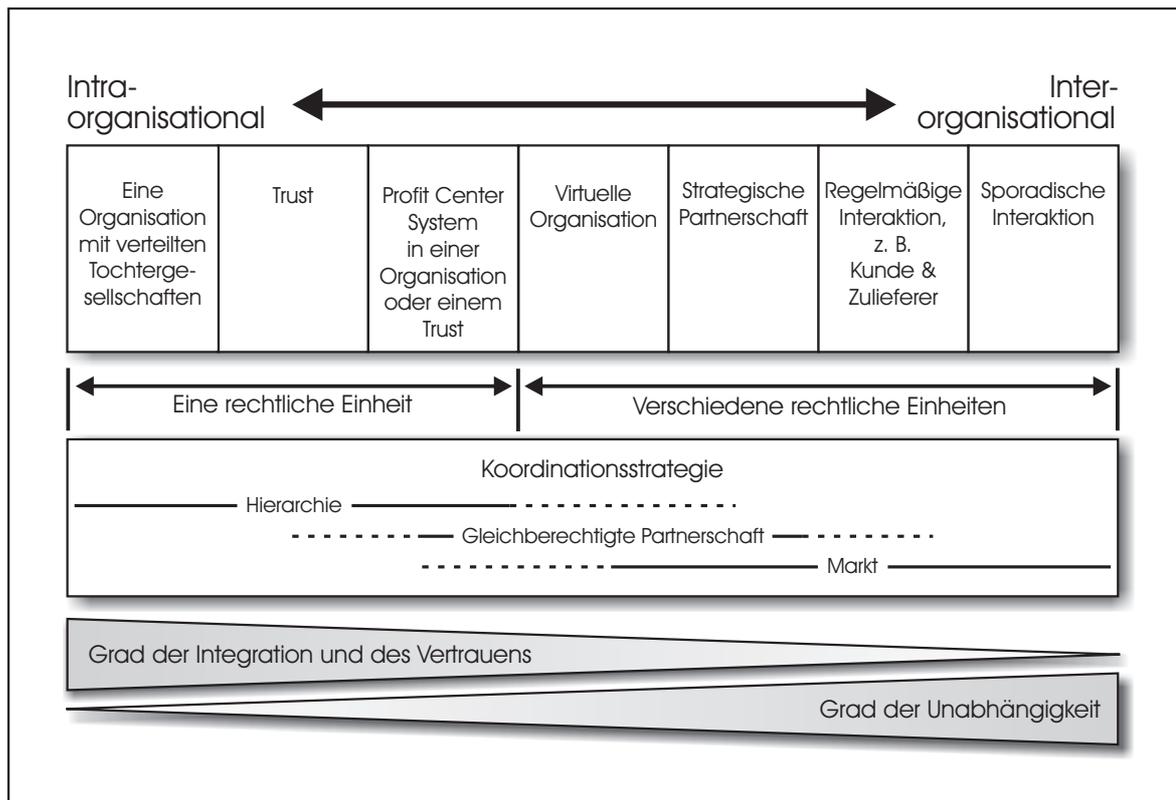


Abbildung 2-2: Kontinuum organisatorischer Gestaltungsformen
(in Anlehnung an [Riemp 1998], S. 115)

Die „neuen“ Organisationsformen zwischen Hierarchie und Markt, die Riemp ([Riemp 1998]) in Form eines Kontinuums visualisiert (vgl. Abbildung 2-2), werden seit Mitte bis Ende der 80er Jahre ausgiebig untersucht und diskutiert. Besonders die Rückbesinnung auf die eigenen, zum strategischen Erfolg beitragenden Kernkompetenzen der Unternehmen und damit einhergehend die Notwendigkeit zur Hinzuziehung notwendiger Komplementärkompetenzen führte auch in der Praxis zur facettenreichen Realisierung von Unternehmenskooperationen. Eine detaillierte Betrachtung der verschiedenen Organisationsformen geht über die Zielsetzung dieser Arbeit hinaus. Der interessierte Leser findet sowohl konkrete Abgrenzungen als auch eine Darstellung der verschiedenen Organisationsformen z. B. bei Faisst ([Faisst 1998]). Im Fokus des Interesses stehen stattdessen in der wissenschaftlichen Diskussion identifizierte globale Trends. Einer dieser Trends ist die zunehmende Entstehung von verteilten bis hin zu virtuellen Organisationstypen, für die eine räumliche und über Zeitzonen hinweg stattfindende zeitliche Verteilung kein Hindernis darstellt, sondern die im Gegenteil dazu in der Lage sind, von diesen neuen Organisationsstrukturen zu profitieren. In der Praxis ist zu beobachten, dass den damit verbundenen neuen Herausforderungen durch Abflachung von Hierarchien und der Bildung von flexiblen Teamstrukturen begegnet wird.

2.1.1.2 Rolle von Information und Informationsüberflutung

In einer zunehmend dynamischen und vernetzten Unternehmenswelt, bei der der Anteil materieller Produkte sinkt, gewinnt Information² als immaterielles Gut eine immer größere Bedeutung. Gleichzeitig ist der Austausch von Informationen die Grundlage jeder Art arbeitsteiligen Handelns, unabhängig davon, ob zwischen kooperierenden Unternehmen oder innerhalb eines Unternehmens (vgl. [Burnett/Brookes-Roones/Keogh 2002], S. 4). Für Edmunds und Morris (vgl. [Edmunds/Morris 2000], S. 18 ff.) ist Information der Schlüssel zum Unternehmenserfolg. Deren elementare Bedeutung erzeugt bei den Mitarbeitern das Gefühl und den Zwang, immer auf dem aktuellsten Informationsstand sein zu müssen. Dies versuchen sie durch die permanente Aufnahme von immer neuen und mehr Informationen zu erreichen. Letzten Endes hängt der private und berufliche Erfolg in einer modernen Gesellschaft unmittelbar mit der Fähigkeit zusammen, große Mengen an neuen Informationen aufzunehmen und einzusetzen. Die im Abschnitt 2.1.1.1 dargestellte Tendenz zu verteilten und virtuellen Organisationsformen und Teams verstärkt die Bedeutung von Informationen weiter. Das Funktionieren von Teams und virtuellen Organisationen ist elementar abhängig vom Vorhandensein, dem Austausch und der Nutzung von Informationen über verschiedenste Medien und Kanäle.

Obwohl der einfachere und umfassendere Zugriff auf Informationen offensichtliche Vorteile bietet, haben die Forschungen der vergangenen Jahre ergeben, dass die Informationsüberflutung zu Stress, geringerer Zufriedenheit mit der Arbeit und Krankheit führen kann (vgl. [Lewis 1996]). Die Literatur zu diesem Forschungsgebiet beschreibt die Situation, in der wir uns befinden, als „infoglut“, sieht uns umgeben von „data smog“, stellt das „information fatigue syndrome“ fest und hat das Paradoxon erkannt, dass, obwohl wir bereits zu viele Informationen haben, jedoch nicht diejenigen finden, die wir wirklich benötigen.

Wilson definiert Informationsüberflutung als „being presented with more information than can be absorbed, being burdened by a large supply of information, that can not be assimilated due to lack of time“ ([Wilson 1995], S. 45). Feather (vgl. [Feather 1998], S. 118) beschreibt diesen Zustand als Punkt, an dem so viel Information verfügbar ist, dass es nicht mehr möglich ist, diese effektiv zu nutzen. Ein zentrales Konzept bei der Beschäftigung mit dem Thema Informationsüberflutung wird in der Relevanz von Informationen für den aktuellen Kontext gesehen. Aufgrund der großen Menge an Informationen, die den Mitarbeitern eines Unternehmens zur Verfügung stehen, wird es immer schwieriger, diejenigen zu identifizieren

² Zu einer Begriffsdefinition siehe z. B. [Probst/Raub/Romhardt 2003], S. 16 f.

und auszufiltern, die für die Organisation selbst oder die aktuelle Aufgabe wirklich relevant und zweckdienlich sind (vgl. [Franz 1999], S. 5). In einer Feldstudie haben Watson-Mannheim und Bélanger unter anderem untersucht, welche Faktoren zur Informationsüberflutung speziell in verteilten, virtuellen Teams führen. Als Gründe für eine stark erhöhte Kommunikation und den damit einhergehenden Effekt der Informationsüberflutung identifizieren sie besonders mangelndes Vertrauen zwischen den Teammitgliedern und Probleme, nicht zum direkten Team gehörende externe Mitarbeiter über den Fortschritt und die eigene Leistung zu informieren (vgl. [Watson-Manheim/Bélanger 2002], S. 77).

Vorschläge zur Lösung des immer dringlicher werdenden Problems berücksichtigen sehr unterschiedliche Aspekte. Sie reichen von der grundsätzlichen Beschäftigung mit den Möglichkeiten zum Aufbau von Vertrauen und damit der Reduktion redundanter Kommunikation bis zu intensiv diskutierten Fragen, wie z. B. mithilfe intelligenter Agenten, Push-Technologien oder Information-Workern die weiterhin wachsende Informationsmenge sinnvoll gefiltert werden kann. Als Ausgangspunkt für die weitere Vertiefung des Themas durch den Leser kann der Beitrag von Edmunds und Morris ([Edmunds/Morris 2000]) genutzt werden, der eine Überblicksdarstellung sowie zahlreiche weiterführende Referenzen enthält.

2.1.1.3 Informationstechnologie als „Enabler“

Die vorangegangenen Überlegungen haben die fundamentale Bedeutung von Information und Kommunikation für wirtschaftliches Handeln dargestellt. Es wurde ebenfalls deutlich, dass nicht die absolute Zunahme der Information maßgeblich für die durch sie zu erzielende Steigerung des Geschäftserfolgs ist. Von wesentlicher Bedeutung ist vielmehr die nutzbare Bereitstellung der tatsächlich benötigten Informationen zu ökonomisch vertretbaren Kosten.

Der Einsatz von Informations- und Kommunikationssystemen (IuK) ist dazu geeignet, die Koordination und Kommunikation in Unternehmen zu verbessern. Das größere Potenzial kommt ihnen jedoch aufgrund der wesentlich größeren Koordinationsprobleme zwischen Organisationen zu (vgl. [Nooteboom 1999], S. 171). Für vernetzte Unternehmen ermöglichen erst IuK-Systeme ganz grundsätzlich genügend effiziente Koordinationsformen zur Nutzung von Effektivitätsvorteilen gegenüber integrierten Unternehmen. Immer häufiger wird dem unternehmensübergreifenden Einsatz von Informationssystemen sogar das Potenzial zugeschrieben, einer Organisationsform ökonomischer Aktivitäten den Weg zu bereiten, die mit dem Begriff der virtuellen Unternehmung belegt wird (vgl. [Sydow/Winand 1998], S. 17 f.).

In der diesbezüglichen wissenschaftlichen Literatur herrscht weitgehende Einigkeit darüber, dass durch Informationstechnologie (IT) ein effizienter Austausch und die gezielte Verteilung von Informationen auch über die Grenzen der Unternehmung hinaus möglich sind. IT wird aus diesem Grund als „Enabler“ für kooperative Organisationsformen identifiziert (vgl. z. B. [Klüber/Alt/Österle 2000], S. 44). Da die Eigenschaften und Nutzungsformen von IT sehr unterschiedlich sind, sieht Rupprecht-Däullary (vgl. [Rupprecht-Däullary 1994], S. 127) es jedoch als wenig Erfolg versprechend an, exakte Ursache-Wirkungs-Zusammenhänge ableiten zu wollen. Sie spricht sich stattdessen für die Identifikation genereller Tendenzen und Wirkungsrichtungen aus. In diesem und im Sinne der in Abschnitt 2.1.1.1 vorgestellten Transaktionskostentheorie kann festgestellt werden, dass IT grundsätzlich zur Senkung der Transaktionskosten beitragen kann (vgl. z. B. [Picot/Reichwald/Wigand 2001], S. 71 und [Nootboom 1999], S. 212).

Neben dieser Eigenschaft kann IT aber auch Abhängigkeiten schaffen, wobei besonders Abhängigkeiten von den Marktpartnern von Bedeutung sind. Spezifische IT-Systeme sind auf die speziellen Anforderungen und Eigenschaften der Marktbeziehung abgestimmt, ein Wechsel des Marktpartners würde negative Effekte, z. B. zeitlicher oder finanzieller Art, verursachen. Ausgehend davon, dass das eigene Unternehmen von einem eventuellen Partnerwechsel (ebenfalls) negativ betroffen wäre, verringern sich durch IT-Systeme auch die Handlungsmöglichkeiten und damit die Flexibilität des Unternehmens (vgl. [Rupprecht-Däullary 1994], S. 132).

Diesem Effekt, der als „Switching Costs“ oder „Lock-In“ bezeichnet wird, kann durch die Verringerung der Spezifität der für die Transaktionen benötigten IT-Systeme begegnet werden. Dies kann beispielsweise dadurch erreicht werden, dass die IT-Systeme so flexibel gestaltet werden, dass sie auch für andere Szenarien und Partner einsetzbar sind. Dem steht aktuell gegenüber, dass große Teile der IT noch nicht ausreichend standardisiert sind oder verschiedene Partner verschiedene konkurrierende Standards einsetzen. Dadurch entstehen wieder spezifische Investments, die ein Steigen der Transaktionskosten zur Folge haben (vgl. [Nootboom 1999], S. 123 und 167; siehe auch die Ausführungen zum Thema Spezifitätsgrad von Transaktionen in Abschnitt 2.1.1).³

³ Weiterführende Darstellungen zu diesem Thema finden sich z. B. in [Clemons/Reddi 1993], [Picot/Reichwald/Wigand 2001], [Faisst 1998], [Gurbaxani/Whang 1991] und [Ebers 1994].

2.1.1.4 Vertrauen

„Als elementares Organisationsprinzip zwischenmenschlicher Austauschbeziehungen spielt Vertrauen auch und gerade bei der Organisation wirtschaftlicher Leistungsbeziehungen eine zentrale Rolle.“ ([Picot/Reichwald/Wigand 2001], S. 123)

Vertrauen wird in der Literatur als eines der konstituierenden Merkmale von verteilten Unternehmensstrukturen und Formen der (virtuellen) unternehmensübergreifenden Zusammenarbeit identifiziert (vgl. z. B. [Powell 1996], [Loose/Sydow 1994]). Nur die Rückbesinnung auf eine Organisationsführung, die weniger durch Kontrolle, sondern mehr durch Vertrauen geprägt ist, ist dazu in der Lage, die Vorzüge virtueller Organisationen nutzbar zu machen. Virtualität erfordert ein großes Maß an Vertrauen, damit sie funktioniert, Technologie alleine ist nicht ausreichend (vgl. [Handy 1995], S. 5).

Nemiro ([Nemiro 2000], S. 105) unterscheidet drei Komponenten, aus denen sich Vertrauen zusammensetzt: a) *Vertrauen in Personen*, die Mitglieder müssen untereinander der Kompetenz und Zuverlässigkeit des anderen vertrauen; b) *Vertrauen in Aufgaben und Ziele*, durch das Verfolgen gemeinsamer Aufgaben und Ziele wird Vertrauen unter den Mitgliedern geschaffen und c) *Vertrauen in Informationskanäle*, die Mitglieder müssen darauf vertrauen, dass die Informationen, die sie über verschiedene Informationskanäle erhalten, die besten verfügbaren Informationen sind und die Informationskanäle eine effiziente Möglichkeit darstellen, auf diese zuzugreifen.

Zahlreiche Autoren (vgl. z. B. [De Laat 1999], [Köszegi 2001] und zu einer Übersicht über weitere Arbeiten [Ring 1999]) beschäftigen sich mit verschiedenen Aspekten wie der Relevanz und der Schaffung von Vertrauen, möglichen Surrogaten für Vertrauen, Absicherungsmaßnahmen bei der Verletzung von Vertrauen und weiteren sich mit Vertrauen befassenden Fragestellungen. Für die vorliegende Arbeit stellen einerseits das Vertrauen selbst, andererseits der Grad des Vertrauens determinierende Faktoren dar, die maßgeblichen Einfluss auf die Art und das Ausmaß der Zusammenarbeit in und zwischen Unternehmen haben. Vertrauen wird jedoch bei den weiteren Betrachtungen als exogener Faktor angesehen und ist daher selbst nicht weiter Gegenstand der Untersuchung.

2.1.1.5 Zusammenfassung

Die vorstehenden Ausführungen haben deutlich gemacht, dass das Unternehmensumfeld sich in den vergangenen Jahren stark dynamisiert und flexibilisiert hat. Zwischen Hierarchie und Markt sind zahlreiche weitere Organisationsformen entstanden, bei denen die Kooperation

zwischen verschiedenen weitgehend selbstständigen Organisationseinheiten bzw. ganzen Organisationen im Vordergrund steht und die durch räumlich und zeitlich verteilte, teilweise virtuelle teambasierte Arbeitsteilung gekennzeichnet sind.

Kooperationen erfordern eine verstärkte Kommunikation und Koordination zwischen den Beteiligten. Dies führt zu einem erhöhten Informationsbedarf, wobei der Faktor Information an sich für alle Unternehmen eine immer höhere Bedeutung erlangt. Informationstechnologie kann auf der einen Seite helfen, diesen Bedarf zu ökonomisch sinnvollen Bedingungen zu befriedigen, und ermöglicht so teilweise erst die Realisierung einiger der neuen Organisationsformen. Auf der anderen Seite kann IT aber auch zu Informationsüberflutung beitragen und so negative Effekte verursachen. Darüber hinaus birgt der Einsatz von kooperations-spezifischen IT-Systemen die Gefahr, sich in Abhängigkeiten zu begeben, die ebenfalls zu negativen Effekten führen können.

Neben der IT spielen zahlreiche weitere, besonders soziokulturelle Faktoren eine entscheidende Rolle, die zum Gelingen von Kooperationen beitragen. Hierzu zählt besonders der zuvor thematisierte Aspekt des Vertrauens. Ohne Vertrauen zwischen den Beteiligten ist die Effektivität des IT-Einsatzes deutlich eingeschränkt. Einerseits werden bei fehlendem Vertrauen keine ausreichenden oder qualitativ hochwertigen Informationen in die IT-Systeme eingestellt; andererseits wird den in ihnen verfügbaren Informationen keine ausreichende Glaubwürdigkeit zugestanden.

Zusammengefasst bedeutet dies, dass IT für die „neuen“ Organisationsformen eine notwendige, aber keine hinreichende Bedingung ist und dass die IT sowohl flexibel als auch standardisiert sein sollte, so dass

- a) der Auf- und Abbau der Verbindung zwischen Kooperationspartnern schnell und unter Kostenaspekten sinnvoll möglich ist, um der zunehmenden Flexibilisierung und Dynamisierung der Wirtschaftswelt Rechnung tragen zu können,
- b) die IT-Systeme sich flexibel parametrisieren lassen, um mit verschiedenen Kooperationspartnern eingesetzt werden zu können und einen „Lock-In“-Effekt zu vermeiden, und
- c) die IT-Systeme dazu beitragen müssen, die Informationsflut zu kanalisieren und im Idealfall nur die Informationen an die Benutzer zu verteilen, die sie für den aktuellen Arbeits- und Kooperationskontext benötigen.

2.1.2 Wissensintensive Unternehmensformen

Der Wandel von einer Industriegesellschaft hin zu einer Informationsgesellschaft und nun zu einer Wissensgesellschaft ist zwar noch nicht abgeschlossen, gleichwohl hat er, wie auch im Abschnitt 2.1.1 dargestellt, bereits jetzt weit reichende Auswirkungen auf Unternehmen und deren Umfeld. Ein weiterer Aspekt dieses Wandels wird in der wissenschaftlichen Literatur unter den Begriffen „knowledge-intensive firms“, „knowledge-based organizations“ oder auch „knowledge-based theory of the firm“ untersucht. Diese bezeichnen denselben Ansatz, deshalb werden die Begriffe meist synonym verwendet. Im Weiteren wird die „wissensintensive Unternehmung“, als der Begriff, der sich im deutschsprachigen Raum durchgesetzt hat, gewählt.

2.1.2.1 Definition und Abgrenzung

Wissensintensive Unternehmen sind im letzten Jahrzehnt Gegenstand umfangreicher Untersuchungen gewesen. Ditillo ([Ditillo 2000]) stellt eine bemerkenswerte Heterogenität in der diesbezüglichen wissenschaftlichen Literatur fest. Dies manifestiert sich auch in den zahlreichen Definitionen, von denen hier beispielhaft nur einige wiedergegeben werden.⁴

- “Knowledge-intensive firms create innovative solutions for the market by integrating the knowledge of their individuals.” ([Ditillo 2000])
- “One of the most distinctive characteristics of knowledge-based organizations (KBOs) is that they have only the expertise of their staff as assets with which to trade.” ([Winch/Schneider 1993], S. 923)
- “Knowledge-intensive firms process what they know into knowledge products and services for their customers. [...] They are less capital-intensive than firms in the manufacturing industry and more learning-intensive than other service industries.” ([Nurmi 1998], S. 26)
- “The term *knowledge-intensive* imitates economists’ labelling of firms as capital-intensive or labour-intensive. These labels describe the relative importance of capital and labour as production inputs. [...] By analogy, labelling a firm as knowledge-intensive implies that knowledge has more importance than other inputs.” ([Starbuck 1992], S. 715)

Die Beschäftigung mit wissensintensiven Unternehmen und damit in Verbindung stehenden Konzepten ist nicht unproblematisch. Die Unterscheidung in wissensintensive und nicht wissensintensive Unternehmen ist keineswegs per se selbsterklärend, da alle Organisationen Wissen⁵ benötigen und einbeziehen. Einen Anhaltspunkt bietet die praktisch allen Definitionen gemeinsame, teils offen ausgesprochene, teils zumindest implizit enthaltene Abgrenzung wissensintensiver Unternehmen gegenüber klassischen Unternehmen derart, dass die

⁴ Eine Übersicht über das Forschungsfeld und weitere Definitionen findet sich z. B. in [Ditillo 2000].

⁵ Zu einer Begriffsdefinition siehe z. B. [Probst/Raub/Romhardt 2003], S. 16 f.

Mitarbeiter und deren Wissen im Mittelpunkt der Betrachtung stehen und das wesentliche Kapital wissensintensiver Unternehmen sind. Auch wenn es durchaus schwer fällt, eine konkrete Abgrenzung anhand dieses Kriteriums vorzunehmen, handelt es sich dabei um für den Kontext der Arbeit relevante Aspekte. Alvesson kommt trotz der Kritik zu dem Schluss, dass es sinnvoll ist, wissensintensive Unternehmen als hinsichtlich der Zuordnung zwar vage, aber hinsichtlich der Bedeutung aussagekräftige Differenzierung zu verwenden. Die Beschäftigung mit ihnen stellt die Betrachtung der genannten Aspekte in den Vordergrund, die sonst häufig weitgehend unberücksichtigt bleiben (vgl. [Alvesson 2000], S. 1102 f.).

2.1.2.2 Teams als elementare Organisationsstruktur

Wissensintensive Unternehmen zeichnen sich dadurch aus, dass sie stark auf Selbstbestimmung und Eigenverantwortung beruhen, auf abgeflachten Hierarchien und der besonders ausgeprägten Notwendigkeit zur Kommunikation, zur Koordination und zur Problemlösung. Formale Koordination allein ist in wissensintensiven Unternehmen unzureichend. Die Wissensarbeiter sind auf ein loses Netz von anderen Wissensarbeitern und ihre Arbeitsumgebung angewiesen. Um trotz der geringen, aber hoch flexiblen inneren Struktur und weitgehenden Unabhängigkeit der einzelnen Wissensarbeiter ein Mindestmaß an Koordination zur Aufrechterhaltung des Unternehmens gewährleisten zu können, ist ein besonders hohes Maß an Kommunikation notwendig. Das interne Koordinations- und Kommunikationsproblem wird dadurch, dass die Organisationsgrenzen immer weniger klar definiert sind und Kunden, Partner und selbst Konkurrenten zusammen ein sich stetig in Veränderung befindliches Netz bilden, zunehmend auch auf den interorganisationalen Bereich ausgedehnt (vgl. [Nurmi 1998], S. 28 f.).

Die Vertreter der „knowledge-based theory of the firm“ (vgl. z. B. [Conner/Prahalad 1996], [Grant 1996], [Grant 1997], [Kogut/Zander 1996]) schlagen für wissensintensive Unternehmen daher verstärkt teambasierte Strukturen vor. Die Mitgliedschaft in den Teams ist hoch flexibel und hängt unmittelbar mit dem für die Erfüllung der Aufgabe benötigten Wissen zusammen. Diese hohe Flexibilität teambasierter Organisationen ist in besonderem Maße dazu in der Lage, das auf die Wissensarbeiter verteilte Spezialistenwissen in geeigneter Weise zusammenzuführen und für Innovationen nutzbar zu machen (vgl. [Ditillo 2000], S. 12). Die vor allem organisationstheoretisch orientierte Literatur auf dem Gebiet der wissensintensiven Unternehmen lässt allerdings die Frage weithin unbeantwortet, wie diese Teams in ihrer Arbeit unterstützt werden können, um ihren Bedarf an Kommunikation, Kollaboration und Koordination zu decken.

2.1.2.3 Zusammenfassung

Obwohl die Erforschung wissensintensiver Unternehmen kein einheitliches Bild vermittelt und nicht ohne Kritik geblieben ist (vgl. z. B. [Foss 1996a], [Foss 1996b]), gibt sie Hinweise auf wichtige Aspekte, deren Bedeutung über wissensintensive Unternehmen hinausgeht. Die Betrachtung wissensintensiver Unternehmen ist so lange als irrelevanter Ansatz anzusehen, wie sie ausschließlich als Möglichkeit zur Abgrenzung zwischen verschiedenen Typen von Unternehmen verwendet wird. Die Relevanz des Konzepts wissensintensiver Unternehmen als eigenständiger Untersuchungsgegenstand ist nur dann gegeben, wenn die Betrachtung dazu benutzt wird, den Fokus und das Interesse auf das Finden von Lösungen im Hinblick auf die Generierung, Nutzung und das Management von Wissen in Organisationen zu lenken. Dies trifft insbesondere auf stark mitarbeiterzentrierte Organisationen zu, hat aber auch eine breitere Ausstrahlung, da alle Organisationen tendenziell wissensintensiver werden (vgl. [Ditillo 2000], S. 6 f.).

Unter diesem Blickwinkel ist das Konzept der wissensintensiven Unternehmung für die vorliegende Arbeit insofern bedeutsam, als es die Interaktion und die Koordination zwischen Personen eines oder mehrerer Unternehmen und die dazu notwendigen Maßnahmen wie z. B. Informationsselektion und -filterung sowie Kommunikation in den Vordergrund stellt. Zunehmende Vernetzung in verteilten, inter- und intraorganisationalen Teamstrukturen wird von der überwiegenden Anzahl von Autoren in diesem Bereich als Möglichkeit angesehen, diesen Herausforderungen zu begegnen. Damit besteht für diesen Bereich eine ausreichende Trennschärfe dahingehend, dass nicht auf die Automatisierung und massenhafte Verarbeitung von Informationen, sondern auf die interpersonelle Schaffung und Nutzung von Informationen und Wissen abgezielt wird.

2.2 Unternehmensportale

Unternehmensportale stellen gleichermaßen ein Konzept und eine Technologie dar, die verspricht, die Herausforderungen, die in den vorangegangenen Abschnitten dargestellt wurden, zu bewältigen.

Da Unternehmensportale bereits seit mehr als sechs Jahren Gegenstand der Diskussion sind, werden zuerst der Status quo und die Entwicklungsgeschichte von Unternehmensportalen überblicksartig dargestellt und deren Einsatzbereiche von anderen Gebieten abgegrenzt. Daran anschließend findet die detaillierte Betrachtung von drei ausgewählten Aspekten von

Portalumgebungen statt, die in der Identifikation der Forschungs- und Praxislücke mündet und auf die bei der Definition des Forschungsziels zurückgegriffen wird.

2.2.1 Grundlagen

2.2.1.1 Historie

Mit der weiten Verbreitung des World Wide Web (WWW) und allen Vorteilen, die diese neue Form der Informationsbereitstellung und des Zugriffs auf Informationen mit sich brachte, entstand zugleich das Problem, dass die Benutzer in der Fülle an Informationen den Überblick verloren und der Nutzwert des Internets gefährdet war. Dies führte zur Entwicklung und Einführung von Suchmaschinen und Katalogen, welche die Orientierung und das Auffinden von Informationen vereinfachen und beschleunigen sollten. Informationsangebote, die um Dienste wie z. B. themen- und interessenbasierte Gemeinschaften, Echtzeitkommunikationsfunktionen, personalisierte Informationen etc. erweitert wurden, erhielten schließlich die Bezeichnung *Internetportale*. Reynolds und Koulopoulos sehen die Entwicklung in der wiederkehrend erhobenen Forderung nach einem zentralen Zugriffspunkt begründet, von dem aus den Benutzern der Zugriff auf alle Internetquellen möglich ist, die sie zur Befriedigung ihrer Informationsbedürfnisse benötigen (vgl. [Reynolds/Koulopoulos 1999]).

Auch innerhalb von Unternehmen wuchs die Menge an verfügbaren Informationen stark an, was deren effektiven Einsatz jedoch immer stärker gefährdete. Die fortschreitende Entwicklung und der Erfolg von Internetportalen blieben daher auch den Unternehmen nicht verborgen. Sie begannen das Potenzial zu erkennen, das damit verbunden war, die gleichen Prinzipien und Technologien zum Management, zur Strukturierung und zum Zugriff auf unternehmensinterne Informationsbestände anzuwenden (vgl. [Dias 2001], S. 273). Obwohl die *Unternehmensportale* aus der Idee, die den Internetportalen zugrunde liegt, entstanden sind, stellen Reynolds und Koulopoulos zu Recht fest, dass es sich bei Internet- und Unternehmensportalen um zwei vollkommen unterschiedliche Zweige der Evolution von Portalen handelt. Diese erfüllen unterschiedliche Zwecke für unterschiedliche Benutzergruppen.

2.2.1.2 Klassifizierung

Die erstmalige Verwendung des Portalbegriffs im Unternehmensumfeld ist auf einen Merrill-Lynch-Report von Shilakes und Tylman zurückzuführen, in dem sie schreiben:

“Enterprise Information Portals are applications that enable companies to unlock internally and externally stored information, and provide users a single gateway to personalized information needed to make informed business decisions.” ([Shilakes/Tylman 1998])

Weiterhin sehen sie Enterprise Information Portale (EIP) als

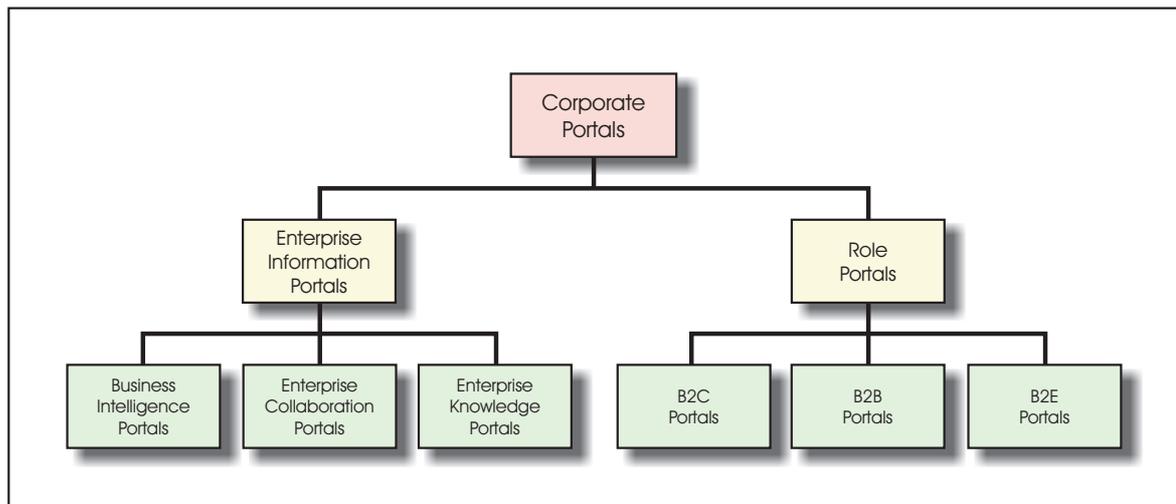
“an emerging market opportunity; an amalgamation of software applications that consolidate, manage, analyze and distribute information across and outside of an enterprise.“ ([Shilakes/Tylman 1998])

Ihre weiteren Ausführungen kennzeichnen EIP als aus den zwei Systemkategorien „decision-making support“ und „collaborative processing“ bestehend.

Auf Basis dieser initialen Definition sind verschiedene Klassifikationen vorgeschlagen worden, die auf dem Prinzip weitgehend aufeinander aufbauender Stufen und einem damit einhergehend unterschiedlichen Funktionsumfang gründen (vgl. z. B. [White 1999], [Murray 1999], [Collins 2001], [Davydov 2000], [Davydov 2001], [Firestone 2003] und für eine Überblicksdarstellung [Dias 2001]). Firestone ([Firestone 2003]) sieht dies als einen hauptsächlich politisch motivierten Prozess, der darauf abzielt, den Portalbegriff in der einen oder anderen Weise zu definieren, um die eigene Portallösung als umfassender oder – aus Sicht eines Beratungsunternehmens – die eigene Methodik als die bessere darzustellen.

Eine Art der Klassifizierung zeichnet sich dadurch aus, dass eine Differenzierung hinsichtlich der Gewichtung der beiden Aspekte Entscheidungsunterstützung und Unterstützung von Zusammenarbeit vorgenommen wird. Abhängig vom Blickwinkel des Autors werden in diesem Zusammenhang in geradezu inflationärem Ausmaß weitere Begriffe eingeführt wie „Corporate Portal“, „Corporate Information Portal“, „Business Portal“, „Workplace Portal“, „Enterprise Collaborative Portal“, „Business Intelligence Portal“ oder „Enterprise Knowledge Portal“. Diese werden teilweise synonym verwendet und verfolgen das Ziel, jeweils den einen oder anderen Aspekt mehr oder weniger stark zu betonen.

Eine andere Klassifizierung orientiert sich an den Zielgruppen bzw. Rollen der Benutzer des Portals (vgl. z. B. [Davydov 2001], [Schelp/Winter 2002], [Bauer 2001]). Zur Unterstützung des Geschäftsverkehrs zwischen Unternehmen werden Business-to-Business(B2B)-Portale eingesetzt, wohingegen Business-to-Consumer(B2C)-Portale auf die Kommunikation mit den (Privat-)Kunden zielen. Business-to-Employee(B2E)-Portale richten sich an die eigenen Mitarbeiter.



**Abbildung 2-3: Klassifizierung von Unternehmensportalen
(in Anlehnung an [Davydov 2001], S. 138)**

Neben diesen beiden in Abbildung 2-3 grafisch dargestellten und am weitesten verbreiteten Ansätzen zur Klassifikation von Unternehmensportalen lassen sich in der Literatur weitere Ansätze z. B. nach Anwendungsbereichen (vertikal, horizontal) oder nach Reichweite (intern, extern, gemischt) der Portale finden (vgl. z. B. [Hinderer 2002], [Bullinger et al. 2002], [Rütschlin 2001]).

Zum heutigen Zeitpunkt ist festzustellen, dass die Unterscheidung der Portale nach decision-making support und collaborative processing nicht mehr zweckmäßig ist. So wie bereits einige der Autoren vorausgesehen haben (vgl. z. B. [White 1999], [Firestone 2003]), sind die Grenzen zwischen diesen Portaltypen mit der Zeit verschwommen, bis sie sich letztlich praktisch vollständig aufgelöst haben. Heutige Unternehmensportale vereinen in weiten Teilen alle von den verschiedenen Benennungen intendierten Funktionalitäten. Diese Konvergenz spiegelt sich auch in Abbildung 2-4 wider. Neben dem beabsichtigten Ziel, die Konvergenz von Portaltypen zu einem universellen Portal zu visualisieren, ist die einem Buch von Firestone entnommene Abbildung darüber hinaus ein gutes Beispiel für die in der Vergangenheit betriebene, bereits kritisch kommentierte, inflationäre Generierung eigener Benennungen für bereits bestehende Konzepte. Diese Entwicklung hatte wesentlichen Anteil daran, dass keine gemeinsame Terminologie und kein einheitliches Verständnis für den Bereich der Portale und die damit verbundenen Konzepte entwickelt werden konnten. Eine direkte Vergleichbarkeit verschiedener Konzepte war damit praktisch ausgeschlossen. Umso wichtiger ist es, die Fortentwicklung des Themenfeldes auch dazu zu benutzen, die Begriffsverwirrung zu beenden und ein einheitliches Begriffsverständnis zu etablieren.

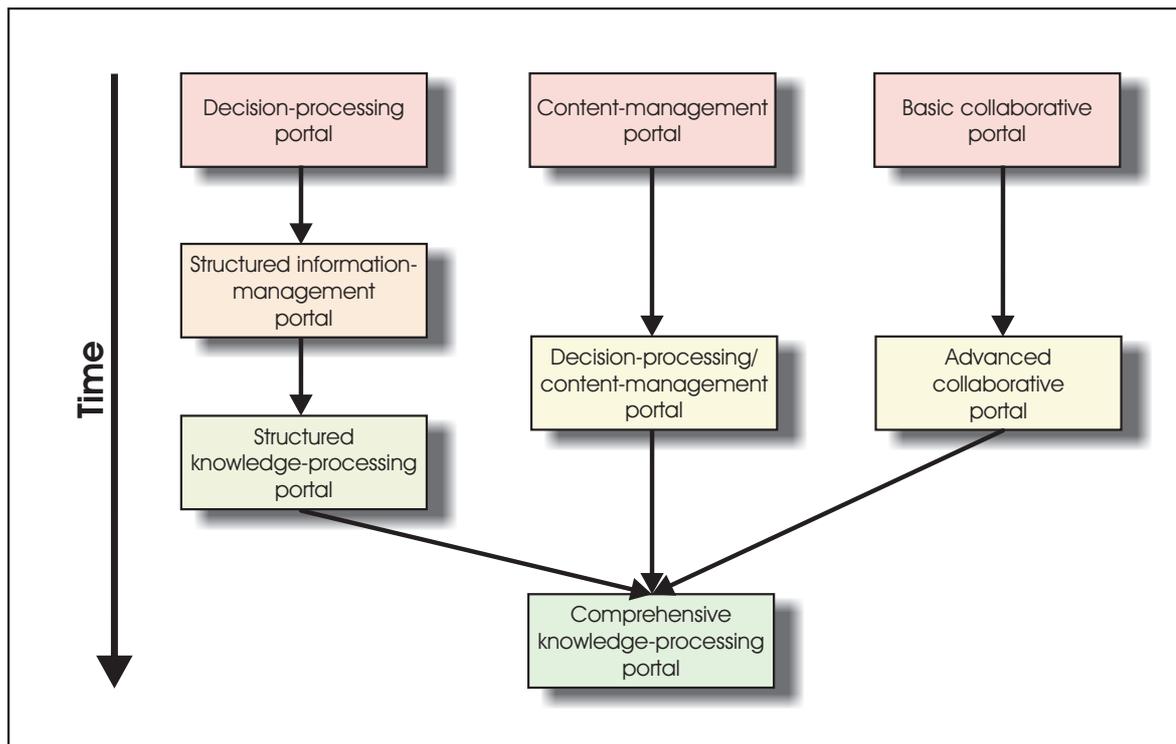


Abbildung 2-4: Portalevolution
(in Anlehnung an [Firestone 2003], S. 392)

Trotz der geäußerten deutlichen Kritik und der mittlerweile erreichten weitgehenden Konvergenz der Portaltypen ist die Kenntnis der Entwicklungsgeschichte der Klassifizierung auch heute noch relevant, da sie zur Erklärung der zum Teil zahlreich vorhandenen Portalinitiativen in Unternehmen beitragen kann.

Der Unterscheidung nach Zielgruppen kommt im besonderen Maße auch weiterhin eine große Bedeutung zu, da die vorwiegende Anzahl an Portalprojekten auf dieser Klassifizierung basierend durchgeführt wird. Das führt in Konsequenz dazu, dass häufig auf der einen Seite für jede Zielgruppe ein eigenständiges, nicht mit den anderen Portalen verbundenes Portal konzipiert, entwickelt und betrieben wird. Auf der anderen Seite wird dadurch aber zumindest im ersten Schritt die Komplexität jedes einzelnen Portalprojekts reduziert.

2.2.1.3 Funktionen und Eigenschaften

Die Funktionalitäten und Eigenschaften, die von einem Unternehmensportal unterstützt werden können, sind sehr vielfältig. Ausgewählte Basisfunktionalitäten und -eigenschaften werden im Folgenden daher nur überblicksartig genannt.⁶

Portalfunktionen und -eigenschaften:

- Die *Personalisierung* zielt darauf ab, den Benutzern für ihr Anforderungsprofil relevante Informationen und Anwendungen zur Verfügung zu stellen. Sie ist damit das zentrale Element, um der Informationsüberflutung entgegenzuwirken und gezielt Informationen zu kanalisieren und zu filtern.
- Die *Integration* vielfältiger Quellen in die Portaldarstellung (Front-End-Integration) umfasst Schnittstellen und Konnektoren zum Zugriff auf die Systeme eines Unternehmens zur Integration der Inhalte und Applikationen im Portal.
- Die *Prozessintegration* verbreitert den Fokus vom reinen Informationszugriff zu einer aktiven Arbeitsplattform.
- *Single Sign-On (SSO)* ermöglicht es dem Benutzer, nach ausschließlicher Anmeldung am Portal auf alle in das Portal integrierte Quellen und Applikationen zuzugreifen, ohne dass eine erneute Anmeldung an jedem einzelnen System notwendig wäre.
- *Flexible und leistungsfähige Suchfunktionen* ermöglichen den Zugriff auf die vom Portal zugänglich gemachten großen Informationsmengen unabhängig von der Ablagestruktur, die vom Portal vorgegeben wird. Essenziell ist, dass die Suche über alle Quellen, die in das Portal integriert sind, möglich ist.
- Die *synchrone und asynchrone Zusammenarbeit* erweitert das Potenzial hinsichtlich der Teamunterstützung in Richtung der Koordination und Kommunikation der Benutzer.
- Eine *einheitliche Benutzerschnittstelle* für alle in das Portal integrierten Quellen vereinfacht für den Benutzer die Navigation und verringert den Lernaufwand.
- Die *Geräteunabhängigkeit* macht das Portal für den Benutzer von jedem Endgerät (Laptop, PDA, Mobiltelefon etc.) zugänglich.

⁶ Für weitergehende Ausführungen kann beispielsweise auf folgende Quellen zurückgegriffen werden: [Dias 2001], [Collins 2001], [Delphi Group 2000], [Davydov 2001], [Bullinger et al. 2002], [Wege 2002], [Bauer 2001].

- *Sicherheitsfunktionen* (z. B. zur Steuerung der Zugriffsberechtigung) bieten orthogonal zu den anderen Funktionen die Grundlage für die Nutzung von Portalen in Unternehmen.

2.2.1.4 Begriffsdefinition

Bisher wurde keine eigene Begriffsdefinition vorgenommen, sondern nur mit dem unscharfen, nicht näher definierten Begriff des Unternehmensportals gearbeitet. Der beschreibende Diskurs der Entwicklungsgeschichte und Klassifizierungsansätze von Unternehmensportalen hat offenbart, dass es schwierig ist, eine einheitliche Begriffsverwendung zu identifizieren. Aus der Analyse und Synthese einiger im Folgenden exemplarisch ausgewählter Definitionen für Portale wird für diese Arbeit eine eigene Definition gebildet.⁷

- "The corporate portal is not a thing, but an application of a broad set of technologies following a very customized information design. These applications combine to provide a unique company window into all the streams of information that employees need, delivered via the familiar browser interface. There are many challenges in providing knowledge workers with a personalized, single-point-of-access desktop that integrates both existing corporate information systems and external information sources." ([Delphi Group 2000], S. 2)
- "Enterprise Portal (EP) is an enterprise-wide integration of business applications to the Web-specifically devised to avail the benefits of the Internet. The Integration of business applications extends the capability of a company to reach out to, and interact with its clients, partners, vendors, and employees. It offers a single point of entry, a single point of access, and a single point of information interchange." ([Hazra 2002], S. 623)
- „Portale bilden den Einstiegspunkt für einen gemeinsamen, personalisierten Zugang zu Web-basierten Anwendungen und Informationssystemen.“ [Flehmig 2001]
- „Mitarbeiter- und Unternehmensportale integrieren Anwendungen, Dienste und Inhalte aus vielen verschiedenen Informationsquellen auf einer Oberfläche am Arbeitsplatz eines Mitarbeiters, Kunden oder Lieferanten (*Single Point of Access*).“ ([Amberg/Holzner/Remus 2003], S. 3)
- „Der Begriff des Portals bezeichnet weniger die funktional geschlossene Konzeption einer konkreten Anwendungsform als vielmehr das Paradigma aggregierter und integrierter Präsentation.“ ([Kaspar/Burghardt/Schumann 2002], S. 178)
- "A portal is a single point of personalized interaction with applications, content processes and people, for the user." ([Confoy/Trabert 2003])

Zentraler Punkt aller Definitionen ist der „Single Point of Access“, den ein Portal für die Benutzer darstellt. Unterschiede ergeben sich vor allem darin, wer die Benutzer sind und für welche Systeme ein Portal der Single Point of Access ist. Besonders erwähnenswert ist der ausdrückliche Hinweis Hazras, dass ein Portal dazu in der Lage sein muss, gleichzeitig sowohl interne als auch externe Gruppen zu berücksichtigen.

⁷ Zu weiteren Definitionen vgl. z. B. [Dias 2001], [Plumtree 2000].

Diese Arbeit definiert ein Portal, wobei hierbei nur auf Unternehmens- und nicht auf Internetportale abgezielt wird, wie folgt:

Ein Portal ist der durch Front-End-Integration realisierte zentrale, personalisierte Einstiegspunkt zu allen Informationen, Applikationen, Prozessen und Personen eines Unternehmens. Durch die Personalisierung wird eine zielgerichtete Versorgung der Benutzer erreicht, die der allgemeinen Informationsüberflutung entgegenwirkt. Diese Aufgabe kann das Portal kontextabhängig gleichermaßen für die eigenen Mitarbeiter als auch für die externen Zulieferer, Kunden und Partner übernehmen.

2.2.1.5 Abgrenzung zu anderen Technologien

Portale sind als Integrationstechnologie in eine existierende gewachsene IT-Infrastruktur einzuordnen. Diese umfasst neben Portalen seit langem weitere Integrationstechnologien wie Electronic Data Interchange (EDI) und Application Integration (AI). An dieser Stelle findet daher eine Abgrenzung der verschiedenen Konzepte statt, die Aufschluss darüber gibt, worin die jeweiligen Aufgaben und Ziele und damit Gemeinsamkeiten und Unterschiede bestehen.

2.2.1.5.1 *Electronic Data Interchange*

Seit Mitte der 70er Jahre ist das Konzept des Electronic Data Interchange bekannt, das auf dem Austausch ausschließlich strukturierter und standardisierter Nachrichten zwischen Anwendungssystemen zweier Unternehmen beruht. Die umfangreiche Literatur zu EDI umfasst eine Vielzahl von Definitionen und Abgrenzungen. Diese unterscheiden sich im Wesentlichen in dem jeweils zugrunde liegenden Verständnis über Einsatzbereiche und Einsatzmöglichkeiten (vgl. zu einer Übersicht z. B. [Pfeiffer 1990]).

Brousseau ([Brousseau 1994]) sieht das wesentliche Ziel von EDI in der Automatisierung der Kommunikation und den mit dem Austausch von Informationen verbundenen Prozessen zwischen zwei Unternehmen. Ziel ist es, die Fehlerfreiheit des Informationsaustauschs sowie dessen Geschwindigkeit zu verbessern. Um einen hohen Automatisierungsgrad zu erreichen, können keine beliebigen, formatlosen, sondern ausschließlich von den Rechnern der Partner interpretierbare, standardisierte und strukturierte Nachrichten ausgetauscht werden.

Der Aufbau von EDI-Infrastrukturen verfolgt das Ziel, den standardisierten, operativen Geschäftsverkehr zwischen Kooperationspartnern zu automatisieren (vgl. [Neuburger 1994], S. 63). Hierzu sind die internen Systeme um Schnittstellen zu erweitern, die sowohl EDI-Nachrichten erzeugen und verschicken als auch empfangen und verarbeiten können. Aufgrund

der hohen Komplexität und großen Anzahl unterschiedlicher Nachrichten, die, obwohl standardisiert, teilweise branchenspezifisch sind, entstehen hohe Implementierungs- und Wechselkosten. Eine ökonomisch effiziente Anbindung ist wegen dieser hohen Implementierungs- und Wechselkosten daher nur bei längerfristigen, intensiven Beziehungen zwischen den EDI-Partnern möglich (vgl. [Rupprecht-Däullary 1994], S. 142).

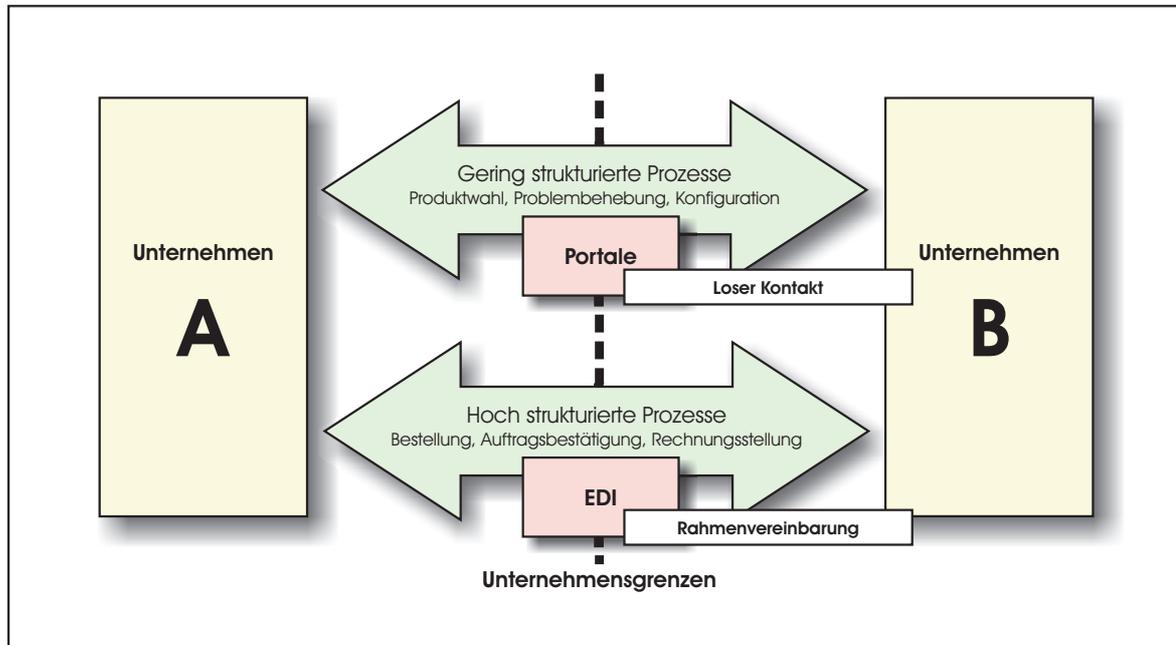


Abbildung 2-5: Einsatzszenarien für Portale vs. EDI
(vgl. [Gurzki 2002])

Als Möglichkeit der Unternehmenskooperation sieht Gurzki ([Gurzki 2002]) den Einsatz von EDI bei hoch strukturierten und standardisierten Prozessen und entsprechend bestehenden festen Rahmenvereinbarungen. Portale für den interorganisationalen Einsatz hingegen positioniert er im Bereich loserer, aber auch flexiblerer Kontakte bei geringer strukturierten Prozessen (vgl. Abbildung 2-5). Palmer vertritt eine ähnliche Position. Für ihn ist EDI bereits im Idealfall nur der Austausch von Informationen zwischen fest miteinander verbundenen Applikationen, wohingegen Portale die strategische elektronische Plattform für erfolgreiche Unternehmen geworden sind. EDI ist ausschließlich auf Transaktionen beschränkt, die in ein starres Korsett ausschließlich strukturierter Nachrichten gepresst werden müssen. Kollaborative Aspekte, wie sie Portale aufweisen, fehlen bei EDI sogar vollständig (vgl. [Palmer 2003]).

2.2.1.5.2 Applikationsintegration

Bis in die 80er Jahre war die IT-Landschaft geprägt von organisationszentrierten und monolithischen Mainframe-Anwendungen, die untereinander kaum in Verbindung standen. In der nächsten Entwicklungsstufe erlangte zunehmend ein Prozessfokus Bedeutung und Client/Server-Architekturen traten auf. EDI wurde zur Verbindung zwischen Unternehmen eingesetzt; zusätzlich waren erste lose Kopplungen zwischen den unternehmensinternen Anwendungen zu beobachten. Die bisher letzte Phase integriert verteilte Systeme auf Prozessebene zu eng gekoppelten Systemen und arbeitet in Echtzeit. Die Verbindung voneinander unabhängiger Anwendungen wird allgemein mit dem Begriff Applikationsintegration belegt (vgl. [Buhl/Christ/Pape 2001], S. 9 ff.).

Enterprise Application Integration

Ein in den letzten Jahren vielfach verwendeter Begriff ist Enterprise Application Integration (EAI). Die unter diesem Begriff diskutierten Systeme und Integrationskonzepte sind als Weiterentwicklung zuvor verfolgter Ansätze zu verstehen. Das Ziel von EAI bleibt gleichermaßen die Kopplung bisher voneinander unabhängiger, isoliert betriebener betrieblicher Applikationen (vgl. [Schelp/Winter 2002], S. 16). Im Gegensatz zu seinen Vorläufern, bei denen die Integration auf direkten Punkt-zu-Punkt-Verbindungen zwischen jeweils zwei miteinander verbundenen Systemen basierte und zu einer hohen Komplexität führte, verfolgt EAI den Ansatz der Schaffung einer separaten Integrationsschicht und wird damit als Middleware zur Integration von Applikationen positioniert.

Die zu integrierenden Applikationen werden nicht mehr direkt miteinander verbunden, sondern ausschließlich mit der Integrationsschicht. Diese übernimmt ihrerseits eine Mediatorfunktion bei der flexiblen Verknüpfung der Applikationen untereinander. Buhl, Christ und Pape ([Buhl/Christ/Pape 2001], S. 13) sehen in EAI die Grundlage für die Flexibilisierung der Geschäftsprozesse. Als entscheidenden Fortschritt von EAI gegenüber früheren Integrationsmethoden kennzeichnen Schelp und Winter (vgl. [Schelp/Winter 2002], S. 16) die allgemeine Komplexitätsreduktion des Gesamtsystems integrierter Unternehmensanwendungen, insbesondere aber die Reduktion der Schnittstellenkomplexität, die durch die Entkopplung der bisher direkt miteinander verbundenen Systeme möglich ist.⁸

⁸ Weitere Definitionen und detaillierte Ausführungen zu den Methoden und Konzepten von EAI finden sich z. B. in [Linthicum 1999], [Dangelmaier et al. 2002], [Erasala/Yen/Rajkumar 2003].

Schelp und Winter sehen Portale und EAI als komplementär zueinander und führen aus:

„Portallösungen benötigen eine vorgelagerte Integration der betrieblichen Applikationen, wenn sie Informationen aus zahlreichen Quellen integriert darstellen sollen und die Komplexität des Gesamtsystems beherrschbar sein soll.“ ([Schelp/Winter 2002], S. 18)

Puschmann bejaht ebenfalls die Komplementarität von Portalen und EAI, betont aber einen anderen Aspekt: Die EAI-Systeme „erweitern die Funktionalität von Portalen um die Möglichkeit der Durchführung von Transaktionen mittels System-zu-System-Integration (Maschine-Maschine).“ ([Puschmann 2003], S. 139)

Letson ([Letson 2001]) betrachtet in seinem Artikel ebenfalls die Beziehung zwischen Portalen und EAI und kommt zu dem Schluss, dass sich diese aufeinander zu bewegen und ergänzen. In einem von ihm zitierten Beispiel von Steve Dille wird ein Enterprise-Resource-Planning(ERP)-System mit anderen Systemen unter Zuhilfenahme von EAI verbunden. Beim Zugriff auf das ERP-System über ein Portal sind durch den Einsatz von EAI wichtige Informationen bereits synchronisiert; allerdings erlaubt die Nutzung von Portalen, weitere unstrukturierte Informationen in den Arbeitskontext des Anwenders einzubringen. Portale erlauben folglich die Kombination von sowohl strukturierten als auch unstrukturierten Informationen. EAI-Technologien sind hingegen nicht in der Lage unstrukturierte Informationen zu synchronisieren. In die gleiche Richtung gehen die Ausführungen des ebenfalls zitierten Marcel Van Hulle, die darauf abzielen, dass mit 85 % der größte Anteil an Informationen eines Unternehmens in unstrukturierter Form vorliegt. Für ihn sind es diese 85 % an unstrukturierten Informationen, die zur erfolgreichen Durchführung von E-Business benötigt werden, wobei diese nicht Bestandteil herkömmlicher automatisierter Applikationsintegration sind.

B2B Application Integration / Next Generation Application Integration

Mit zunehmender Kooperation zwischen Unternehmen ist EAI nicht mehr ausreichend, da diese ausschließlich auf innerbetriebliche Integration angewendet wird. Techniken, die als Fortführung von EAI über Unternehmensgrenzen hinweg eine Applikationsintegration verfolgen, werden z. B. unter den Bezeichnungen „B2B Application Integration“ (vgl. [Linthicum 2001]) oder „Next Generation Application Integration“ (vgl. [Linthicum 2004]) diskutiert. Linthicum führt die Notwendigkeit von B2B Application Integration auf den unaufhaltsamen Trend hin zu einer ereignisgesteuerten Wirtschaft zurück. In dieser führt das Auftreten von Nachfrage zu deren unmittelbarer Befriedigung. Dies ist seiner Ansicht nach nur umsetzbar, wenn sich die Möglichkeiten und die Reichweite der bestehenden, sowohl inter- als auch intraorganisationalen IT-Infrastruktur am Paradigma der Ereignisbezogenheit

orientiert. Entweder die Unternehmen automatisieren ihre grundlegenden Schnittstellen, um auf entsprechende Geschäftsereignisse unmittelbar reagieren zu können, oder sie sind gezwungen, den Markt zu verlassen. Eine Alternative sieht Linthicum nicht, für ihn ist das die Realität des E-Business (vgl. [Linthicum 2001], S. 3).

Des Weiteren kategorisiert er die zur Verfügung stehenden Ansätze zur Applikationsintegration in die vier Typen, die auch im Rahmen von EAI diskutiert werden (vgl. z. B. [Buhl/Christ/Pape 2001], [Linthicum 1999] und [Dangelmaier et al. 2002]):

- Data-Oriented AI (Integration auf Datenebene) bezeichnet den Datenaustausch zwischen Datenspeichern, in der Regel Datenbanken.
- Application Interface-Oriented AI (Integration auf API-Ebene) umfasst die Nutzung vorhandener Schnittstellen zum Zugriff auf vorhandene Anwendungen, ohne diese manuell ändern zu müssen.
- Method-Oriented AI (Integration auf Prozessebene) hat die gemeinsame Nutzung von Geschäftslogik zum Ziel. Funktionen werden nur einmal umgesetzt und von allen Anwendungen gemeinsam genutzt. Dies ist aber nur bei invasiven Eingriffen in die Programme oder bei der Neuimplementierung möglich.
- Portal-Oriented AI (Integration auf Anwenderebene) wird von den EAI-Vertretern als „primitive“ Methode betrachtet. Sie umfasst die Integration verschiedener Anwendungen durch zusammengefasste Präsentation der Benutzerschnittstelle in einer Oberfläche.

Linthicum (vgl. [Linthicum 2001], S. 91 ff., [Linthicum 2004], S. 99 ff.) analysiert die Vor- und Nachteile des Portal-Oriented-Application-Integration(POAI)-Ansatzes.

Als Vorteile führt er an:

- Es handelt sich um einen nicht invasiven Ansatz, bei dem es unnötig ist, die Back-End-Systeme innerhalb oder zwischen Unternehmen direkt miteinander zu verbinden. Dadurch können die hohen Kosten und Risiken, die damit verbunden sind, vermieden werden.
- Üblicherweise ist POAI wesentlich schneller zu implementieren als der Echtzeitdatenaustausch zwischen Back-End-Systemen.
- Die Portaltechnologie kann mittlerweile als ausgereift angesehen werden; es gibt zahlreiche abgeschlossene Projekte, die als Referenz bei der Durchführung eigener Portalprojekte dienen können.

Als Nachteile stellt er heraus:

- Informationsflüsse erfolgen nicht in Echtzeit und erfordern menschliche Interaktion.
- Informationen müssen durch eine weitere Ebene abstrahiert werden, was dazu führen kann, dass die Komplexität der Lösung insgesamt erhöht wird.

Linthicum zufolge präferieren die Benutzer in zahlreichen Anwendungsgebieten der Applikationsintegration weiterhin die Interaktion mit Back-End-Systemen durch eine Benutzerschnittstelle (POAI), anstatt die Informationen automatisch quasi hinter den Kulissen transferieren zu lassen. Er sieht aber den Trend einer Orientierung weg von Portalen hin zu einem Echtzeitaustausch von Informationen (vgl. [Linthicum 2004], S. 103). Für ihn stehen Portale – obwohl heutzutage in den Augen vieler eine Notwendigkeit – dem ultimativen Ziel der Applikationsintegration, dem Austausch von Unternehmensinformationen in Echtzeit, ohne Notwendigkeit des menschlichen Eingriffs, entgegen. Er formuliert das Ziel der Applikationsintegration als:

“to design the user out of the process, thus removing the greatest source of latency in the exchange of information, and to support the new event-driven economy.“ ([Linthicum 2004], S. 111)

Die einseitige Beschränkung auf B2B Integration mit dem Fokus auf hoch strukturierte Daten, die automatisch in Echtzeit zwischen Systemen ausgetauscht werden können, führt zu dieser unausgewogenen Einschätzung. Reynolds und Koulopoulos betonen eine völlig andere Sichtweise, indem sie die Schaffung eines zentralen Zugriffspunkts zur Unterstützung der zunehmenden wissenszentrierten Arbeitsabläufe der heutigen Arbeitswelt in den Vordergrund stellen (vgl. [Reynolds/Koulopoulos 1999]). In dieselbe Richtung gehen auch die Ausführungen in Abschnitt 2.1.2.

Zusammengefasst lässt sich festhalten, dass Portale einen benutzerzentrierten Fokus haben, bei dem sowohl strukturierte als auch unstrukturierte Daten und Informationen berücksichtigt werden. Im Gegensatz dazu verfolgt der Application-Integration-Ansatz als ultimatives Ziel eine System-zu-System-Integration ohne Benutzereingriff auf Basis hoch strukturierter Daten. Autoren, die beide Themen aus der Perspektive von Portalen betrachten, erkennen durchweg das Potenzial, das sich durch die gemeinsame Anwendung der beiden komplementären Technologien erzielen lässt. Stringente Verfechter des AI-Ansatzes verkennen oder verneinen häufig die Möglichkeiten, welche die Portaltechnologie eröffnet, und nehmen damit implizit in Kauf, dass durch Einseitigkeit suboptimale Lösungen entstehen oder Lösungen evtl. wegen hoher Kosten und Risiken überhaupt nicht realisiert werden.

2.2.2 Ausgewählte Aspekte von Portalumgebungen

Die allgemeine Einführung in die Grundlagen und Konzepte von Portalen konnte auf Grund des Umfangs des Themengebiets nur überblicksartig erfolgen. Im Folgenden werden drei ausgewählte, als Basis und Motivation für die vorliegende Arbeit dienende Aspekte von Portalumgebungen detaillierter vorgestellt.

2.2.2.1 Unterstützungsfunktion teambasierter Kooperation

In Übereinstimmung mit der in Abschnitt 2.1.2.2 dargestellten Vorteilhaftigkeit von Teamstrukturen für wissensintensive Unternehmen sehen Picot, Reichwald und Wigand eine wachsende Bedeutung teamartiger Kooperation und Koordination im Zuge allgemein zunehmender Komplexität der zu erfüllenden Aufgaben (vgl. [Picot/Reichwald/Wigand 2001], S. 281). Besondere Bedeutung kommt dabei der Nutzung und Bearbeitung einer gemeinsamen und einheitlichen Informationsbasis zu (vgl. [Borghoff/Schlichter 1998], S. 125). Diese erlaubt es, die gleichen Informationen allen Mitgliedern einer Gruppe zugänglich zu machen, um kooperative Entscheidungsprozesse zu unterstützen. Der gemeinsame Informationsraum erleichtert zudem die implizite Koordination der Mitglieder (vgl. [Busbach 1996], S. 97).

Portale verfügen über inhärente Funktionalitäten, welche die Arbeit von (räumlich und zeitlich verteilten) Teams unterstützen und ihnen eine gemeinsame Informationsbasis zur Verfügung zu stellen. Brophy und Venters formulieren das Ziel, das Portale verfolgen, als:

“need to efficiently support the work individuals perform. Such support should be tailored to the specific needs of the individual, and support the actions such individuals collectively perform.”
([Brophy/Venters 2001], S. 424)

Sie fassen Workplace-Portale als Portale mit kollaborativer Ausrichtung auf, deren konzeptionelles Modell auf der Interaktion zwischen Benutzern aufbaut und bei dem Portale eine mediative Rolle zwischen diesen wahrnehmen. Für Davydov ([Davydov 2001], S. 157) ist es erstmals mit der Einführung kollaborativer Mitarbeiterportale möglich geworden, voll funktionsfähige, kollaborative Systeme zu entwickeln, die explizit auf die Lösung der Probleme verteilter (oder virtueller) Teams ausgerichtet sind. Die Funktion von Portalen geht über die Unterstützung einzelner Individuen weit hinaus. Sie übernehmen die Aufgabe eines Koordinations- und Kommunikationsraumes, in dem bereichs- und unternehmensübergreifend Ideen und Know-how ausgetauscht, Experten gefunden sowie Produkte und Dienstleistungen erstellt werden (vgl. [Detlor 2000], S. 99).

Als zentrale Eigenschaft von Portalen zur Unterstützung von Teams sehen zahlreiche Autoren die Möglichkeit der Etablierung von gemeinsamen Arbeitsumgebungen, die auch als „Shared Information Workspace“ bezeichnet werden:

- “Collaborative portals enable teams of users to establish their own virtual project areas or communities and decide what they need to work together.” ([Murray 1999])
- “A common application of portals is the ability to create a shared workspace, often short-lived and self-managed, while incorporating resources and online information. [...] Most business benefits related to portals are derived from the ability to dynamically form teams without the restriction of geography, organization hierarchy, or even corporate boundaries.” ([Palmer 2003])
- “The shared workspace with any level of access control and workspace, as a retention facility, allows various levels of users to cooperate with each other seamlessly.” ([Jin Kim/Chaudhury/Rao 2002], S. 58)
- „Gruppenportale sind Systeme, die einen gemeinsamen Zugangspunkt für alle in einem Kooperationskontext relevanten Objekte bieten. [...] Gruppenportale verringern die Such- und Zugriffszeiten und fokussieren die Aufmerksamkeit des Nutzers auf die gruppenrelevanten Inhalte.“ ([Bullinger/Warnecke/Westkämper 2002], S. 768)
- „Typisch [für kollaborative Portale] sind gemeinsame Arbeitsbereiche, auf die Teammitglieder von verschiedenen Standorten zugreifen und in denen sie Dokumente ablegen und bearbeiten können.“ ([Föcker/Lienemann 2000], S. 19)

Detlor ([Detlor 2000]) identifiziert die drei Bereiche „Content Space“, „Communication Space“ und „Coordination Space“ als konstituierende Komponenten zur Schaffung eines Shared Information Workspaces. Die Abbildung 2-6 visualisiert deren Zusammenwirken.

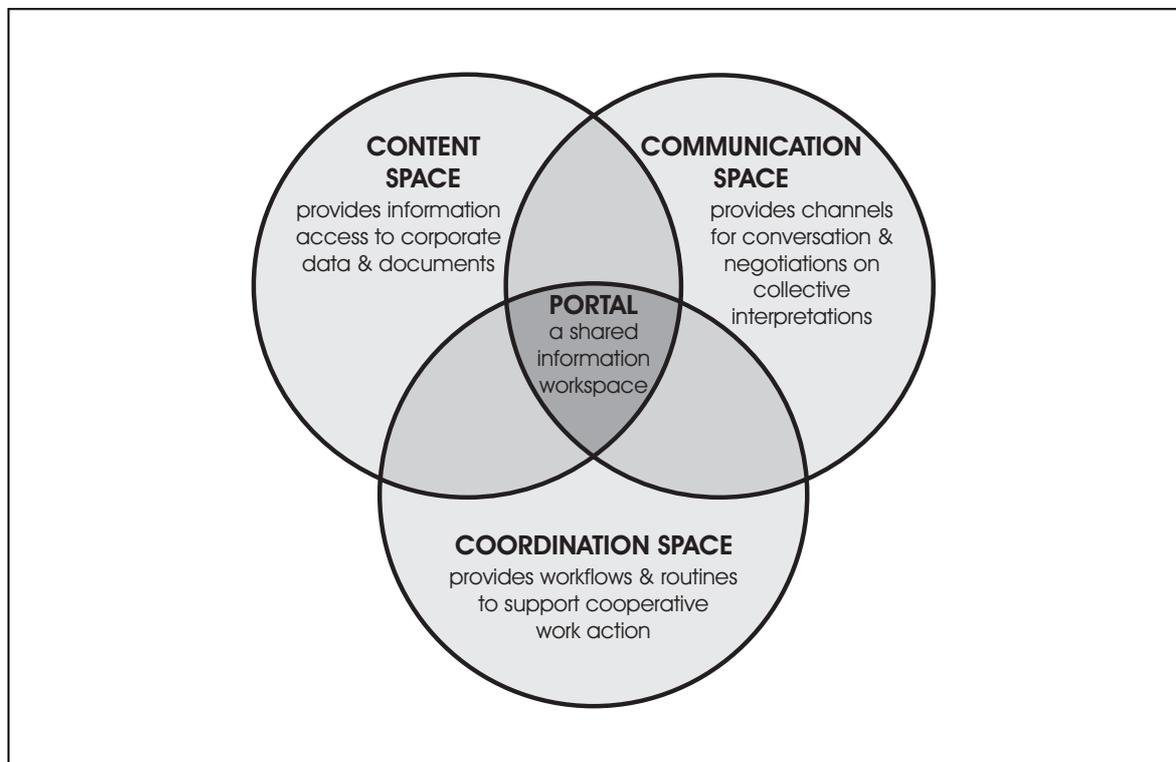


Abbildung 2-6: Portal als Shared Information Workspace
(vgl. [Detlor 2000])

Der Content Space umfasst die gemeinsam genutzten Informationen und bietet Funktionen zum Auffinden und Bearbeiten der Daten und Dokumente. Der Communication Space ermöglicht die Kommunikation zwischen den Teammitgliedern, um so ein gemeinsames Verständnis zu schaffen. Der Coordination Space unterstützt die koordinierte Zusammenarbeit im Team.

Mit ihrer gesamten Ausrichtung und ihren Funktionen sind Portale in besonderem Maße dazu geeignet, teamartige Kooperation sowohl bei zentralen als auch dezentralen und virtuellen Strukturen sowohl innerhalb als auch zwischen Unternehmen zu unterstützen.

2.2.2.2 Interorganisationale Nutzung von Portalen

Erhebungen zu Portalprojekten und -initiativen der vergangenen Jahre zeigen, dass Unternehmen vornehmlich Mitarbeiterportale realisiert haben (vgl. Rollenportale in Abschnitt 2.2.1.2). Diese bieten den internen Mitarbeitern eine integrative Sicht auf interne und extern zugängliche, ihre Arbeit betreffende Informationen. Der im Abschnitt 2.1.1.1 aufgezeigte Wandel zu einer verstärkten Kooperation bzw. zunehmenden engeren Vernetzung zwischen Unternehmen führt zu der Frage, welches Potenzial Portale in diesem Umfeld, z. B. gegenüber bisher üblichen Extranets, bieten.

Im Rahmen von Unternehmenskooperationen entsteht für die Teilnehmer die Notwendigkeit, Informationen, die für die Zusammenarbeit relevant und bereits im eigenen Intranet vorhanden sind, auch Partnern, Zulieferern und Kunden zugänglich zu machen. Dabei können die Empfänger die Informationen entweder explizit, z. B. per E-Mail erhalten (Push-Prinzip) oder aber diese aus einem Pool an Informationen, z. B. in den Datenbanken eines Extranets, selbst auswählen (Pull-Prinzip). Bei der Zusendung von E-Mails findet die erste Selektion beim Absender statt, nicht beim Empfänger. Der Absender wird alle aus seiner Sicht potenziell relevanten Informationen verschicken, obwohl der Empfänger nur einen gewissen Teil tatsächlich benötigt oder selbst als relevant einstufen würde. Bei der Auswahl von Informationen aus dem Extranet besteht das Problem darin, dass die aktuell relevanten aus der viel größeren Menge aller verfügbaren Informationen manuell extrahiert werden müssen. Beide Ansätze bewirken in ihrer einfachen Form für den Adressaten einen Informationsüberfluss, der die effiziente Nutzung verhindert und im Ergebnis häufig zu keinem Informationsgewinn, sondern zu einem Informationsverlust führen kann. Dies ist darin begründet, dass die eigentlich vorhandenen und sinnvoll nutzbaren Informationen jetzt in der Menge der insgesamt verfügbaren Informationen verloren gehen.

Die zwei Basisfunktionen eines Portals, Integration und Personalisierung, legen die Eignung von Portalen für diesen Problembereich nahe. Durch die Integration unterschiedlicher Quellen kann ein kontrollierter Zugang zu allen für die Zusammenarbeit wichtigen Informationsquellen erfolgen. Die Personalisierung sorgt für die benutzer- und partnerspezifische Auswahl und Anzeige der im aktuellen Kontext relevanten Informationen aus dem wesentlich größeren Pool an vorhandenen Informationen. Dadurch kann der Informationsüberflutung gezielt und effektiv entgegengewirkt werden. PiroNet NDH fasst dies folgendermaßen zusammen:

„Damit Partner und Unternehmen über das Extranet schnell kooperieren können, ist auch hier ein maßgeschneidertes Portal Voraussetzung. Nur aktuelle und individuelle Informationen sowie Services auf der Website binden die Partner und optimieren die Arbeitsprozesse. Ein Geschäftspartner empfindet es als mühsam, wenn er sich im Extranet erst durch Angebote oder Verträge kämpfen muss, die für ihn überhaupt nicht relevant sind.“ ([PiroNet NDH 2002])

Plumtree verwendet für diese Art von Portal den Begriff des „Extranet Corporate Portals“ und definiert diesen als „the gate to otherwise inaccessible information, maintained by an organization to which the portal users do not belong“ ([Plumtree 1999]). In anderen Veröffentlichungen findet sich die Bezeichnung Business-to-Business(B2B)-Portal, die sich an der Zielgruppe orientiert. Palmer (vgl. [Palmer 2003]) stellt den ökonomischen Nutzen des Einsatzes eines Portals heraus. Portale ermöglichen die Reduktion der Transaktionskosten, indem sie die Zusammenarbeit zwischen den Partnern fördern, den direkten Zugriff auf Geschäftsprozesse ermöglichen und ein bislang unbekanntes Maß an Verbundenheit schaffen, wodurch die ehemals klaren Grenzen zwischen den Unternehmen verwischen.

Portale lassen sich, unter Rückgriff auf die Ausführungen aus Abschnitt 2.1.1.3, als eine wichtige Form der Informationstechnologie identifizieren, die eine „Enabler“-Funktion für aktuelle und zukünftige Entwicklungen im Unternehmensumfeld hat. Durch die inhärenten Funktionen zur schnellen Integration neuer und Anpassung der Zusammenstellung bestehender Informationssysteme und Anwendungen bieten Portale ein hohes Maß an Flexibilität und können so sich stetig verändernden Anforderungen (vgl. Abschnitt 2.1) angepasst werden.

2.2.2.3 Das Multi-Portal-Problem

Neben den durch den Einsatz von Portalen erhofften Verbesserungen ergeben sich auch mögliche Risiken. Bei den Risiken kommt dem im Folgenden thematisierten Multi-Portal-Problem bereits heute, zukünftig aber mehr denn je eine entscheidende Rolle zu.

2.2.2.3.1 Situationsbeschreibung

Die zahlreichen Vorteile, die der Einsatz von Portalen verspricht, haben in den letzten Jahren zu dem Phänomen der Proliferation von Portalen in Unternehmen geführt. Die Proliferation bezieht sich zum einen direkt auf die Anzahl von Portalen, zum anderen aber auch auf die Anzahl von (inkompatiblen) Portal-Frameworks, auf denen die Portale basieren.

Gootzit und Phifer von der Gartner Group beschreiben den Zustand folgendermaßen: „Most [enterprises] find themselves in a situation in which they’ve deployed multiple portals within their architectures.“ ([Gootzit/Phifer 2003]). Jeffrey Man charakterisiert einen der Gründe hierfür als „if a portal is good, more portals must be better“ ([Mann 2002]). Verschiedene Autoren identifizieren weitere Gründe für diese Entwicklung:

- Anbieter von Standardsoftware verwenden für diese häufig ein eigenes Portal. Teilweise ist der vollständige Funktionsumfang nur bei Einsatz dieses Portals nutzbar. (vgl. [Phifer 2003], [Mann 2002])
- Portale sind häufig durch Initiativen von Abteilungen oder in dezentralen Geschäftseinheiten entstanden, zwischen denen keine Koordination erfolgte. (vgl. [Phifer 2003], [Gootzit/Phifer 2003], [Mello 2002])
- Portale wurden nach Zielgruppen (B2E, B2B, B2C; vgl. Abschnitt 2.2.1.2) aufgebaut. Bereits für jede Zielgruppe wurden teilweise mehrere Portale realisiert; insgesamt ist es durch diesen Ansatz üblich, mehrere Portale im Einsatz zu haben. (vgl. [Phifer 2003], [Davydov 2001])
- Unterschiedliche Technologie- und Herstellerpräferenzen führen zu unterschiedlichen Portalplattformen und Portalen. (vgl. [Phifer 2003])

Ein zusätzlicher Faktor ist, dass neben den mehreren internen auch noch zahlreiche externe Portale von Partnern im weitesten Sinne existieren, auf die von verschiedenen Anwendern zugegriffen werden muss.

Die alleinige Existenz mehrerer Portale in einem oder zwischen mehreren Unternehmen ist noch nicht unmittelbar als problematisch anzusehen, wären diese miteinander verbunden. Auf Grund fehlender Kompatibilität zwischen den Portalen verschiedener Hersteller, aber auch durch nicht vorhandene Kooperationsbeziehungen zwischen den Portalen, die auf Basis des Portal-Frameworks eines Herstellers entwickelt wurden, entstehen verschiedentlich als „stovepipe portals“ (vgl. [Gootzit/Phifer 2003]), „siload portals“ (vgl. [McInturff 2003]) oder „departmental silos“ (vgl. [Mello 2002]) bezeichnete Portale.

Als Folgen dieser Entwicklung werden identifiziert:

- Die gemeinsame Nutzung von Informationen wird durch die Proliferation voneinander unabhängiger Portale wesentlich erschwert. (vgl. [Narth 2003])
- Die Produktivität der Benutzer und der Wert der einzelnen Portale sinkt durch das erneute Entstehen von Barrieren, die einer einfachen gemeinsamen Nutzung von Informationen zwischen Mitarbeitern und Geschäftspartnern entgegenstehen. (vgl. [Mello 2002])
- Die unabhängig voneinander entstandenen, isolierten Portalinseln schaffen neue Herausforderungen, um eine umfassende Kommunikation und Interaktion der Benutzer zu ermöglichen. Das eigentliche Problem wurde nicht gelöst, sondern nur auf eine höhere Ebene verlagert. (vgl. [Gootzit/Phifer 2003])

Mann ([Mann 2002]) appelliert eindringlich an die IT-Verantwortlichen, sich auf die ursprünglichen Faktoren, die den Erfolg von Portalen begründet haben, zurückzubesinnen. Die zunehmende Desorientierung der Benutzer, die durch die unzähligen Quellen von Informationen und Anwendungen entstanden war, konnte durch die Einführung zuerst gemindert werden, droht sich ihm zufolge durch die Einführung oder Tolerierung zu vieler Portale aber erneut auf einer höheren Ebene fortzusetzen.

2.2.2.3.2 Existierende Lösungsvorschläge

Grundsätzlich sind sich alle Autoren einig, dass die kommende Herausforderung darin besteht, die verschiedenen existierenden Portalprojekte zu konsolidieren bzw. zu integrieren:

- “A significant challenge remains – integrating their internal portals and then extending that integration to portals outside of the enterprise.” ([Gootzit/Phifer 2003])
- “You want to take all of your portals and clean them and consolidate them so that you have one hot entry.” ([McInturff 2003])
- “Many organizations are facing the issues of too many portals, which were originally intended to be an integrating platform. [...] [They] will have to face the issue of portal consolidation – a natural consequence of portal proliferation.” ([Mann 2002])
- “Economic and competitive advantages will force rapid evolution from portal ‚mania‘ to portal consolidation, in which the corporate portal truly becomes a source of aggregation, community, and business efficiency.” ([Davydov 2001], S. 160)

Als zahlreiche Beiträge umspannender Ansatz zur Überwindung des Problems werden föderierte (engl. federated) Portale gesehen:

- “We’re heading toward federated portals, where you [have] multiple portal communities accessed by a single enterprise portal. You do not want to log on to multiple portals.” (Colin White in [Frye 2002])

- “Federation-supporting functionality will enable users to be at the center of their portal universes. This portal fabric will be an environment of interconnected portals focused on maximizing the user’s experience.” ([Gootzit/Phifer 2003])
- “How can a single portal strategy be established? The solution lies in creating federations of role portals that cover the spectrum of a company’s interests and needs.” ([Davydov 2001], S. 160)
- “A distributed architecture for an enterprise portal involves a network of connected, department-sized portal servers working together as a group of federated portals. [...] [One] advantage of the distributed architecture model is its ability to include portals from an organization’s partners and suppliers as part of the federation. In effect, the portal network supports information sharing, collaboration, and decision making throughout a company’s value chain, not just between and within its internal departments. This is in contrast to the emphasis of traditional supply chain and value chain automation, which is primarily focused on the transaction side of business [...]. [...] [Another] advantage [...] is that basic information and collaboration data is shared between portals even if each portal presents the information differently or implements a different user interface.” ([Eckel 1999])

In einer Studie untersuchen Phifer et al. ([Phifer et al. 2003]) verschiedene Portalthemen im Hinblick auf die jeweilige Intensität der Wahrnehmung (Visibility) in der Fachöffentlichkeit, deren Reifegrad (Maturity) und die Zeit, die diese bis zum Erreichen des Status einer Standardtechnologie für Portale benötigen werden (Time to Plateau). Sie unterscheiden in Bezug auf die Föderation von Portalen einerseits die *Föderierung von Portalen eines Herstellers* und andererseits die *Föderierung von Portalen verschiedener Hersteller* (vgl. Abbildung 2-7). Letztere befindet sich nach ihrer Ansicht noch ganz am Anfang der Entwicklung mit einem zeitlichen Horizont von 5 bis 10 Jahren, wohingegen die Föderierung von Portalen eines Herstellers bereits weiter fortgeschritten sei und in weniger als 2 Jahren etabliert sein könne.

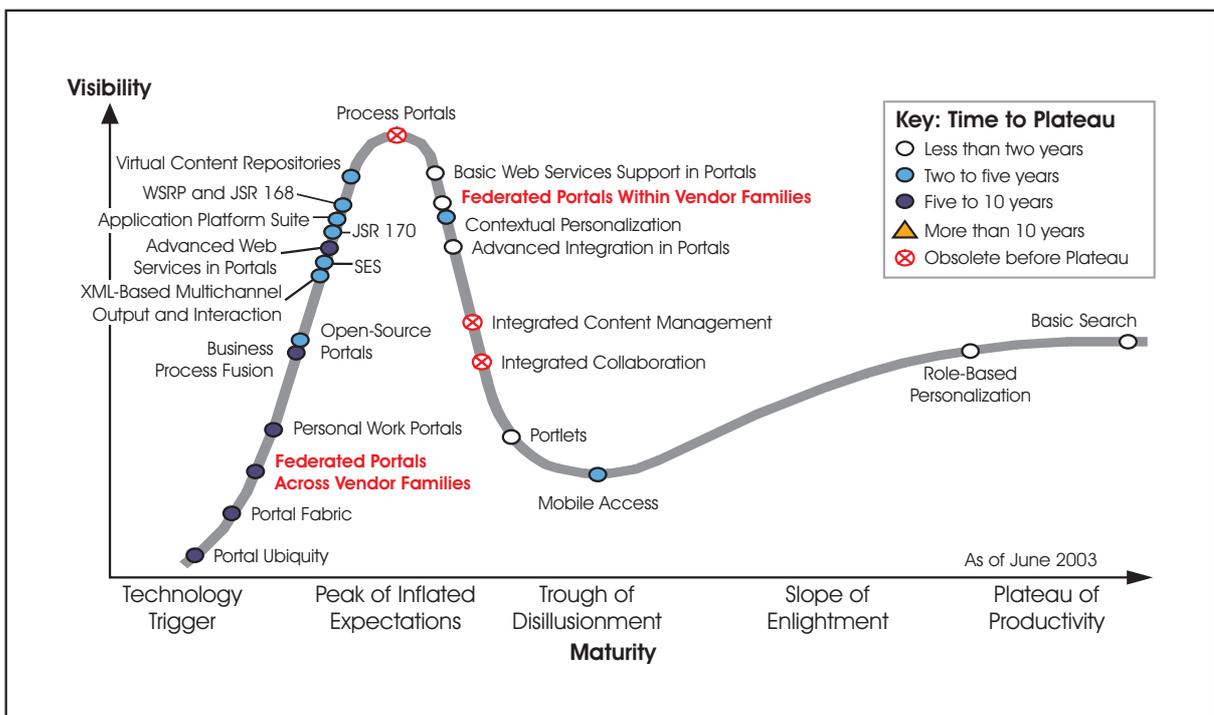


Abbildung 2-7: Hype Cycle for Portal Ecosystems, 2003
([Phifer et al. 2003])

Phifer ([Phifer 2003]) identifiziert fünf Kernelemente, welche die Interoperabilität bei einer Multi-Portal-Installation maßgeblich beeinflussen:

- Benutzerverzeichnisse zur Verwaltung der Benutzer von Portalen und in diese integrierte Anwendungssysteme.
- Sicherheit zum Zugriff auf Portale und auf in diese integrierte Anwendungssysteme.
- Personalisierungsinformationen, zur personalisierten Anzeige von Informationen und Anwendungen im Portal.
- Meta-Daten zur standardisierten Beschreibung von Portalinhalten.
- Portlets⁹, als Informationskomponenten eines Portals.

Die Konsolidierung von Benutzerverzeichnissen und Bemühungen zur Schaffung von Single-Sign-On-Lösungen für verschiedene Anwendungssysteme, die bereits in der Zeit vor Aufkommen der Portale relevant waren, erleichtern zumindest die innerbetriebliche Föderation. Im Bereich der Sicherheit, der Personalisierungsinformationen und der Meta-Daten, um nur einige Beispiele zu nennen, fehlen Standards bzw. vorhandene werden von den Herstellern nicht genutzt (z. B. Extensible Access Control Markup Language vgl. [OASIS 2003] und Dublin Core vgl. [DCMI 2004]). Einzig im Bereich der Portlets sind erste Standards verfügbar.

Dies veranlasste Phifer ([Phifer 2003]), das Konzept des so genannten „Überportals“, das auch an anderer Stelle (vgl. z. B. [Roberts-Witt 2000]) diskutiert wird, als Zwischenstufe bis zur Verfügbarkeit erster Standards für föderierte Portale vorzuschlagen. Die Verfügbarkeit erster Standards erwartet er mit 70 % Wahrscheinlichkeit für das Jahr 2005. Er definiert dieses Konzept als:

“German for ‘over’, uber implies a high level, overarching portal. The uberportal is a high-level, horizontal portal framework on top of one or more horizontal or vertical portals. The uberportal is essentially the entry point, or home portal, in a multiportal deployment. [...] One doesn’t buy an uberportal, one builds it.” ([Phifer 2003])

Grundsätzlich sollte ein Überportal alle fünf der oben genannten Aspekte berücksichtigen. Da dies aus technischen und/oder Kostengründen meist nicht möglich ist, schlägt Phifer das „poor man’s uberportal“ vor, das folgende Eigenschaften aufweisen soll:

- Konsistente Benutzerverzeichnisstruktur (einheitlich oder repliziert).
- Single Sign-On mit allen beteiligten Portalen.

⁹ Zu einer Definition des Begriffs Portlet vgl. Abschnitt 4.2.2.

- Anmeldung führt den Benutzer immer zum Überportal.
- Zugriff auf tiefer liegende (Sub-)Portale über Verknüpfungen oder Seiten aus dem Überportal.
- Bei der Auswahl einer Verknüpfung auf ein Subportal wird dort die personalisierte Startseite des Benutzers, in der Regel in einem neuen Fenster, geöffnet.
- Bei der Beendigung der Aufgaben im Subportal hat der Benutzer das Fenster zu schließen und kehrt damit in das Überportal zurück.

Zusätzlich schlägt Phifer drei Modelle (front-door, side-door und back-door) vor, wie die Informationen zwischen dem Überportal und den Subportalen verteilt werden können, und orientiert sich dabei an der Nutzungshäufigkeit einzelner Informationen durch die Benutzer.

Diese Aussagen bleiben alle entweder auf der Ebene rein abstrakter Formulierung allgemeiner Ziele, nämlich der Föderierung von Portalen, oder bieten funktionell so weit eingeschränkte Konzepte, dass ihr Wert zumindest mittel- bis langfristiger mehr als fraglich ist.

2.2.3 Forschungs- und Praxislücke

Unternehmensportale haben nach einer etwa sechsjährigen rasanten Entwicklung eine Konsolidierung und Vereinheitlichung der Basiskonzepte und -funktionalitäten erfahren. Mit der technologischen Stabilisierung – bei allerdings weiterhin fehlenden Standards und daraus folgenden herstellerabhängigen Implementierungen – positionieren sie sich als unverzichtbares Instrument zur personalisierten Informationsversorgung. Damit wenden sie sich gegen die allgemeine Informationsüberflutung, unterstützen gleichzeitig teambasierte Strukturen und dienen als Single Point of Access für interne und externe Benutzer.

Ihr zunehmender Einsatz wirkt zahlreichen durch zurückliegende und aktuelle Entwicklungen im Unternehmensumfeld (vgl. Abschnitt 2.1) aufgeworfenen Problemen entgegen. Umfangreiche Studien (vgl. z. B. [Plumtree 2002]) sagen voraus, dass bis 2006 bis zu 90 % der Unternehmen ein oder mehrere Portale im Einsatz haben werden und dass diesen ein wesentlicher Einfluss auf den Geschäftserfolg und die Wettbewerbsfähigkeit zukommen wird.

Die durch den Erfolg und die damit einhergehende Verbreitung entstehende Proliferation von Portalen schafft aber auch neue Probleme. Der als wesentliche Eigenschaft herausgestellte Single Point of Access geht hierdurch wieder verloren. Die Unternehmen stehen erneut vor der Aufgabe der Integration verschiedener Informationsquellen und Anwendungen, die

eigentlich mit der Einführung von Portalen gelöst werden sollte, jedoch auf einer höheren Ebene.

Die Proliferation von Portalen wird in der wissenschaftlichen Literatur bisher nur vereinzelt erwähnt. In diesen Fällen kommt die Betrachtung jedoch über das ausschließliche Erkennen des Problems und einiger mit ihm verbundener Fragestellungen nicht hinaus. Das Fehlen von Beiträgen aus Theorie wie Praxis, die mit detaillierten Analysen, entsprechenden Konzeptionalisierungen oder daraus abgeleiteten Modellen oder gar Umsetzungen zu einem Erkenntnisgewinn in diesem Forschungsfeld beitragen würden, ist als Forschungs- und gleichfalls als Praxislücke zu identifizieren. Die Situation ist umso erstaunlicher, als bereits kurze Zeit nachdem das Konzept der Unternehmensportale in die Diskussion eingebracht wurde, einige Autoren auf das potenzielle Problem aufmerksam gemacht haben. Ein Ansteigen der Erwähnungen legt nahe, dass dem Thema inzwischen stärkere Bedeutung beigemessen wird. Substanzielle Fortschritte bei der Behandlung sind bisher aber nicht zu erkennen.

2.3 *Forschungsziel*

2.3.1 Zieldefinition

Die erste Definition des Ziels der vorliegenden Arbeit wurde bereits in Abschnitt 1.1 gegeben und im Abschnitt 1.2 durch die Darstellung der wissenschaftstheoretischen Ausrichtung ergänzt. Die vorbereitenden Darstellungen in den vorherigen Abschnitten, die schlussendlich zur Offenlegung der die Arbeit motivierenden Forschungs- und Praxislücke führten, ermöglichen nun eine – unter Beibehaltung der wissenschaftstheoretischen Ansätze – Konkretisierung und eine bessere Nachvollziehbarkeit. Hieraus ergibt sich die folgende Zieldefinition der Arbeit:

Basierend auf der Erkenntnis, dass zur nachhaltigen Sicherung des Potenzials der Nutzung von Portalen, bei deren gleichzeitiger zunehmender intra- wie interorganisationalen Proliferation, die Föderierung von Portalen notwendig ist, hat im Rahmen der Arbeit zuerst eine Operationalisierung zu erfolgen, was unter dem Begriff föderierter Portale zu verstehen ist. Aufbauend auf der Analyse der Anforderungen, die an föderierte Portale in verschiedenen Szenarien zu stellen sind, ist die Konzeption eines Modells zur Föderierung von Portalen durchzuführen. Die Tauglichkeit des abstrakten Konzepts ist durch eine informationstechnische Realisierung und durch den praktischen Einsatz in Fallstudien oder Anwendungsszenarien zu validieren.

Die *Operationalisierung* des Begriffs föderierter Portale hat vor dem Hintergrund zu erfolgen, dass nur dann eine Konzeption erstellt werden kann, wenn der zu konzeptionalisierende Gegenstand klar definiert ist. Wie die Ausführungen in Abschnitt 2.2.2.3.2 gezeigt haben, liegt eine solche klare Definition nicht vor. Dies ist vor allem auf den ausschließlich narrativ beschreibenden Charakter, die mangelnde Substanz und Präzision, mit der die Diskussion bis heute geführt wurde, zurückzuführen.

Übergeordnetes Ziel bei der *Konzeption* eines Modells zur Föderierung von Portalen ist es, Vorschläge für zukünftige Standards bzw. Standardisierungsbemühungen in diesem Bereich zu machen. Die Konzeption selbst hat dabei so offen wie möglich zu erfolgen. Das umfasst auf der einen Seite aus betriebswirtschaftlicher Sicht die Möglichkeit zur Abbildung sehr unterschiedlicher Formen der inner- und zwischenbetrieblichen Kooperation mit den unterschiedlichen Anforderungen an die Flexibilität und Spezifität einer Föderation von Portalen (vgl. Abschnitt 2.1.1.1). In der Rolle eines gleichfalls Kern- und doch weitgehend transparenten Unterstützungssystems der Kooperation muss dieses minimalinvasiv sein. Die Föderation von Portalen sollte sich an die Gegebenheiten der Kooperation anpassen und nicht umgekehrt. Auf der anderen Seite hat sich die Konzeption aus technischer Sicht an vorhandenen technologischen Standards und am existierenden Portalumfeld zu orientieren. Nur unter Berücksichtigung dieser technischen Aspekte und des übergeordneten Ziels ist ein Architekturentwurf zu gewährleisten, der sich zur Realisierung der Föderierung von Portalen unterschiedlicher Anbieter eignet.

Das zu entwickelnde Konzept stellt nur einen abstrakten Rahmen dar, der über die Tauglichkeit und Umsetzbarkeit des Konzepts wenig aussagt. Daher hat im Anschluss eine informationstechnische *Realisierung* zu erfolgen. Diese basiert auf der Erweiterung eines existierenden Portal-Frameworks um die notwendigen Dienste und Schnittstellen zum Aufbau, zur Verwaltung und zur Durchführung der Föderation. Neben der rein funktionalen Validierung des Konzepts ist auch eine anwendungsorientierte Validierung in Form von Fallstudien oder Anwendungsszenarien anzustreben. Diese können Aufschlüsse darüber geben, inwieweit das erarbeitete Konzept föderierter Portale und seine Umsetzung dazu in der Lage ist, die IT-unterstützte Zusammenarbeit der inner- und zwischenbetrieblichen Kooperation zu verbessern. Die Erfahrungen und Beobachtungen können darüber hinaus genutzt werden, um das Konzept und die Realisierung zu einem späteren Zeitpunkt in iterativen Schritten zu überarbeiten und zu verbessern.

2.3.2 Andere Forschungsansätze

In Abschnitt 2.2.3 wurde herausgestellt, dass im Bereich föderierter Portale kaum nennenswerte wissenschaftliche Ansätze existieren. Dies wird im Folgenden durch die Analyse einer Veröffentlichung (Abschnitt 2.3.2.1) untermauert. Aufgrund des Fehlens substanzieller Quellen im Bereich der Föderierung von Portalen werden drei Ansätze vorgestellt, die sich im weitesten Sinne mit verwandten Themengebieten und Eigenschaften befassen. Dargelegt werden jeweils ein Überblick über das Ziel, mögliche Berührungs- und Unterscheidungspunkte zwischen dem Ansatz und dem in dieser Arbeit verfolgten Forschungsziel sowie ggf. Quellenangaben zur weiteren Vertiefung.

2.3.2.1 Puschmann und Alt: „Process Portals inter-organizational networking of businesses“

Puschmann und Alt ([Puschmann/Alt 2004]) thematisieren in ihrem Beitrag ebenfalls die Problematik, die mit der Proliferation von Portalen verbunden ist. Sie verwenden den Begriff des Prozessportals, um die exponierte Stellung der Integration von und die Ausrichtung an Prozessen in Portalen hervorzuheben.

Sie schlagen eine erweiterte Portalarchitektur vor, die produktneutral auch interorganisationale Aspekte berücksichtigt und gleichzeitig der portalimmanenten Integration zwischen Benutzerpräsentation, Applikationslogik und Datenhaltung Rechnung trägt. Als Technologien berücksichtigen sie für die verschiedenen Bereiche Portlets, Web Services und Enterprise Application Integration. Abschließend findet die Vorstellung einer Fallstudie statt, die zusammen mit einem großen Automobilhersteller durchgeführt wurde. Die Autoren beschreiben die Erfahrungen, die bei der Umsetzung und Konsolidierung der Portalinitiativen mithilfe der erweiterten Portalarchitektur gemacht wurden. Sie positionieren die Fallstudie derart, dass sie gleichzeitig zur Validierung der von ihnen vorgeschlagenen Architektur dient. Das Hauptaugenmerk in dem Beitrag liegt vornehmlich in der Portlet-basierten Integration von Anwendungen und Prozessen in ein oder mehrere Portale jedoch nicht in der tatsächlichen Integration und Verknüpfung von Portalen untereinander.

Die vorgestellte Architektur für den Einsatz im interorganisationalen Umfeld bleibt auf einer derart abstrakten Meta-Ebene stehen, dass eine substanzielle Bewertung schwer fällt. Es entsteht der Eindruck einer unverbindlichen Sammlung von Konzepten, die durchaus dazu geeignet ist, im genannten Anwendungsbereich eingesetzt zu werden, aufgrund ihrer größtmöglichen Allgemeinheit jedoch wenig Orientierung bietet. Das Beispiel eines Automobil-

herstellers ist wenig repräsentativ und aussagekräftig. Die zuvor dargestellte Architektur ist so unspezifisch, dass sie nahezu alle Möglichkeiten der Realisierung offen lässt. Obwohl sie beim Entwurf besonders interorganisationale Aspekte berücksichtigen soll, werden im Beispiel keinesfalls eigenständige Unternehmen betrachtet. Stattdessen wird in der Zusammenfassung explizit als Erfolgsfaktor die Koordination und Kommunikation der Portalarchitektur durch eine zentrale Einheit wie z. B. die IT-Abteilung genannt. Dies entspricht bereits hinsichtlich intraorganisationaler Rahmenbedingungen einem Sonderfall.

Die Fragestellung, welche die Autoren zu behandeln anstreben, ist von unmittelbarer Bedeutung für die vorliegende Arbeit. Im Verlaufe der Ausführungen verlieren sie diese aber weitgehend aus den Augen, um schlussendlich ein mehr oder weniger klassisches Portalprojekt zu beschreiben, das nur entfernt etwas mit der eingangs dargestellten Zielsetzung zu tun hat, nämlich die Folgen der Proliferation von Portalen zu überwinden.

2.3.2.2 ANSA: „Establishing Co-operation in Federated Systems“

Das bereits 1985 in England gestartete ANSA (Advanced Networked Systems Architecture) Projekt beschreibt als Ziel der von ihr entwickelten Architektur die Möglichkeit, verteilte Systeme entwickeln zu können, die sowohl allein stehend als auch als ein Ganzes (föderiert) arbeiten können. Dabei soll die Verteiltheit für den Entwickler vollständig transparent sein (vgl. [ANSA 1991], S. 1).

Die vier Hauptaufgaben, die im Rahmen des Projekts bearbeitet werden sollen, stellen Beasley et al. ([Beasley et al. 1994], S. 7) folgendermaßen dar:

- Heterogene Interaktion: Unterstützung der technologischen, namens- und objektbezogenen sowie semantischen Beschreibungsaspekte bei der Interoperabilität föderierter Systeme.
- Föderiertes Objektmanagement: Unterstützung des Gesamtmanagements von Objekten und Ressourcen in föderierten Systemen.
- Föderierte Entwicklung: Unterstützung beim Design und der Entwicklung interoperabler Clients und Server durch unterschiedliche Organisationen.
- Föderierte Unternehmensverhandlungen: Unterstützung bei allen institutionellen und administrativen Aspekten zur Etablierung der Interoperabilität föderierter Systeme.

Im Unterschied zu den Zielsetzungen der vorliegenden Arbeit liegt der Fokus beim ANSA-Projekt auf der Entwicklung eines allgemeinen verteilten bzw. föderierten Objektsystems und den damit einhergehenden Fragestellungen. Entwickeln von Applikationen soll ein offenes

und standardisiertes Framework zur Verfügung gestellt werden, mit dem sie neue Anwendungen entwickeln bzw. bestehende Anwendungen erweitern können, die dann zur Unterstützung von Netzen kooperierender Suborganisationen eingesetzt werden können.¹⁰

2.3.2.3 FOSTER: “Information and Communication Technology support infrastructures and interoperability for Virtual Organisations”

Im Rahmen des FOSTER-Projekts, eines europaweiten, von der Europäischen Union finanzierten und aus zahlreichen Unterprojekten bestehenden Forschungsprojekts und seiner Vorläufer, wird das Ziel verfolgt, eine Architektur zu definieren, die Dienste und Interoperabilitätsregeln für Informations- und Kommunikationstechnologien (engl. information and communication technology, ICT) zur Bildung und zum Einsatz in virtuellen Organisationen (VO) festlegt (vgl. [Camarinha-Matos/Menzel/Cardoso 2003], S. 4 ff.).

Ein Zwischenbericht stellt einen Überblick über erste Ergebnisse der Beschäftigung mit dem Themenkomplex vor. Abbildung 2-8 zeigt einerseits die dem Gesamtprojekt zugrunde liegende Konzeptionalisierung verschiedener Klassen von IuK-Technologien, die in dessen weiterem Verlauf untersucht werden sollen. Andererseits werden diese Technologien jeweils verschiedenen Klassen von VO zugeordnet.

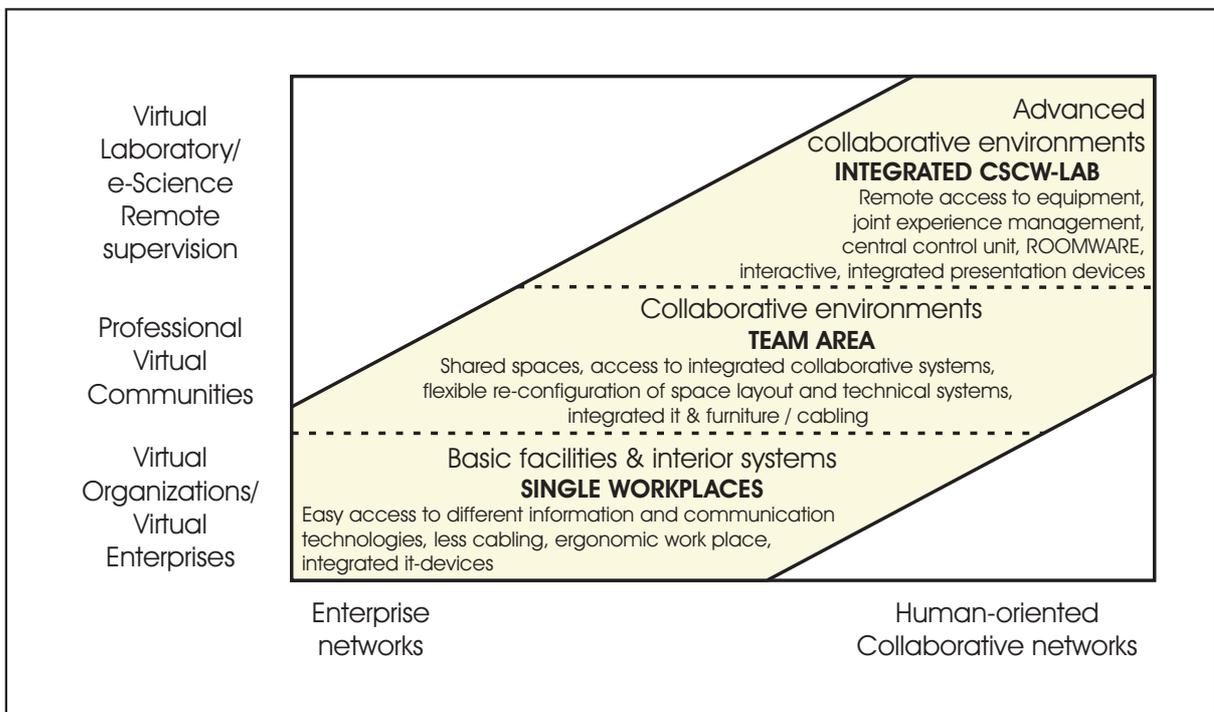


Abbildung 2-8: Klassen von VO und zugehörige IuK-Technologien (vgl. [Camarinha-Matos/Menzel/Cardoso 2003], S. 24)

¹⁰ Weiterführende Informationen finden sich auf der Internet-Präsenz der Organisation, <http://www.ansa.co.uk>.

Der Zwischenbericht nennt hierzu die von den Unterprojekten bisher zur Implementierung von IuK-Systemen zur Unterstützung von VO herangezogenen Paradigmen:

- Multi-Agent-Systeme
- Knowledge Management
- Portale
- Objektorientierung
- Relationale/föderierte DB
- Serviceorientierung

Das speziell im Kontext der vorliegenden Arbeit relevante Portalparadigma wird folgendermaßen thematisiert:

“A ‘popular’ term – portal – could be considered a special case of service federation infrastructure. This term is often used in the literature to represent rather different structures, from (i) a simple web entry point to a number of resources without any other form of collaboration among the resource providers, to (ii) some form of virtual organization. In this latter case, the basic idea is the existence of a ‘centralized access point’, where the various VO members are ‘represented’ and that might offer a ‘joint representation’ to the outside, as well as a platform for interactions among the VO members.” ([Camarinha-Matos/Menzel/Cardoso 2003], S. 12)

Die Nützlichkeit des Portalparadigmas als geschlossene Konzeption wird jedoch im Verlauf der weiteren Ausführungen des Berichts in Zweifel gezogen und schlussendlich nicht weiter verfolgt. Der Fokus verbleibt bei einzelnen Systemen und Tools, die zur Unterstützung virtueller Organisationen genutzt werden können; eine übergeordnete integrative Ebene, wie sie ein Portal darstellen könnte, ist nicht zu erkennen.

2.3.2.4 Zhang et al.: „Enterprise Virtualisation: Concept, Methodology, and Implementation“

Der Beitrag von Zhang et al. ([Zhang et al. 2001]) stellt wichtige Konzepte, die auch bei Portalen zur Anwendung kommen, dar, ohne jedoch Portale selbst zu benennen. Er trägt aber zu einem Grundverständnis der technologischen Anforderungen und Ziele von Portalen bei und bietet damit aus einem anderen Betrachtungswinkel Ansätze für Möglichkeiten zur Förderung von Unternehmenssystemen.

Kerngedanke ist die Virtualisierung des Unternehmens. Zur Erklärung des Begriffs werden die Konzepte des virtuellen Raums (engl. virtual space, VS) und der Virtualisierung eingeführt. Unter einem virtuellen Raum verstehen die Autoren

“an unbounded information process field which is built on computer networks“ und „in the VS, various information and data can be exchanged, processed, converted, stored, and displayed in security.“ ([Zhang et al. 2001], S. 218)

Virtualisierung definieren sie als

“mapping process through which the inner structure and implementation of an entity in real space are hidden, and only its outer functional features are mapped to the VS so that this entity becomes visible, operable, manageable, and portable in the VS.” ([Zhang et al. 2001], S. 218)

Resultierend daraus leiten sie mithilfe dieser Begriffe ihre Definition der Virtualisierung des Unternehmens ab:

“Enterprise virtualisation is a set of mapping processes, called virtualisation, which map key elements of an enterprise from limited real space to unbounded virtual space. The purpose of enterprise virtualisation is to make the enterprise visible, operable, manageable, and re-configurable.” ([Zhang et al. 2001], S. 218)

Damit beschreiben sie in abstrakter Form gleichermaßen die Prinzipien, die bei der Anwendung von Portalen auf Unternehmenssysteme notwendig sind. Nur wenn die Ressourcen und Prozesse computerbasiert zugreifbar gemacht werden, kann eine Öffnung des Unternehmens hin zum virtuellen Raum erfolgen. Andernfalls bleibt es dort unsichtbar. Das Ziel besteht darin, auf weite Teile der Unternehmenssysteme und damit des Unternehmens aus dem virtuellen Raum heraus zuzugreifen und diese steuern zu können. Aufgrund der andauernden Dynamik in den Unternehmensprozessen und der -umwelt ist es notwendig, die virtualisierten Ressourcen und Prozesse sehr flexibel in immer neuer Form rekonfigurieren zu können. Hieraus entstehen beispielsweise virtuelle Arbeitsbereiche für Projekte und Teams kooperierender Mitarbeiter, die ebenso schnell eingerichtet wie verändert und wieder aufgelöst sowie so sehr schnell den Unternehmensanforderungen angepasst werden können.

Das beschriebene Konzept und die Implementierung sind durch die produktionswirtschaftliche und Computer Integrated Manufacturing(CIM)-Perspektive der Autoren motiviert. Sie sind als insgesamt proprietär zu bewerten und im Wesentlichen auf der Ebene von Objektmodellen einzelner (Produktions-)Ressourcen angesiedelt.

3 Föderationen

Nachdem das Forschungsziel der vorliegenden Arbeit, die Konzeption und Umsetzung eines Modells zur Föderierung von Portalen, definiert wurde, wird im Folgenden dargelegt, welchen Ursprung und welche Bedeutung der Begriff Föderation hat. Des Weiteren wird betrachtet, in welchen Gebieten, insbesondere in der Informationstechnologie, Föderationen ebenfalls zur Anwendung kommen und welche Problemfelder und Konzepte sich in diesen Forschungsfeldern durch Föderationen ergeben.

Die konkrete Begriffsverwendung im Kontext von Portalen und die für diesen Anwendungsbereich relevanten Problemfelder und Konzepte werden in Kapitel 4 unter Rückgriff auf dieses Kapitel dargelegt.

3.1 Ursprung des Begriffs Föderation

Der Ursprung des Begriffs Föderation liegt im politischen Bereich, was sich in den Begriffsdefinitionen verschiedener Lexika widerspiegelt:

- *Föderierte, der; -n, -n verbündeter Staat, Verbündeter föderieren; sich verbinden, zusammenschließen*
([Langenscheidt 2003])
- *Föderalismus; (lat.) Allg.: F. ist ein Ordnungsprinzip, das auf weitgehender Unabhängigkeit einzelner Einheiten beruht, die zusammen aber ein Ganzes bilden (z. B. mehrere Länder, Provinzen einen Staat; mehrere Vereine einen Verband etc.). Pol.: F. stellt eine politische Ordnung dar, bei der die staatlichen Aufgaben zwischen Gesamtstaat und Einzelstaaten aufgeteilt werden, und zwar so, dass beide politischen Ebenen für bestimmte (verfassungsgemäß festgelegte) Aufgaben selbst zuständig sind.*
([Schubert/Klein 1997], S. 105)

Allgemein bezeichnet eine Föderation in der politischen Ordnung ein Bündnis, d. h. einen vertraglichen Zusammenschluss von Staaten und Staatenverbänden, die im Folgenden als Subjekte der Föderation bezeichnet werden. Aktuelle Beispiele für Föderationen sind die Bundesrepublik Deutschland und die Vereinigten Staaten von Amerika, jeweils als Zusammenschluss aus Teilstaaten, den Bundesländern bzw. Bundesstaaten, sowie die Europäische Union, als Zusammenschluss von Staaten.

Im Wesentlichen sind in allen Fällen die Subjekte der Föderation als eigene, an sich souveräne (Teil-)Staaten zu identifizieren, die über eine eigene Gesetzgebung verfügen und so bis zu einem gewissen Grad autonom sind. Durch den Eintritt in eine Föderation wird ein Teil dieser *Autonomie* aufgegeben und einzelne Aufgaben und Kompetenzen an die durch den

Zusammenschluss entstandene übergeordnete Institution abgegeben. Durch den Beitritt zu einer Föderation verpflichtet sich das Subjekt, bestimmte gemeinsame Regeln anzuerkennen und zu beachten.

Im Spannungsfeld zwischen weiter bestehender (Teil-)Autonomie und Verpflichtungen gegenüber der Föderation besteht ein wesentliches Konfliktpotenzial, das auf unterschiedliche Weise gelöst bzw. gemindert werden kann. Dazu gehört z. B. die konkurrierende Gesetzgebung, bei der die Föderation Rahmenbedingungen festlegt, innerhalb deren die Subjekte dann die genaue Ausgestaltung selbst bestimmen können.

Obwohl die Basisautonomie der Subjekte die wesentliche Gemeinsamkeit der genannten Beispiele ist, ergeben sich bei genauer Betrachtung, besonders im Hinblick auf die an einer Föderation beteiligten Subjekte, wesentliche Unterschiede. Sind im Falle der Bundesrepublik Deutschland die Bundesstaaten im Allgemeinen in ihrer inneren Struktur sehr ähnlich, so weisen die Staaten, welche die Europäische Union bilden, teilweise sehr unterschiedliche politische Strukturen auf. Dieses Kriterium wird als *Heterogenität* bezeichnet. Die genannten Beispiele verdeutlichen, dass sich nicht nur gleichartige (homogene), sondern auch verschiedenartige (heterogene) Subjekte im Rahmen einer Föderation zusammenschließen können.

Ergänzend ist festzuhalten, dass auch eine hierarchische Beziehung zwischen Föderationen vorliegen kann. So enthält die Europäische Union als Föderation das Subjekt Bundesrepublik Deutschland, das selbst eine Föderation aus den Subjekten der Bundesländer ist.

Die beiden Dimensionen Autonomie und Heterogenität spannen als wesentliche Komponenten im politischen Bereich eine Fläche auf, in der verschiedenartige Formen der Föderation möglich sind und die anhand dieser unterschieden werden können. Vertiefende Darstellungen sowohl zur historischen Entwicklung als auch zur konkreteren Ausgestaltung und den Unterscheidungsmerkmalen von politischen Föderationen finden sich z. B. in Nohlen und Schultze ([Nohlen/Schultze 1995]) und den darin referenzierten Quellen.

3.2 Der Begriff Föderation in der Betriebswirtschaftslehre

Der Begriff der Föderation als solcher ist in der Betriebswirtschaftslehre nicht weit verbreitet. Keen ([Keen 1990]) beschreibt föderierte Organisationen als Eigenschaften von Zentralisation und Dezentralisation verbindend. Leistungsfähige Kommunikationstechnologie ermöglicht es, dezentralisierte Organisationseinheiten, die relativ unabhängig voneinander agieren, zu einem gewissen Grad zentral zu überwachen und zu kontrollieren. Dies zwingt Organisationen nicht

zwischen zentralisiertem und dezentralisiertem Aufbau zu wählen (vgl. [Burriss 1993], [Heydebrand 1989], [Keen 1990]).

Das von Keen behandelte Themenfeld ist als Teil der Organisationsforschung seit zwei Jahrzehnten Gegenstand umfangreicher besonders organisationstheoretischer Forschung. Bezugnehmend auf den Begriffsursprung kann festgestellt werden, dass sich die in der Literatur diskutierten Unternehmensformen zwischen Hierarchie und Markt unter anderem ebenfalls nach den Dimensionen Heterogenität und Autonomie unterscheiden lassen. Heterogenität zeigt sich beispielsweise in der Größe oder den Kompetenzen der Unternehmen und der Stufe der Wertschöpfungskette, auf der sich die Unternehmen befinden. Autonomie lässt sich z. B. an der Rechtsform und den vertraglichen Regelungen, welche die Unternehmensformen kennzeichnen, festmachen.

Eine vertiefte Betrachtung findet an dieser Stelle nicht statt. Die bereits in Abschnitt 2.1.1 gegebene Einführung in die entsprechenden Fragestellungen berücksichtigte im Wesentlichen die Autonomie der beteiligten Organisationen als Unterscheidungsmerkmal. In diesem Unterkapitel wurde, im Rückgriff auf den politischen Ursprung des Begriffs der Föderation, diesem Verständnis als weiterer Aspekt die Heterogenität der Organisationen hinzugefügt.

3.3 Der Begriff Föderation in der Informationstechnologie

Bei der Übertragung eines Begriffs aus einem etablierten auf einen neuen Anwendungsbereich findet in der Regel eine Abstraktion statt. Durch diese Abstraktion verschwinden zwar Details aus dem Ursprungsbereich; die zentralen Eigenschaften und Prinzipien bleiben jedoch erhalten und werden auf den Zielbereich angepasst. Dies trifft auch für die Anwendung des Föderationsbegriffs auf Problemstellungen in der Informationstechnologie zu (vgl. [Conrad 1997], S. 4).

Heimbigner und McLeod ([Heimbigner/McLeod 1981]) wandten das Konzept der Föderation erstmalig im Bereich der IT an. Ihr Ansatz verfolgte das Ziel, den koordinierten Austausch und die gemeinsame Nutzung von verteilten rechnerbasierten Informationen zu ermöglichen und dabei unabhängig von der Kontrolle einer zentralisierten Organisation zu sein. Daran angelehnt ist nach Beasley et al. ([Beasley et al. 1994], S. 1) das Ziel einer Föderation von IT-Systemen darin zu sehen, die Kooperation zwischen autonomen Organisationen zum Zweck der gemeinsamen Nutzung von Diensten und Ressourcen zu vereinfachen. Problemfelder bei der Etablierung einer derartigen computergestützten Kooperation identifizieren sie sowohl im technischen, organisatorischen als auch prozessorientierten Bereich.

Der Fokus ihrer Definition liegt darauf, dass es sich bei den autonomen Organisationen um eigenständige Unternehmen handelt, so dass interorganisationale Beziehungen vorliegen. Dies stellt aber eine nicht notwendige Einschränkung dar, so dass im Kontext dieser Arbeit auch intraorganisationale Beziehungen, z. B. zwischen Abteilungen oder Profit-Centern, explizit mit eingeschlossen werden. Obwohl der Fokus der Definition auf der Föderation von IT-Systemen liegt, offenbart sie, dass neben der ausschließlich technischen weitere Dimensionen zu berücksichtigen sind. Die vorliegende Arbeit klammert diese willentlich von der Betrachtung aus und fokussiert sich im Wesentlichen auf die technische Dimension. Dies erfolgt zur Reduktion der weiterhin gegebenen Komplexität des Untersuchungsgegenstands. Obwohl organisatorische und prozessorientierte Problemfelder außerhalb des Untersuchungsbereichs liegen, müssen diese Bestandteile weiterer Forschungen sein, um ein ganzheitliches Modell zu erhalten.

Im Hinblick auf die Relevanz für den eigentlichen Forschungsgegenstand sind die im Folgenden dargestellten drei Bereiche der Nutzung von Föderationen in der IT gezielt aus einem breiteren Feld von Forschungsgebieten ausgewählt worden.

3.3.1 Föderierte Datenbanksysteme

Als Ausgangspunkt der Erforschung von Föderationen im Kontext von IT-Systemen hilft die Kenntnis der Fragestellungen und der Ergebnisse im Bereich föderierter Datenbanksysteme bei der Ableitung spezifischer Fragestellungen und Konzepte für artverwandte Forschungsgebiete, konkret für diese Arbeit föderierte Portale (vgl. Kapitel 4). Ziel dieses Unterkapitels ist die (selektive) Darstellung der für die weitere Untersuchung nutzbaren Analysen und Konzepte. Über die angegebenen Quellen kann das gleichwohl umfangreichere Gebiet föderierter Datenbanksysteme vertieft werden. Zusätzliche einführende und weitergehende Informationen zu verteilten und Multidatenbanksystemen finden sich z. B. bei Kemper und Eikler ([Kemper/Eickler 1999]), Bobak ([Bobak 1996]), Rahm ([Rahm 1994]) und Dadam ([Dadam 1996]).

Ausgehend von der Arbeit von Heimbigner und McLeod ([Heimbigner/McLeod 1981]) Anfang der 80er Jahre entwickelte sich durch ihre Veröffentlichung „A Federated Architecture for Information Management“ ([Heimbigner/McLeod 1985]) ab Mitte der 80er Jahre die Erforschung von föderierten Datenbanksystemen als neue Forschungsrichtung. Diese ist in vielen Bereichen auch die Grundlage für weitergehende Forschungen, bei denen in den 90er

Jahren eine Erweiterung auf föderierte Informationssysteme vorgenommen wurde (vgl. Abschnitt 3.3.2).

Als Haupttriebfeder für die Entstehung des neuen Forschungszweigs sehen Heimbigner und McLeod den Wandel in den Informationsgegebenheiten und -bedürfnissen zunehmend dezentralerer Büroumgebungen. Erfolgte die Entwicklung integrierter Datenbanksysteme im Wesentlichen im Hinblick auf die hoch integrierte und zentralisierte Massendatenhaltung, so entspricht dies nicht den Anforderungen zunehmender dezentralisierter Informationsverwendung. Statt einer zentralen Datenbank, die alle Unternehmensinformationen aufnimmt, besteht die Tendenz zur Proliferation von kleinen voneinander unabhängigen Datenbanken, ohne dass diese zentral gesteuert wird. In einem derartigen Umfeld ist es notwendig, ein gezieltes Management der Informationsflüsse zu betreiben, das auf Flexibilität, partielle Integration und gemeinsame Nutzung sowie Autonomie der Einzelsysteme ausgerichtet ist (vgl. [Heimbigner/McLeod 1985], S. 254).

Conrad pointiert die Entstehung dieses Forschungs- und Entwicklungszweigs besonders treffend. Zwar sei die Integration von Datenbeständen zur Schaffung eines einheitlichen Zugriffs eines der zentralen Ziele der Föderierung von Datenbanken, dieses werde allerdings nicht zum ersten Mal verfolgt. Vielmehr war es genau der Anspruch der Integration bis dahin verteilt gehaltener Daten einzelner Anwendungsprogramme, der überhaupt zur Entwicklung von traditionellen Datenbanksystemen und zu ihrem breiten Einsatz geführt hat (vgl. [Conrad 1997], S. 1).

3.3.1.1 Begriffsdefinition und Basisarchitektur

Föderierte Datenbanksysteme (FDBS) gehören zur Klasse der Multidatenbanksysteme (MDBS). MDBS zeichnen sich durch die Zusammenfassung mehrerer Datenbanksysteme (DBS) zu einem Verbund aus. Jedes DBS wird als Bestandteil des Verbunds als Komponenten-Datenbanksystem (engl. Component Database System, CDBS) bezeichnet. Das MDBS überwacht die Arbeit der CDBS, wobei jedes CDBS von einem eigenen Komponenten-Datenbankmanagementsystem (CDBMS) verwaltet wird.

Ein föderiertes Datenbanksystem ist in diesem Rahmen als eine Sammlung aus kooperierenden, aber *autonomen* CDBS definiert. Um eine substanzielle gemeinsame Informationsnutzung zu erreichen, wird die Koordination und teilweise Kontrolle der CDBS von einem föderierten Datenbankmanagementsystem (FDBMS) übernommen. Ein wichtiger Aspekt eines FDBS ist es, dass die CDBS weiterhin ihre lokalen Aufgaben erfüllen und gleichzeitig

an einer oder mehreren Föderationen beteiligt sein können (vgl. [Heimbigner/McLeod 1985], [Sheth/Larson 1990]).

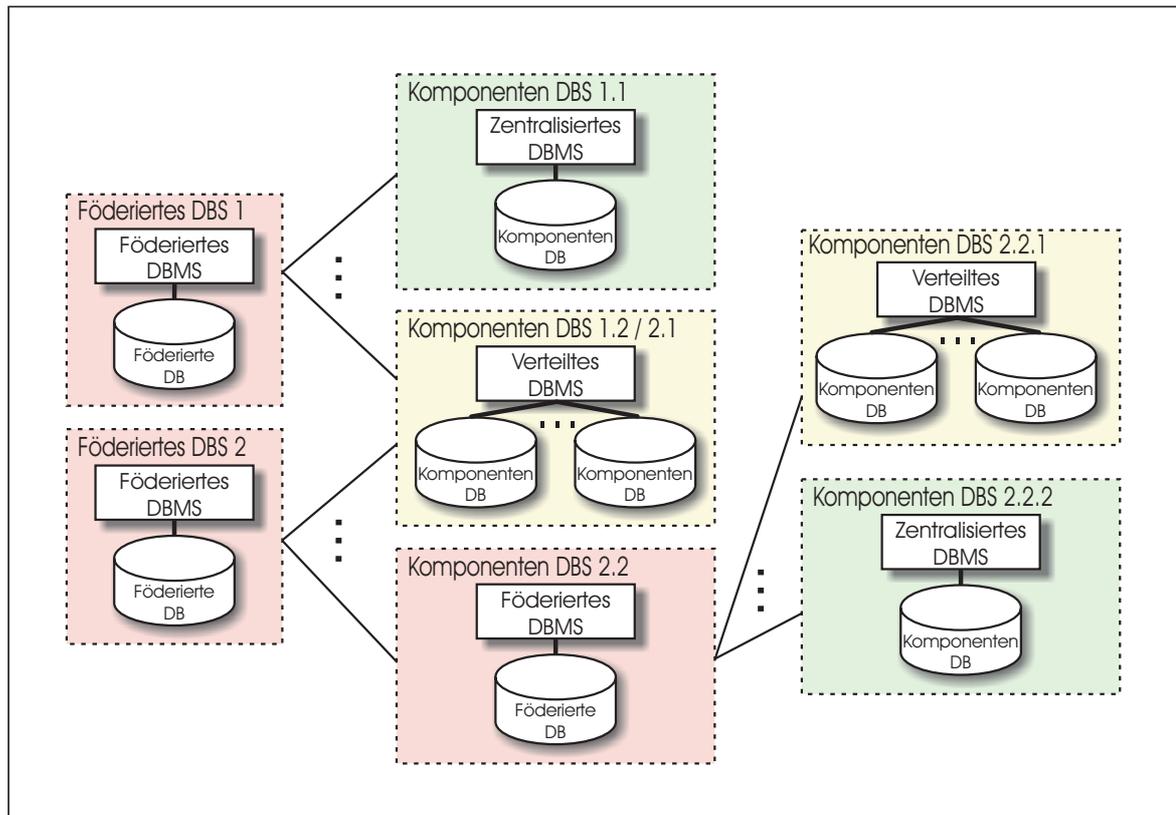


Abbildung 3-1: Beispiel einer Topologie eines föderierten Datenbanksystems
(vgl. [Davie/Botha 2001], S. 59)

Abbildung 3-1 zeigt eine mögliche Topologie eines FDBS. Eine Komponente kann dabei sowohl an mehreren Föderationen teilnehmen (CDBS 1.2 / 2.1), als auch selbst eine Föderation (CDBS 2.2), sowie sowohl zentralisiert (CDBS 1.1 und CDBS 2.2.2) als auch verteilt (CDBS 1.2 und CDBS 2.2.1) sein.

Conrad skizziert den allgemeinen Aufbau eines föderierten Datenbanksystems (FDBS) als über einen Föderationsdienst miteinander verbundene DBMS, die durch die Teilnahme an der Föderation zu CDBS werden. Der Föderierungsdienst kapselt alle Funktionen, z. B. das Zugriffsmanagement und die Ausführung verteilter Anfragen, die zur Realisierung einer Föderation notwendig sind. Abbildung 3-2 zeigt zur Veranschaulichung der Verbindung der DBMS untereinander den Föderierungsdienst als separate Komponente. Durch die Vermittlung dieser Verbindungskomponente können neue globale Anwendungen transparent auf den Gesamtdatenbestand des FDBS zugreifen. Conrad führt weiter aus, dass nicht alle Architekturen für FDBS eine separate Komponente vorsehen. Ist diese nicht vorhanden, geht er davon aus, dass die CDBS direkt miteinander kommunizieren können und globale Anwendungen,

die unabhängig von den einzelnen CDBS auf den Gesamtdatenbestand der Föderation zugreifen, nicht existieren (vgl. [Conrad 1997], S. 6).

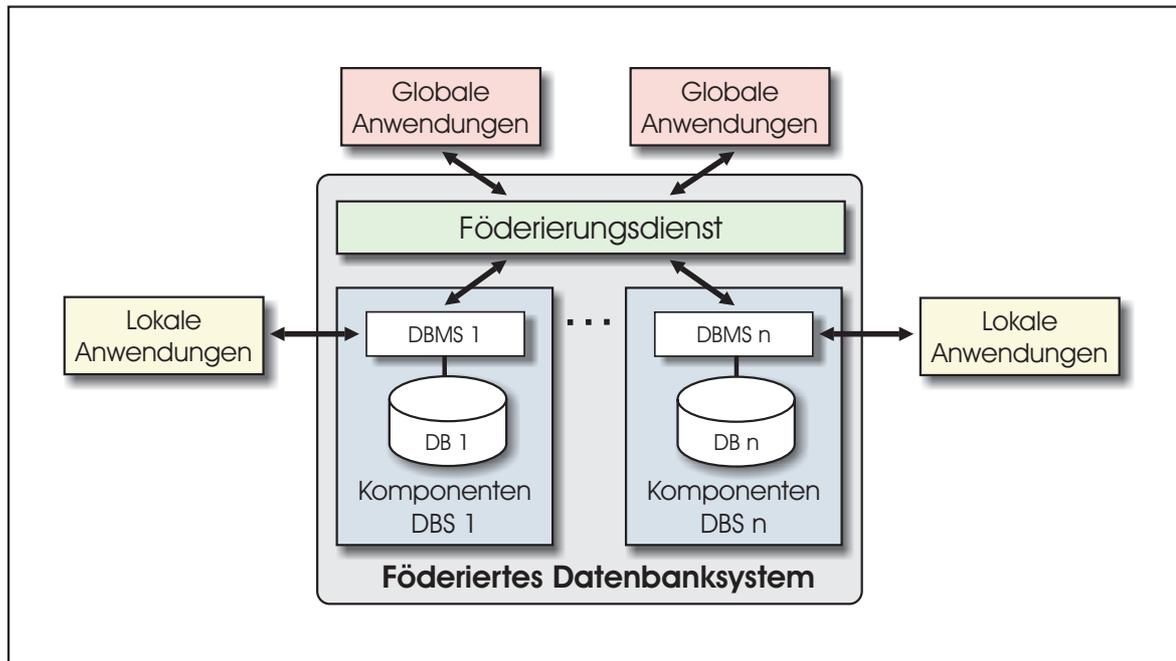


Abbildung 3-2: Allgemeine Architektur föderierter Datenbanksysteme
(vgl. [Conrad 1997], S. 7)

Obwohl bereits implizit in der vorherigen Aussage enthalten, ist gesondert herauszustellen, dass bereits vorhandene lokale Anwendungen a) ohne Änderungen weiter so eingesetzt werden können wie vor dem Beitritt des DBS zur Föderation und b) unter gewissen Bedingungen, z. B. gegebener Zugriffsberechtigung, nun auch auf die Datenbestände der geschaffenen Föderation zugreifen können.

Der logisch als separate Komponente definierte Föderierungsdienst kann in Anlehnung an Radeke (vgl. [Radeke 1996], S. 19 f.) – dort als Kopplungsschicht bezeichnet – physisch im Wesentlichen auf zwei Arten realisiert werden (vgl. Abbildung 3-3):

- *Punkt-zu-Punkt-Verbindungen*: Jedes CDBS wird um eine Föderierungsdienst-Schicht erweitert, die dafür verantwortlich ist, die Verbindung zu den anderen CDBS herzustellen und die Föderationsaufgaben wahrzunehmen.

Dieser Ansatz erfordert keine zentrale Föderationsinstanz, benötigt aber $n \cdot (n-1) / 2$ bidirektionale Verbindungen zwischen den CDBS, wobei n die Anzahl der beteiligten CDBS ist. Darüber hinaus sind die hier nicht näher spezifizierten Aufgaben der Föderation auf n Föderierungsdienst-Schichten zu verteilen und zu koordinieren.

- *Zentraler Föderierungsdienst*: Die CDBS sind über einen zentralen Föderierungsdienst miteinander verbunden, der sowohl die Kommunikation der CDBS untereinander koordiniert, als auch zentral alle Föderationsaufgaben wahrnimmt.

Dies verringert die Anzahl der notwendigen Verbindungen auf n , erfordert aber eine von den CDBS unabhängige, zentral zu implementierende und zu verwaltende Komponente.

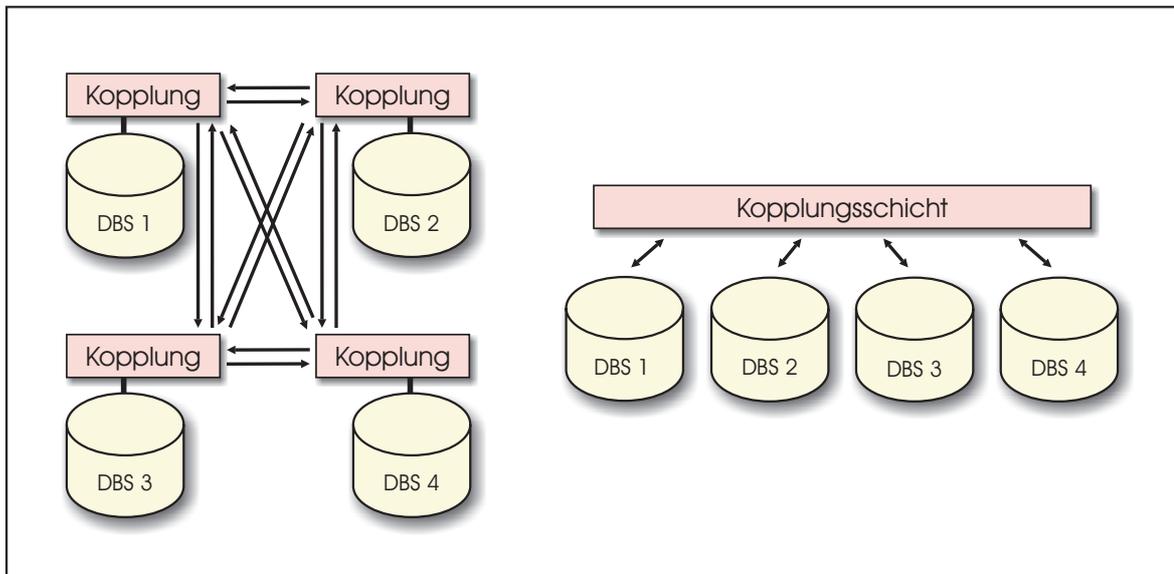


Abbildung 3-3: Alternativen zur physischen Realisierung des Föderationsdienstes
(in Anlehnung an [Radeke 1996], S. 19)

3.3.1.2 Charakteristika föderierter Datenbanksysteme

Zur Klassifizierung von Multidatenbanksystemen, zu denen föderierte Datenbanken gehören, lassen sich die Merkmale *Verteilung*, *Heterogenität* und *Autonomie* heranziehen. Die folgenden Ausführungen zu den Begriffen basieren im Wesentlichen auf den Darstellungen von Sheth und Larson ([Sheth/Larson 1990]), Conrad ([Conrad 1997]), Davie und Botha ([Davie/Botha 2001]) und Essmayr et al. ([Essmayr et al. 1995]) und können dort vertieft werden.

Verteilung

Hinsichtlich der Verteilung unterscheiden die meisten Autoren nach der Speicherung der Daten in einer oder mehreren Datenbanken. Conrad ([Conrad 1997], S. 45) erweitert dieses Verständnis um die Verteilung des Schemas, also der Meta-Daten. Abhängig davon, ob die Daten auf einem oder auf mehreren (ggf. geographisch verteilten) mit einem Kommunikationsnetzwerk verbundenen Rechnern abgelegt werden, wird von zentraler oder verteilter Speicherung gesprochen. Bei zentraler Datenspeicherung können die Daten auf mehrere Datenbanken verteilt werden, die jedoch von einem zentralen System verwaltet werden. Die

Art der Verteilung der Daten kann sehr unterschiedlich sein. Das Vorhalten mehrerer Kopien von Teilen oder der vollständigen Daten ist ebenso möglich wie eine im relationalen Verständnis vertikale oder horizontale Verteilung.

Die verteilte Datenspeicherung kann intentional erfolgen, um beispielsweise die Effekte erhöhter Verfügbarkeit und verkürzter Zugriffszeiten zu nutzen, wie es im Zusammenhang mit verteilten Datenbanksystemen immer beabsichtigt ist. Bei föderierten Datenbanksystemen ergibt sich die Verteilung der Daten jedoch aus dem Vorhandensein von mehreren voneinander unabhängigen Datenbanksystemen, die bereits vor dem eigentlichen Aufbau der Föderation bestehen.

Heterogenität

Heterogenität kann auf den Unterschieden der beteiligten DBMS basieren oder auf semantische Unterschiede zurückzuführen sein. *Heterogenität des DBMS* lässt sich in verschiedene Ebenen unterteilen:

- *System-Ebene*: Unterschiede in Bezug auf Hardware, Betriebssystemsoftware und Kommunikationssysteme können durch die Entwicklung von Standards in diesen Bereichen als weitgehend gelöst angesehen werden.
- *Datenmodell*: Unterschiedliche Datenmodelle (beispielsweise relational vs. objektorientiert) weisen verschiedene Strukturen und Modellierungskonstrukte auf, die bei der Modellierung eines Problems zu sehr unterschiedlichen und teilweise inkompatiblen Strukturen in den Schemata führen. Bedingt durch die verschiedenen Datenmodelle sind häufig Unterschiede in den Abfragesprachen, Integritätsbedingungen etc. vorzufinden.
- *Sicherheitsmodell*: Heterogene Sicherheitsmodelle unterscheiden sich hauptsächlich in Aspekten der Granularität, des Autorisationsparadigmas und des Zugriffskontrollmechanismus.

Die Identifikation von Heterogenität aufgrund unterschiedlicher DBMS ist in der Regel einfach möglich und für viele Fälle sind entsprechende Methoden zu ihrer Überwindung verfügbar.

Semantische Heterogenität entsteht dann, wenn kein einheitliches, übereinstimmendes Verständnis über die Bedeutung und Interpretation von gleichen oder zusammengehörigen Daten sowie über die beabsichtigte Verwendung besteht. Zwei der hieraus entstehenden semantischen Konflikte sind (vgl. [Visser et al. 1997], S. 3):

- gleiche Namen für unterschiedliche Konzepte (Homonyme) und
- verschiedene Namen für dasselbe Konzept (Synonyme).

Die Identifikation semantischer Probleme stellt sich häufig als schwierig dar, da Schemata für diese Aufgabe in der Regel keine ausreichenden Informationen bereitstellen. Aus diesem Grund gelingt die automatische Erkennung und Überwindung ausschließlich in spezifischen Fällen. Zur Lösung des allgemeinen Problems existieren nur wenige Ansätze.

Autonomie

Der Grad der Autonomie ist ein Maß dafür, inwieweit die organisatorischen Entitäten, die für die Verwaltung verschiedener DBS zuständig sind, in ihren Entscheidungen abhängig oder unabhängig sind. In einer Föderation ist davon auszugehen, dass die CDBS unter der voneinander unabhängigen Kontrolle verschiedener Administrationsbereiche stehen, die häufig nur dann bereit sind, anderen Zugriff auf die eigenen Daten zu geben, wenn sie selbst weiterhin die Kontrolle behalten. Essmayr et al. ([Essmayr et al. 1995]) identifizieren einen Trade-off zwischen der Funktionalität der Föderation und dem Grad der Autonomie, auf die der Administrationsbereich eines CDBS zu verzichten bereit ist. Dies führt ihren Ausführungen nach dazu, dass Verhandlungen zwischen den Partnern der Föderation darüber geführt werden müssen, inwieweit diese bereit sind, einen Teil ihrer Autonomie aufzugeben, um eine substanzielle Föderation aufzubauen.

Der allgemeine Begriff der Autonomie lässt sich in unterschiedliche Teilaspekte zerlegen, die eine differenzierte Analyse erlauben. Veijalainen und Popercu-Zeletin ([Veijalainen/Popercu-Zeletin 1988], S. 84 f.) unterscheiden zwischen Entwurfs-, Kommunikations- und Ausführungsautonomie. Sheth und Larson ([Sheth/Larson 1990], S. 187 f.) erweitern diese Begriffe um den Aspekt der Verbindungsautonomie und Jonscher und Dittrich ([Jonscher/Dittrich 1993], S. 6) führen zusätzlich den Begriff der Autorisierungsautonomie ein.

- *Entwurfsautonomie* ist unter zwei Aspekten zu betrachten. Einerseits sind die in eine Föderation zu integrierenden CDBS unabhängig voneinander entworfen worden. Dies lässt sich darauf zurückführen, dass der Aufbau einer Föderation in der Regel nachgelagert zum Aufbau der einzelnen DBS erfolgt, die Föderation also beim Entwurf der Einzelsysteme

noch nicht besteht. Die Auswahl der verwendeten Datenmodelle, semantischen Repräsentation etc. kann in diesem Fall vollständig an den zu erfüllenden Aufgaben orientiert werden und unterliegt keiner externen Steuerung durch die Föderation. Können durch die Föderationsebene keine Veränderungen in den lokalen Datenbankschemata vorgenommen werden, ist vollständige Entwurfsautonomie für die CDBS gegeben. Andererseits führt die konsequente Anwendung der Entwurfsautonomie dazu, dass die CDBS jederzeit in der Lage sind, ihre lokalen Datenbankschemata beliebig zu ändern.

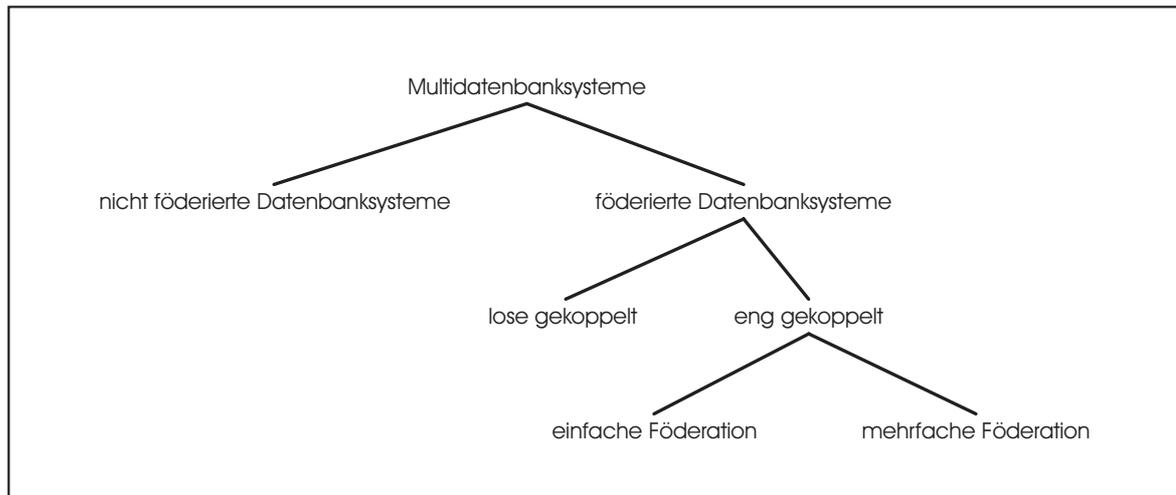
Heterogenität des Daten- und Sicherheitsmodells innerhalb von FDBS ist im Wesentlichen auf die Entwurfsautonomie der CDBS zurückzuführen.

- *Kommunikationsautonomie* ermöglicht es den beteiligten Administrationsbereichen der CDBS, selbst zu entscheiden, wann und mit wem die CDBS kommunizieren.
- *Ausführungsautonomie* gestattet es den beteiligten Administrationsbereichen der CDBS, selbst zu entscheiden, welche Anwendungsprogramme, Anfragen und Änderungsoperationen die CDBS ausführen und zu welchen Zeitpunkten.
- *Verbindungsautonomie* beinhaltet die Freiheit der beteiligten Administrationsbereiche der CDBS, selbst darüber zu entscheiden, einer Föderation beizutreten, diese jederzeit wieder zu verlassen und selbst zu bestimmen, mit welchen Daten die CDBS an der Föderation teilnehmen.
- *Autorisierungsautonomie* führt dazu, dass die beteiligten Administrationsbereiche der CDBS die Kontrolle darüber behalten, welche Benutzer der Föderation auf die lokalen Daten des CDBS zugreifen können oder nicht. Die Autorisierungsautonomie ist ein Bestandteil der Verbindungsautonomie und wird daher in der Literatur in der Regel nur dann explizit definiert, wenn föderierte Datenbanken unter dem Blickwinkel der besonderen Relevanz von Sicherheitsaspekten betrachtet werden.

Die detaillierte Darstellung der verschiedenen Arten von Autonomie und der mit ihnen verbundenen Auswirkungen verdeutlichen ebenfalls, dass zur Etablierung einer substanziellen Föderation eine teilweise Einschränkung der Autonomie notwendig ist. Aus pragmatischen Gründen ist es für eine Datenbankföderation etwa nicht möglich, eine jederzeit stattfindende Änderung des Datenbankschemas einer CDBS zuzulassen. Ähnliches trifft auch auf die anderen Arten der Autonomie zu.

3.3.1.3 Taxonomie von Multi- und föderierten DBMS

Ausgehend von den in den vorherigen Abschnitten dargestellten Charakteristika Verteiltheit, Heterogenität und Autonomie von Datenbanksystemen wurden verschiedene Taxonomien entwickelt. Eine erste von Sheth und Larson ([Sheth/Larson 1990], S. 189 f.) eingeführte Taxonomie berücksichtigt ausschließlich die Autonomie der CDBS und ist in Abbildung 3-4 dargestellt.



**Abbildung 3-4: Taxonomie von DBS unter Berücksichtigung der Autonomie der CDBS
(in Anlehnung an [Sheth/Larson 1990], S. 190)**

Ausgehend von der Definition von MDBS in Abschnitt 3.3.1.1 können diese in zwei Hauptkategorien unterschieden werden. Ein „nicht föderiertes Datenbanksystem“ ist eine Integration von CDBS, die *nicht* autonom sind, d. h. vollständig auf der übergeordneten Ebene verwaltet werden. Dies führt dazu, dass nur eine Ebene des Managements der beteiligten CDBS existiert und alle Operationen gleichartig ausgeführt werden. Ein nicht föderiertes DBS unterscheidet nicht zwischen lokalen und nicht lokalen Benutzern. Es erscheint den Benutzern logisch wie ein verteiltes DBS. Im Gegensatz dazu besteht ein „föderiertes Datenbanksystem“ aus an sich autonomen CDBS, die aber zum Zweck der (partiellen) gemeinsamen Nutzung von Daten an der Föderation teilnehmen und auf einen Teil ihrer Autonomie verzichten. Es existiert keine zentrale Kontrolle. Jedes CDBS respektive dessen Administratoren behalten die Kontrolle über den Zugriff auf die von ihnen zu verwaltenden Daten. Im Wesentlichen korrespondiert diese Unterscheidung mit der in Abschnitt 3.1 dargelegten politischen Interpretation des Begriffs der Föderation.

Föderierte DBS können abhängig davon, wer die Föderation verwaltet, weiterhin in lose bzw. eng gekoppelte Systeme untergliedert werden. Bei loser Kopplung ist es die Aufgabe des Benutzers, die Föderation für sich aufzubauen und zu verwalten. Da keine zentrale Admi-

nistration der Föderation existiert, liegt auch die vollständige Kontrolle – unter Berücksichtigung der Rechte auf den CDBS – in den Händen der Benutzer. Verschiedene Benutzer können unabhängig voneinander ihre eigene Föderation betreiben, was grundsätzlich einen hohen Grad an Flexibilität erlaubt. Eine Datenbankföderation wird als eng gekoppelt bezeichnet, wenn die Administratoren der Föderation dafür verantwortlich sind, diese aufzubauen, zu betreiben und eine aktive Rolle bei der Verwaltung des Zugriffs auf die einzelnen CDBS einnehmen. Dazu muss eine Abstimmung zwischen den Administratoren der Komponenten und denen der Föderation darüber stattfinden, auf welche Daten global, in welcher Form und von wem zugegriffen werden kann. Bei unterschiedlichen Interessen der beteiligten Benutzer und Administratoren müssen ggf. Kompromisse gefunden werden. Die Benutzer können sich auf die eigentliche Nutzung beschränken, ohne den Aufwand des Aufbaus und der Pflege der Föderation selbst zu tragen.

Als letzte Kategorisierung unterscheiden Sheth und Larson eng gekoppelte FDBS danach, ob sie auf die Bildung einer globalen, föderierten Sicht (Schema) beschränkt sind (Single Federation), welche die integrierte Beschreibung aller in die Föderation einfließenden Daten darstellt, oder ob es nebeneinander mehrere föderierte Sichten geben kann (Multiple Federations), die ggf. jeweils nur einen für den Anwendungszweck spezifischen Ausschnitt der benötigten Daten der Föderation abbilden. Bei loser Kopplung ist es immer implizit möglich, mehrere Sichten zu erzeugen.

Ausgehend von dieser eindimensionalen auf die Autonomie beschränkten Taxonomie, ist von Özsü und Valduriez ([Özsü/Valduriez 1991], S. 69 f.) eine alle drei Dimensionen einbeziehende Taxonomie vorgeschlagen worden. Der dadurch aufgespannte Raum ist in Abbildung 3-5 visualisiert. Obwohl ausschließlich die Extremausprägungen benannt sind und so eine jeweils dichotome Merkmalsausprägung suggeriert wird, sind die Merkmalsausprägungen als in weiten Teilen kontinuierlich aufzufassen.

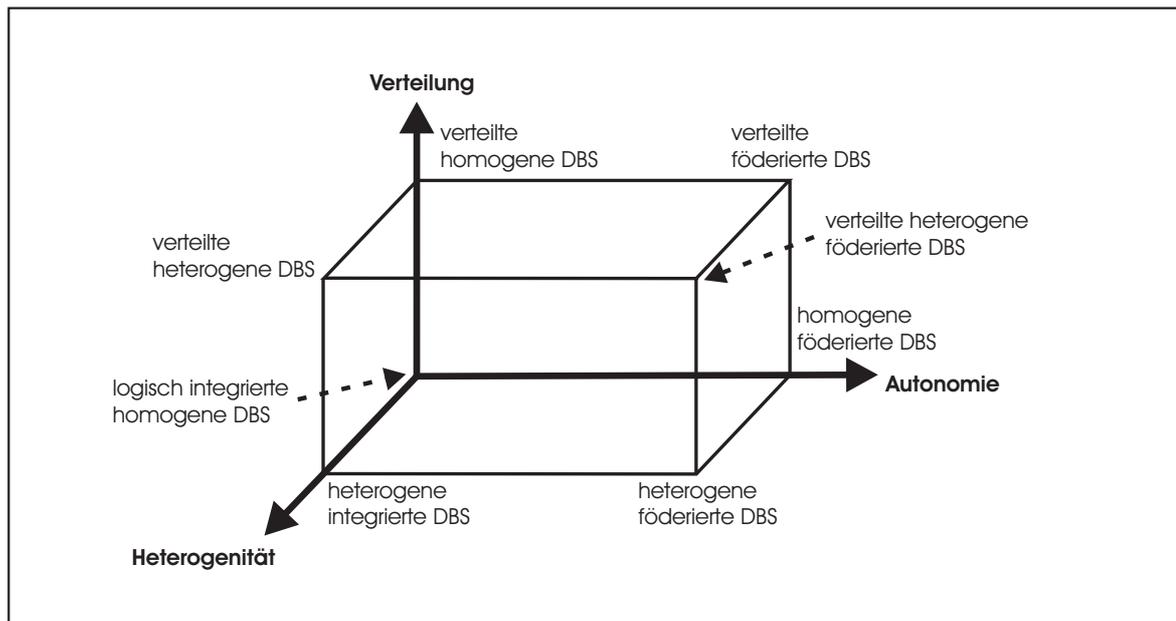


Abbildung 3-5: Taxonomie von DBS unter Berücksichtigung der Dimensionen Verteilung, Heterogenität und Autonomie
(in Anlehnung an [Özsu/Valduriez 1991], S. 70)

3.3.1.4 Aspekte der Verteilung

Obwohl bei verteilten Datenbanken die Daten verteilt gespeichert werden, fordert Date in diesem Zusammenhang, dass ein verteiltes System sich für den Benutzer wie ein *nicht* verteiltes System darstellen muss (vgl. [Date 1990], S. 654).

Ziele bei der Verteilung

Grundsätzliches Ziel der Verteilung ist, diese für den Benutzer transparent zu gestalten, so dass sie für ihn keine offensichtliche Bedeutung und Auswirkung hat. Diese allgemeine Transparenzforderung gliedert Date in 12 Anforderungen (vgl. [Date 1990], S. 656 ff.), von denen im Folgenden nur auf diejenigen eingegangen wird, die direkt Aspekte der Verteilung und nicht ausschließlich der Heterogenität oder Autonomie betreffen.

Durch die *Unabhängigkeit von einem zentralen System* kann gewährleistet werden, dass dieses nicht zum Flaschenhals der Verteilung wird. Darüber hinaus kann vermieden werden, dass ein zentrales System den „Single point of failure“ darstellt. Dies ist andernfalls dadurch gegeben, dass bereits der Ausfall des zentralen Systems einen Ausfall des Gesamtsystems nach sich zieht, da alle anderen Systeme von diesem abhängig sind.

Bei der *Standortunabhängigkeit* (auch als *Standorttransparenz* bezeichnet) ist es aus Benutzersicht nicht relevant zu wissen, wo die Daten, auf die er zugreift, physikalisch abgelegt sind. Er sollte dazu in der Lage sein, so zu verfahren, als ob die Daten – zumindest aus logischer Sicht – an einer zentralen Stelle gespeichert sind.

Die *Fragmentierungsunabhängigkeit* (auch als *Fragmentierungstransparenz* bezeichnet) verfolgt das Ziel, den Kommunikationsaufwand zwischen den verteilten Datenbanken zu reduzieren, indem die Daten dort abgelegt werden, wo am häufigsten auf sie zugegriffen wird. Aus Sicht des Benutzers sollte es ebenfalls nicht relevant sein, ob und wie die Daten fragmentiert sind; er sollte auf sie zugreifen können, als ob sie – zumindest aus logischer Sicht – nicht fragmentiert sind.

In enger Verbindung mit der Fragmentierungsunabhängigkeit steht die *Replizierungsunabhängigkeit* (auch als *Replizierungstransparenz* bezeichnet). Besteht an mehreren Stellen die Notwendigkeit des häufigen Zugriffs, können Kopien der betroffenen Daten an den entsprechenden Stellen vorgehalten werden. Dies führt im Allgemeinen zu einer Erhöhung der Zugriffsgeschwindigkeit und der Verfügbarkeit der Daten. Gleichwohl ergibt sich die Notwendigkeit, die Kopien an allen Stellen synchron zu halten. Auch hier muss es für den Benutzer ohne Bedeutung sein, ob die Daten repliziert werden oder nicht. Er sollte auf sie zugreifen können, als ob sie – zumindest aus logischer Sicht – nicht repliziert sind.

Mithilfe eines *verteilten Transaktionsmanagements* ist sicherzustellen, dass nebenläufige Lese- und Schreiboperationen nicht zu Inkonsistenzen und Integritätsverletzungen führen. Zusätzlich ist im Fehlerfall eine kontrollierte Behandlung und Rücknahme der evtl. unvollständig ausgeführten Transaktion zu gewährleisten.

Die *verteilte Anfragebearbeitung* umfasst zwei Aspekte. Zum einen ist zu entscheiden, welche der verteilten Datenbanken an der Erfüllung der Anfrage zu beteiligen sind. Dies hat unter Berücksichtigung sowohl möglicherweise vorhandener Fragmentierung als auch Replizierung zu geschehen. Darüber hinaus spielt die Optimierung hinsichtlich der Zugriffs- und Antwortgeschwindigkeit eine entscheidende Rolle. Die Kosten der Kommunikation mit den beteiligten Systemen sind zu minimieren. Als Kostenkriterium sind verschiedene Messgrößen denkbar, etwa die realen Kommunikationskosten oder die Länge der durchschnittlichen Antwortzeit.

Probleme und Ansätze bei der Verteilung

Die Berücksichtigung der genannten Ziele beim Aufbau von verteilten Datenbanken führt zu zahlreichen Problemen. Aus den von Date identifizierten Bereichen (vgl. [Date 1990], S. 666 ff.) wird wegen deren besonderer Relevanz für die späteren Betrachtungen im Folgenden ausschließlich auf das „Katalog-Management“ und die „Verteilung von Aktualisierungen“ eingegangen.

Das *Katalog-Management* ist insofern bei verteilten Datenbanksystemen relevant, als diese neben den Informationen nicht verteilter Datenbanksysteme zusätzlich Steuerungsinformationen benötigen, um die Forderungen nach Standort-, Fragmentierungs- und Replizierungsunabhängigkeit erfüllen zu können und sich daraus die Frage ergibt, wo der Katalog selbst oder Teile von ihm gespeichert sein sollen. Es existieren verschiedene Ansätze, zu denen die zentrale, die vollständig replizierte, die partitionierte/verteilte oder eine Kombination der zentralen mit der partitionierten Ablage gehören. Alle genannten Ansätze beinhalten negative Interdependenzen mit den von Date formulierten Zielen. Daher ist – abhängig davon, wie diese im konkreten Fall gewichtet werden – eine der zuvor genannten Strategien entweder direkt anwendbar oder muss angepasst werden.

Durch die Nutzung von Replikation kommt der *Verteilung von Aktualisierungen* besondere Bedeutung zu. Gegenüber der gängigen Definition, wonach Replikation als Akt oder Resultat der Reproduktion beschrieben wird (also kurz als Erzeugen einer Kopie), definiert Buretta Replikation im IT-Bereich folgendermaßen:

“[...] replication is much more than simply the copying of any object; it must also address the implementation and management of the complete copying process. In essence, replication is a copy management service.” ([Buretta 1997], S. 4)

Serain ([Serain 2002], S. 13) weist zu Recht darauf hin, dass Replikation von Daten nicht mit Verteilung von Daten gleichzusetzen ist. Im ersten Fall existieren die gleichen Daten in mehreren Instanzen, im zweiten Fall existiert jedes Datenobjekt nur einmal und die Menge der Daten ist über mehrere Systeme verteilt. Gleichwohl werden beide Aspekte häufig gemeinsam eingesetzt.

Grundsätzlich ist beim Einsatz von Replikation zu gewährleisten, dass die Veränderung einer Replik dazu führt, dass auch alle anderen Repliken aktualisiert werden. Hierbei lassen sich die beiden Grundmodelle der *synchronen* und der *asynchronen* Systeme unterscheiden (vgl. [Buretta 1997], S. 9 ff.).

Synchrone Replikation erlaubt es, jederzeit die Konsistenz zwischen den Datenbanken zu garantieren, indem Aktualisierungen von ihrem Ursprung unmittelbar ohne Latenz über ein dediziertes Protokoll an alle anderen Datenbanken weitergegeben werden. Neben den Auswirkungen spezieller Protokolle zum Verteilen der Aktualisierungen auf die Geschwindigkeit des synchronen Ansatzes ist das Problem der Verfügbarkeit aller beteiligten Datenbanken kritisch zu betrachten. Sollten zum Zeitpunkt der Aktualisierung eine oder mehrere Datenbanken auf Grund von Netzwerk- oder Datenbankproblemen nicht verfügbar sein, würde die Aktualisierung fehlschlagen. Das kann dazu führen, dass die durch die Replikation angestrebte höhere Verfügbarkeit gegenüber dem nicht replizierten Zustand negiert wird.

Asynchrone Replikation ist demgegenüber nicht in der Lage, die Konsistenz zwischen den Datenbanken jederzeit zu garantieren; daher wird auch von weicher Konsistenz gesprochen. Die Latenz ist immer größer null und die Aktualisierung erfolgt ausschließlich asynchron und losgelöst von der sie auslösenden Aktualisierung. Diese Form der Replikation bietet die Möglichkeit, das zuvor dargestellte Problem zu umgehen, indem beispielsweise die Aktualisierung erfolgreich abgeschlossen wird, obwohl ein oder mehrere Systeme zum entsprechenden Zeitpunkt nicht verfügbar waren. An diese werden die Aktualisierungen bei deren Wiederverfügbarkeit übertragen. Mit diesem Vorteil geht aber gleichzeitig die Gefahr einher, dass Inkonsistenzen entstehen, indem beispielsweise in Konflikt zueinander stehende Aktualisierungen auf noch nicht synchronen Datenbanken vorgenommen werden.

Zur Konzeptionalisierung der replizierten Datenhaltung werden die beiden Modelle Master/Slave und Update-anywhere verwendet, die angelehnt an Buretta ([Buretta 1997], S. 44 ff.) folgendermaßen charakterisiert werden können:

- Beim *Master/Slave-Modell* wird jedem einzelnen Datenelement explizit ein Master, auch als primäre Quelle bezeichnet, zugewiesen. Vereinbarungsgemäß ist eine Aktualisierung nur gegenüber der primären Quelle möglich, alle anderen sekundären Quellen haben nur Leseberechtigung. Durch die Einführung einer primären Quelle je Informationseinheit wird die Replikation vereinfacht. Da jederzeit eine verbindliche Instanz der Informationseinheit existiert, können Kopien zentral abgeglichen und wieder in einen konsistenten Zustand überführt werden. Bei Verwendung dieses Modells wird teilweise die Unabhängigkeit von einem zentralen System aufgegeben. Dieses Problem tritt insofern auf, als eine Änderung eines Datenelements, dessen primäre Quelle nicht verfügbar ist, nicht möglich ist.

- Das *Update-anywhere-Modell* basiert auf der Gleichberechtigung aller Quellen. Aktualisierungen können jederzeit an jeder Quelle vorgenommen werden. Bei asynchroner Replikation können bei diesem Modell Konflikte auftreten. Diese sind jedoch nicht bereits im Vorfeld, sondern erst zum Zeitpunkt der Replikation erkennbar und müssen durch automatische oder manuelle Konfliktauflösungsstrategien korrigiert werden. Mithilfe eines entsprechenden Systementwurfs der zugreifenden Applikationen und organisatorischen Regelungen kann die Konflikthäufigkeit jedoch verringert werden.

3.3.1.5 Aspekte der Heterogenität und Autonomie

Die Dimensionen der Heterogenität und Autonomie werden in der Literatur zu föderierten Datenbanken ebenfalls durch einen konzeptionellen Ansatz berücksichtigt.

Die ANSI/X3/SPARC Study Group für Datenbanksysteme hat bereits Ende der 70er Jahre einen Vorschlag für eine Drei-Ebenen-Datenbeschreibungsarchitektur vorgelegt (vgl. [Tsichritzis/Klug 1978]). Diese hat zum Ziel, die Datensicht des Benutzers und der Applikationen von der tatsächlichen physikalischen Speicherung logisch zu entkoppeln und die technischen Details der Datenbank zu verbergen. Die logische Entkopplung, auch als Abbildung (Mapping) bezeichnet, wird durch die Verwendung unterschiedlicher Schemata erreicht.

Die in Abbildung 3-6 dargestellte ursprüngliche Drei-Ebenen-Schemaarchitektur umfasst das interne und das konzeptionelle Schema sowie die externen Schemata. Das *interne Schema* beschreibt die konkreten physikalischen Eigenschaften und die Speicherung der Daten. Das *konzeptionelle Schema* beschreibt die Daten der Datenbank durch die Nutzung des logischen Modells der verwendeten Datenbank (z. B. relationales oder objektorientiertes Modell) und lässt sich in einer von den internen Strukturen unabhängigen und durch das logische Modell vorgegebenen Datenbeschreibungssprache (Data Definition Language, DDL) ausdrücken. Darüber hinaus definiert das logische Modell mithilfe einer Datenmanipulationssprache (Data Manipulation Language, DML) Operationen, die auf dem konzeptionellen Schema ausgeführt werden können. Die *externen Schemata* bilden jeweils einen Ausschnitt des konzeptionellen Schemas ab, der an die konkreten Anforderungen der jeweiligen Benutzer und Applikationen angepasst ist.

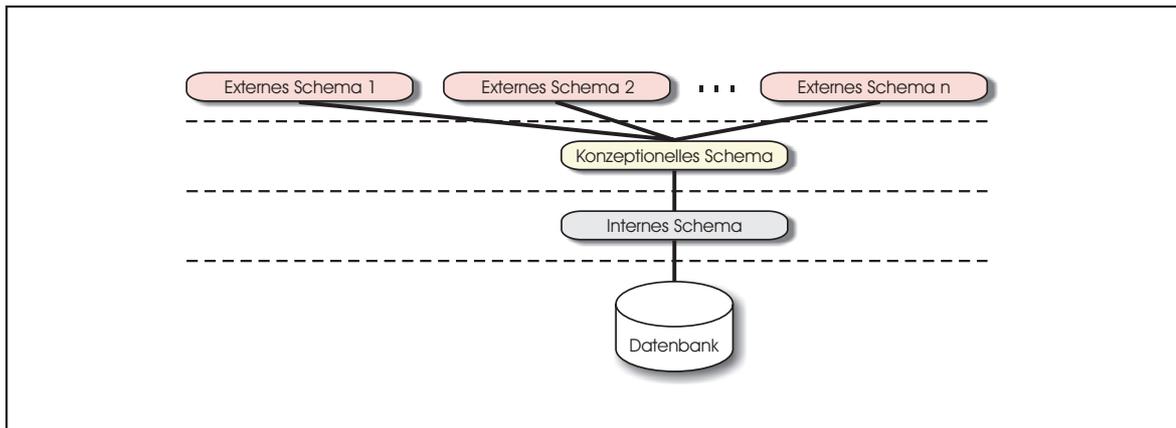


Abbildung 3-6: Drei-Ebenen ANSI/SPARC DBMS Schemaarchitektur
(vgl. [Davie/Botha 2001], S. 62)

Der Terminologie von Sheth und Larson ([Sheth/Larson 1990], S. 192 ff.) folgend kommen die drei Typen „Zugriffs-“, „Transformations-“ und „Filterungsprozessor“ zur Anwendung, um die verschiedenen Schemata im Fall der Drei-Ebenen-Architektur aufeinander abzubilden (vgl. Abbildung 3-7).

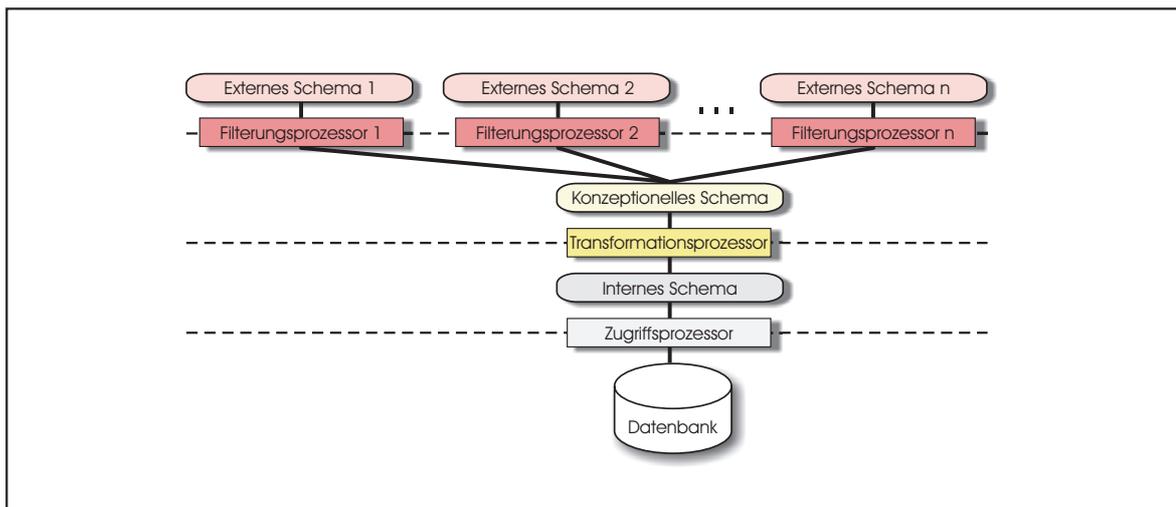


Abbildung 3-7: Komponenten der Drei-Ebenen-Schemaarchitektur
(vgl. [Sheth/Larson 1990], S. 198)

Der *Zugriffsprozessor* nimmt Kommandos und Daten entgegen und liefert seinerseits Daten zurück, indem er die Kommandos in einer konkreten Datenbank ausführt. Der *Transformationsprozessor* übersetzt Kommandos aus einer Sprache (Ausgangssprache) in eine andere Sprache (Zielsprache) oder transformiert Daten eines Formats (Ausgangsformat) in die Repräsentation eines anderen Formats (Zielformat). Transformationsprozessoren sorgen für die Überbrückung der Heterogenität zwischen verschiedenen Datenmodellen, was als Datenmodelltransparenz bezeichnet wird. *Filterungsprozessoren* ermöglichen es, die Kommandos und Daten, die von Prozessor zu Prozessor weitergegeben werden können, einzuschränken.

Sie können sowohl syntaktische, semantische als auch zugriffsbedingte Filterungsaufgaben erfüllen.

Obwohl die Drei-Ebenen-Schemaarchitektur für zentralisierte DBMS ausreichend ist, kann sie für FDBS nicht direkt verwendet werden und bedarf der Erweiterung. Diese wurde erstmals umfassend von Sheth und Larson ([Sheth/Larson 1990], S. 198 ff.) in Form der Fünf-Ebenen-Schemaarchitektur für FDBS vorgenommen (vgl. Abbildung 3-8), die als quasi Referenzarchitektur für FDBS gilt.

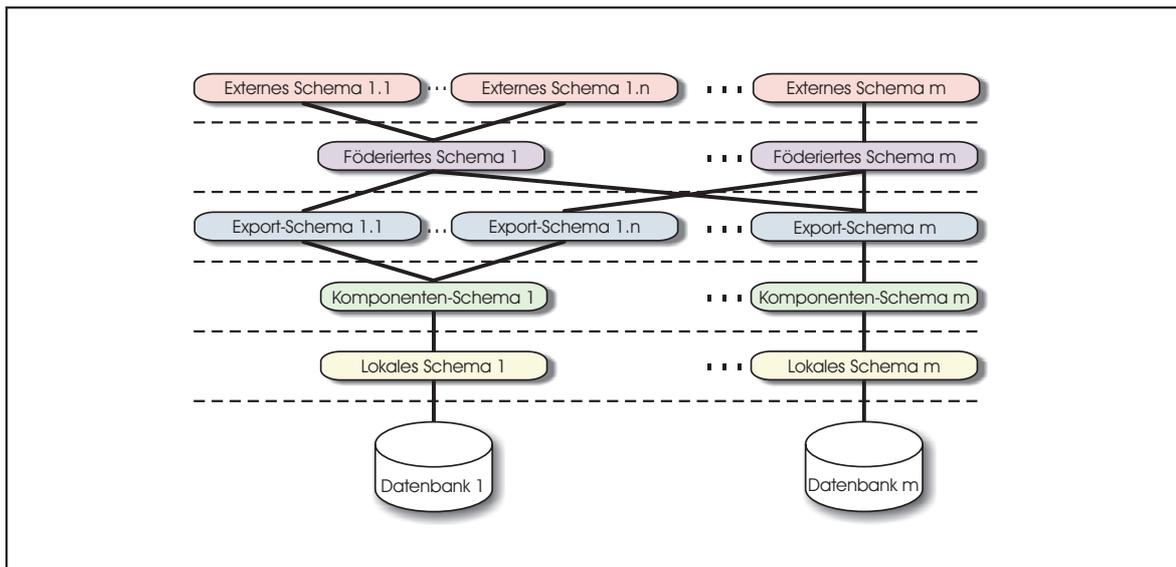


Abbildung 3-8: Fünf-Ebenen-Schemaarchitektur eines FDBS
(vgl. [Sheth/Larson 1990], S. 199)

Das *lokale Schema* entspricht hierbei dem konzeptionellen Schema aus der Drei-Ebenen-Schemaarchitektur (das interne Schema ist kein Bestandteil der Fünf-Ebenen-Schemaarchitektur) und wird im nativen Datenmodellformat des CDBS formuliert. Dadurch ist es möglich, dass verschiedene lokale Schemata durch verschiedene Datenmodelle repräsentiert werden.

Das *Komponenten-Schema* wird aus den lokalen Schemata durch Übersetzung in das so genannte Canonical oder Common Data Model (CDM) des FDBS abgeleitet. Durch die Komponenten-Schemata ist es möglich, unterschiedliche lokale Schemata in eine einheitliche Darstellung zu überführen. Gleichzeitig können evtl. in den Komponenten-Schemata fehlende, semantische Informationen ergänzt werden. Das Komponenten-Schema berücksichtigt im Wesentlichen die Konsequenzen der Entwurfsautonomie.

Das *Export-Schema* definiert eine Untermenge der Operationen und Daten des Komponenten-Schemas, die der Föderation zur Verfügung gestellt werden sollen. Hierbei können auch Zu-

griffsbeschränkungen für die Benutzer der Föderation berücksichtigt werden. Export-Schemata ermöglichen die Realisierung der Verbindungs- und Autorisierungsautonomie.

Das *föderierte Schema* stellt die Integration mehrerer Export-Schemata dar. Dabei enthält es die notwendigen Informationen, um die Verteilung der Daten abzubilden. Abhängig davon, ob es mehrere Klassen von föderierten Benutzern gibt, kann es ein oder mehrere föderierte Schemata geben. Als Klassen von föderierten Benutzern werden Gruppen von Benutzern und/oder Applikationen bezeichnet, die eine Anzahl miteinander in Beziehung stehender Aktivitäten ausführen und daher logisch gruppiert werden können.

Die letzte Ebene der Fünf-Ebenen-Schemaarchitektur umfasst die *externen Schemata*. Das externe Schema definiert einen Ausschnitt des föderierten Schemas für eine spezifische Benutzergruppe oder Gruppe von Applikationen. Durch das externe Schema lässt sich die Komplexität des föderierten Schemas auf das für den Anwendungsfall notwendige reduzieren. Zusätzlich lassen sich weitere Integritätsbedingungen und Zugriffsbeschränkungen festlegen.

Um die verschiedenen Schemata im Fall der Fünf-Ebenen-Architektur aufeinander abbilden zu können (vgl. Abbildung 3-9), führen Seth und Larson ([Sheth/Larson 1990], S. 195 f.) den „Aggregierungsprozessor“ als weiteren Prozessor-Typ ein.

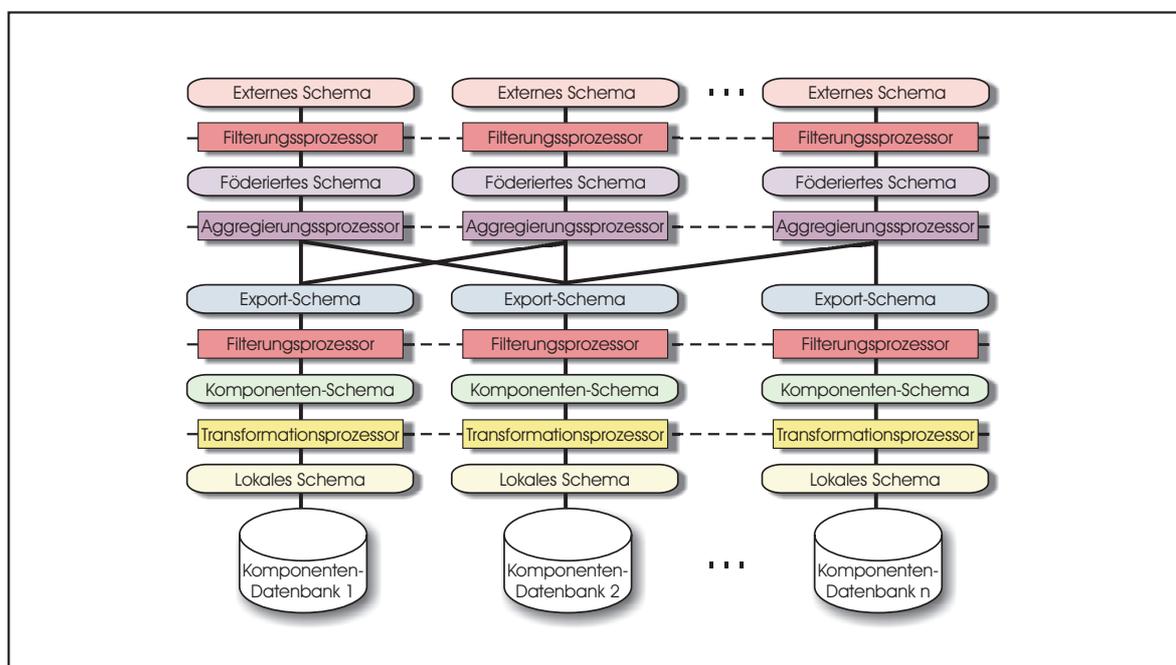


Abbildung 3-9: Komponenten der Fünf-Ebenen-Schemaarchitektur
(vgl. [Sheth/Larson 1990], S. 199)

Der *Aggregierungsprozessor* kann die Operationen, mit denen er von einem anderen Prozessor aufgerufen wird, aufteilen und an zwei oder mehrere andere Prozessoren weitergeben. Daten, die von mehreren Prozessoren erzeugt wurden, können andererseits von einem

Aggregierungsprozessor zusammengeführt und an einen anderen Prozessor weitergegeben werden. Durch diese Funktionalitäten unterstützt der Aggregierungsprozessor sowohl die Standort-, die Verteilungs- als auch die Replizierungstransparenz.

Zusammenfassend ist festzuhalten, dass die auf fünf Ebenen erweiterte Schemaarchitektur in besonderem Maße die Heterogenität und Autonomie von FDBS berücksichtigt und auch Aspekte der Verteilung einbezieht. Die Autonomie der CDBS wird dadurch gewahrt, dass diese über das Export-Schema kontrollieren, welche Daten nach außen für die Föderation sichtbar gemacht werden. Das Problem der Heterogenität der CDBS wird durch die Übersetzung in das CDM gelöst. Durch das Hinzufügen, Weglassen bzw. durch Umorganisation der Komponenten kann die Fünf-Ebenen-Schemaarchitektur weiterhin an andere und sich wandelnde Anforderungen angepasst werden.

3.3.1.6 Sicherheit in föderierten Datenbanksystemen

Obwohl Fragen der Sicherheit in FDBS bereits von Sheth und Larson ([Sheth/Larson 1990], S. 220 f.) angedeutet wurden, weisen Jonscher und Dittrich ([Jonscher/Dittrich 1994], S. 24]) darauf hin, dass der Systementwurf sich zu sehr auf funktionale Notwendigkeiten beschränkt, Sicherheitsaspekte aber nur unzureichend berücksichtigt.

Besonders im Zusammenhang mit FDBS ist Sicherheit bereits beim Systementwurf zu berücksichtigen, da es von deren Gewährleistung abhängt, ob eine Föderation überhaupt zu Stande kommt. Die für die CDBS Verantwortlichen werden in der Regel nur dann bereit sein, ihr System in eine Föderation einzubringen, wenn ihre Daten innerhalb der Föderation ebenso sicher verwaltet werden wie in ihren lokalen Systemen. Essmayr et al. ([Essmayr et al. 1995]) führen hierzu aus, dass das Sicherheitssystem der Föderation einerseits mindestens so sicher sein muss wie jedes der lokalen Systeme, andererseits sollte es aber für die Benutzer der Föderation so weit wie möglich transparent sein.

Aus diesem Grund geht dieses Unterkapitel ausführlich auf ausgewählte Aspekte der Sicherheit in FDBS ein. Hierzu werden zuerst allgemeine Zugriffskontrollkonzepte und anschließend die sich aus den Dimensionen Verteilung, Heterogenität und Autonomie (vgl. Abschnitt 3.3.1.2) ergebenden Fragestellungen für FDBS dargestellt. Die Konzepte werden ausschließlich unabhängig von konkreten technischen Implementierungen oder Verfahren betrachtet.

Die Ausführungen basieren im Wesentlichen auf den Arbeiten von Jonscher und Dittrich ([Jonscher/Dittrich 1994], [Dittrich/Jonscher 1994]), Essmayr et al. ([Essmayr et al. 1995]), Jajodia und Wijesekera ([Jajodia/Wijesekera 2001]), Davie und Botha ([Davie/Botha 2001]) und di Vimercati und Samarati ([di Vimercati/Samarati 1996]).

Zugriffskontrollkonzepte

Die Zugriffskontrolle stellt einen wesentlichen Bestandteil von Sicherheit dar. Sie wird als Zusammenfassung aller Systemmechanismen verstanden, die benötigt werden, um eine Anfrage eines bestimmten Subjekts (aktive Entität eines Systems) daraufhin zu überprüfen, ob diese ausgeführt werden darf oder nicht, um dann die sich daraus ergebende Entscheidung entsprechend umzusetzen. Dabei basiert die Zugriffskontrolle auf einer festgelegten Vorschrift, die diese steuert. Grundlage sind Zugriffsrechte, die es den Subjekten erlauben oder verbieten, eine spezifische Aktion oder Operation auf einem zugriffsgeschützten Objekt auszuführen. In Abhängigkeit davon, welche Zugriffsrechte spezifiziert und durchgesetzt werden, lassen sich zwei wesentliche Ansätze unterscheiden (vgl. [Denning 1982]).

Der *Mandatory Access Control (MAC)*-Ansatz wurde zur automatischen Durchsetzung unternehmensweiter Sicherheitsvorschriften entwickelt und findet vor allem in staatlichen Einrichtungen Anwendung. Multilevel Security (MLS) ist, aufbauend auf einem Modell von Bell und LaPadula ([Bell/LaPadula 1975]), die am weitesten verbreitete Umsetzung von MAC. Zugriffsrechte im MLS-System basieren auf so genannten Sicherheitsklassen, die sowohl Subjekten als auch zu schützenden Objekten zugewiesen werden. Ein Zugriff auf ein Objekt ist nur dann möglich, wenn das Subjekt mindestens derselben oder einer höheren Sicherheitsklasse wie das Objekt zugeordnet ist. Dieser Ansatz wird aufgrund seiner geringen Verbreitung im Folgenden nicht näher betrachtet.

Der *Discretionary Access Control (DAC)*-Ansatz basiert auf Subjekten und zu schützenden Objekten, wobei Zugriffsrechte explizit zugewiesen werden. Zugriffsrechte können logisch als Tupel – Subjekt, geschütztes Objekt, Aktion, Delegationsoption – abgebildet werden. Nur die ersten drei Komponenten sind notwendig, die verbleibende ist optional. Mithilfe eines solchen Tupels kann ausgedrückt werden, dass es einem Subjekt erlaubt oder verboten ist, die Aktion auf dem geschützten Objekt auszuführen. Ist die Delegationsoption gesetzt, kann das Subjekt das Zugriffsrecht an ein anderes Subjekt weitergeben.

Abhängig davon, welche Zugriffsrechte vergeben werden können, lassen sich drei Ausprägungen von DAC-Systemen unterscheiden:

- *Positive Systeme*: Es können nur Erlaubnisse definiert werden.
- *Negative Systeme*: Es können nur Verbote definiert werden.
- *Gemischte Systeme*: Es können sowohl Erlaubnisse als auch Verbote definiert werden. Zusätzlich ist eine Vorschrift notwendig, um Konflikte zwischen beiden Systemen aufzulösen.

In der Regel ist die Basis der Autorisierung nicht vollständig, so dass es nicht für alle Subjekte, Aktionen und geschützten Objekte ein explizit definiertes Zugriffsrecht gibt. Um ein geschlossenes System zu erhalten, muss eine so genannte Abgeschlossenheitsannahme (Closure Assumption) definiert werden:

- *Geschlossene-Welt-Annahme*: Eine Aktion ist so lange verboten, bis eine Erlaubnis definiert ist.
- *Offene-Welt-Annahme*: Eine Aktion ist so lange erlaubt, bis ein Verbot definiert ist.

Aspekte der Heterogenität und Autonomie

Ausgehend von der Taxonomie in Abschnitt 3.3.1.3 sind lose und eng gekoppelte FDDBS zu unterscheiden. Lose gekoppelte FDDBS weisen gegenüber üblichen, einen entfernten Zugriff zulassenden DBS keine wesentlichen Unterschiede auf. Daher werden im Weiteren nur eng gekoppelte FDDBS betrachtet. Jonscher und Dittrich ([Jonscher/Dittrich 1994]) stellen für alle im Folgenden dargelegten Konzepte fest, dass die anzuwendenden Richtlinien außerhalb von Computersystemen vereinbart werden müssen, anschließend jedoch von zu entwickelnden Datenbanktechnologien entsprechend durchzusetzen sind. Dies verdeutlicht abermals, dass bei der Betrachtung grundsätzlich verschiedene Dimensionen, z. B. auch eine organisatorische Dimension, zu berücksichtigen sind (vgl. auch Abschnitt 3.3).

Heterogenität in Sicherheitsaspekten findet sich neben der möglicherweise unterschiedlicher Granularität der geschützten Objekte und zu autorisierenden Subjekte vor allem in unterschiedlichen lokalen Sicherheitsvorschriften der an einem FDDBS beteiligten CDDBS und der Föderation selbst. Neben der grundsätzlichen Möglichkeit, dass sowohl MAC- als auch DAC-Systeme an einem FDDBS teilnehmen, besteht selbst bei der ausschließlichen Verwendung von DAC-Systemen eine große Anzahl von Kombinationsmöglichkeiten zwischen CDDBS, die zu Konflikten führen können. Es ergeben sich Kombinationen aus positiven, negativen und gemischten Systemen in Verbindung mit einer Geschlossenen- oder Offenen-Welt-Annahme (vgl. Zugriffskontrollkonzepte in diesem Kapitel).

Zur Lösung der aus der Heterogenität der verschiedenen DAC-Systeme entstehenden Probleme ist die Vereinbarung von Konfliktauflösungsvorschriften notwendig. In Abhängigkeit von der erforderlichen Sicherheit können zwei Basisvorschriften unterschieden werden, die entsprechend anzupassen sind:

- Anwendung der restriktivsten Rechte (beispielsweise Geschlossene- überschreibt Offene-Welt-Annahme): garantiert maximale Sicherheit einhergehend mit einer potenziellen Einschränkung der Flexibilität.
- Anwendung der weitreichendsten Rechte (beispielsweise Offene- überschreibt Geschlossene-Welt-Annahme): garantiert maximale Flexibilität einhergehend mit einer Einschränkung der Sicherheit.

Zur Analyse der Implikationen von *Autonomie* in Sicherheitsfragen ist der Zugriff auf Daten der Föderation auf zwei verschiedenen Ebenen zu betrachten: auf Ebene der Föderation selbst, an welche die Benutzer Anfragen bzgl. der Daten der Föderation stellen, und auf der lokalen Ebene der CDBS, an welche die Föderation im Rahmen der Anfrage der Benutzer selbst Anfragen stellt. Di Vimercati und Samarati ([di Vimercati/Samarati 1996], S. 88 f.) schlagen als Ansätze zur erstmaligen Autorisierung eines Benutzers folgende Optionen vor:

- Der Benutzer besitzt selbst ein Benutzerkonto bei der Föderation und meldet sich an dieser an. Weitere Zugriffe können dann mit der Identität erfolgen, mit der sich der Benutzer an der Föderation angemeldet hat.
- Der Zugriff auf die Föderation selbst ist nicht direkt beschränkt, Zugriffsrechte auf die Daten der Föderation werden über die Identität des Benutzers auf dem System, von dem die Verbindung ausgeht, etabliert.
- Eine Autorisierung erfolgt erstmalig auf Ebene der CDBS. Dadurch werden aber ggf. bereits Informationen über die Föderation, z. B. die Definition des Schemas, offenbart, die nicht zugänglich sein dürfen.

Die Veröffentlichungen von Jajodia und Wijesekera ([Jajodia/Wijesekera 2001]), Davie und Botha ([Davie/Botha 2001]), di Vimercati und Samarati ([di Vimercati/Samarati 1996]) und Jonscher und Dittrich ([Jonscher/Dittrich 1994]) berücksichtigen nur den ersten Ansatz. Ausgehend von der Fragestellung, wie beim Zugriff auf die CDBS mit der Authentifizierung und Autorisierung des Benutzers, die beide auf Ebene der Föderation erfolgten, verfahren wird, werden drei Stufen der in Abschnitt 3.3.1.2 genannten Autorisierungsautonomie entwickelt.

Das Konzept der Autorisierungsautonomie umfasst dabei sowohl die Authentifizierung als auch die Autorisierung.

Bei *voller Autorisierungsautonomie* muss jedes Subjekt bei jedem CDBS, auf das es über die Föderation implizit zugreift, ein eigenes Benutzerkonto besitzen und von diesem selbst authentifiziert werden. Lokale Zugriffsentscheidungen werden immer mit der lokalen Subjektidentität getroffen.

Obwohl die Autorisierungsautonomie der CDBS maximal ist und auch eine hohe Sicherheit gewährleistet, wird dem Subjekt die Föderation dadurch offenbart, dass es sich mehrfach authentifizieren muss. Dies schränkt zugleich die angestrebte Transparenz ein. Darüber hinaus ist dieser Ansatz mit großem administrativem Aufwand verbunden und zugleich durch erhöhte Fehleranfälligkeit gekennzeichnet.

Bei *mittlerer Autorisierungsautonomie* authentifiziert sich das Subjekt ausschließlich gegenüber der Föderation. Beim Zugriff auf die CDBS authentifiziert sich ausschließlich die Föderation gegenüber den CDBS und übermittelt dabei zusammen mit der Anfrage die Identität des Subjekts. Die Identität kann entweder der Identität des Subjekts bezogen auf die Föderation oder bezogen auf das Ursprungssystem des Zugriffs auf die Föderation entsprechen. Da die Identität von den CDBS nicht selbst überprüft wird, besteht die Möglichkeit, die Identität des Subjekts auf Ebene der Föderation zu wechseln. Dieses Vorgehen wird als „subject switching“ bezeichnet und ist ebenfalls Gegenstand von Untersuchungen (vgl. z. B. [Yang/Wijesekera/Jajodia 2001]). Ob und in welcher Form das Wechseln der Identität des Subjekts erlaubt ist, muss durch Verhandlungen und Festlegung verbindlicher Richtlinien geregelt werden. Die übermittelte Identität wird von den CDBS für lokale Zugriffsentscheidungen verwendet.

Der Ansatz birgt die Problematik, dass auf Seiten der CDBS Zugriffsregeln bzgl. Subjekten definiert werden müssen, die i. d. R. nicht bekannt sind oder zumindest nicht verwaltet werden.

Bei *schwacher Autorisierungsautonomie* authentifiziert sich das Subjekt selbst ebenfalls nur gegenüber der Föderation und nur die Föderation authentifiziert sich ihrerseits gegenüber den CDBS. Bei der Übermittlung der Anfrage wird jedoch keine Benutzeridentität transferiert. Die Zugriffsentscheidungen der CDBS werden auf Basis der abstrakten (artificial) Identität der Föderation getroffen. Auf lokaler Ebene bestehen keine benutzerabhängigen Zugriffsbeschränkungen, diese können ausschließlich auf der Föderationsebene festgelegt werden.

Mittlere und schwache Autorisierungsautonomien erfordern einen steigenden Grad an Vertrauen zwischen den CDBS und der Föderation bzw. den an der Föderation beteiligten organisatorischen Entitäten bzgl. der Einhaltung der vereinbarten Vorschriften. Dieses Vertrauen muss außerhalb technischer Systeme aufgebaut werden.

Die Definition der drei Stufen der Autorisierungsautonomie berücksichtigt bei den meisten Autoren ausschließlich das Subjekt in Form eines Benutzers oder einer Applikation. Rollen oder Gruppen werden in der Regel nicht betrachtet, da deren Berücksichtigung weitere Fragestellungen aufwirft:

- Können Gruppen von der Föderation an die CDBS weitergegeben werden?
- Falls ja, müssen die Gruppen sowohl auf Föderations- als auch CDBS-Ebene definiert sein?
- Falls nein, inwiefern können die Rechte, die mit der Zugehörigkeit zu einer Gruppe verbunden sind, aufgelöst und in dieser Form weitergegeben werden?
- Sind Gruppen auf der Ebene der Föderation von denen auf der Ebene der CDBS zu unterscheiden, z. B. durch Qualifikation der Gruppen der Föderation unter Einbeziehung eines Föderationsbezeichners?
- Wie können Probleme mit Homonymen und Synonymen bei der Verwendung von Gruppennamen sowohl auf der Ebene der Föderation als auch auf der Ebene der CDBS gelöst werden?

Diese Fragen werden weithin noch nicht behandelt, eine Ausnahme stellen Jajodia und Wikesekera ([Jajodia/Wijesekera 2001]) dar, die entsprechende Fragestellungen und teilweise Lösungsansätze thematisieren.

Innerhalb des Verbunds von zu einem FDBS zusammengeschlossenen CDBS kann jede CDBS im Rahmen ihrer Autorisierungsautonomie selbst festlegen, welche Form der Authentifizierung und Autorisierung zum Zugriff auf sie notwendig ist. Es ist die Aufgabe des übergeordneten FDBMS, die unterschiedlichen Anforderungen der CDBS zu verwalten und bei der Kommunikation mit ihnen entsprechend zu berücksichtigen.

3.3.2 Föderierte Informationssysteme

Busse et al. ([Busse et al. 1999]) beschreiben föderierte Informationssysteme (FIS) als logische Weiterentwicklung der Forschungsrichtung föderierter Datenbanksysteme. Hauptcharakteristikum des neuen Begriffs ist die Verallgemeinerung der möglichen Komponenten einer Föderation um Nicht-Datenbanken. Die Ausführungen von Conrad et al. gehen in dieselbe Richtung:

“In contrast to the classical notions of ‘federated database systems’ and ‘multi-database systems’, the term ‘federated information system’ intends to include not only structured information sources but also semi-structured and even unstructured information. These inclusions often have implications on the three dimensions of autonomy, heterogeneity, and distribution.” ([Conrad et al. 1999])

Als ein Beispiel für die Auswirkungen kann die Schemaintegration genannt werden, die bei FDBS zur Berücksichtigung der Dimensionen Heterogenität und Autonomie eingesetzt wird (vgl. Abschnitt 3.3.1.5). Sie ist weithin ungeeignet, wenn Daten nicht mehr ausschließlich durch Schemata beschrieben werden können, wie es bei semi- und unstrukturierten Datenquellen häufig der Fall ist.

Generell haben FIS in allen Dimensionen hohe Freiheitsgrade; Conrad et al. ([Conrad et al. 1999]) zufolge erscheinen praktikable Lösungen zur Föderierung nur möglich, wenn in der ein oder anderen Dimension einschneidende Beschränkungen hingenommen werden. Etablierte bereichsspezifische Standards und Ontologien könnten die Interoperabilität verbessern (vgl. [Gaedke/Turowski 1999], [Hasselbring 2000]). Die Autoren stellen aber fest, dass Standards, die detailliert genug wären, um praktische Interoperabilität zu leisten, nicht oder nur in unzureichendem Maße existieren.

Busse et al. schlagen für FIS eine Drei-Ebenen-Architektur (vgl. Abbildung 3-10) vor, bei der die Benutzer und Applikationen durch eine so genannte „Föderationsschicht“ auf die heterogenen Datenquellen zugreifen. Die Föderationsschicht erlaubt den einheitlichen Zugriff auf die in der Regel durch so genannte „Wrapper“ angebundenen Datenquellen.

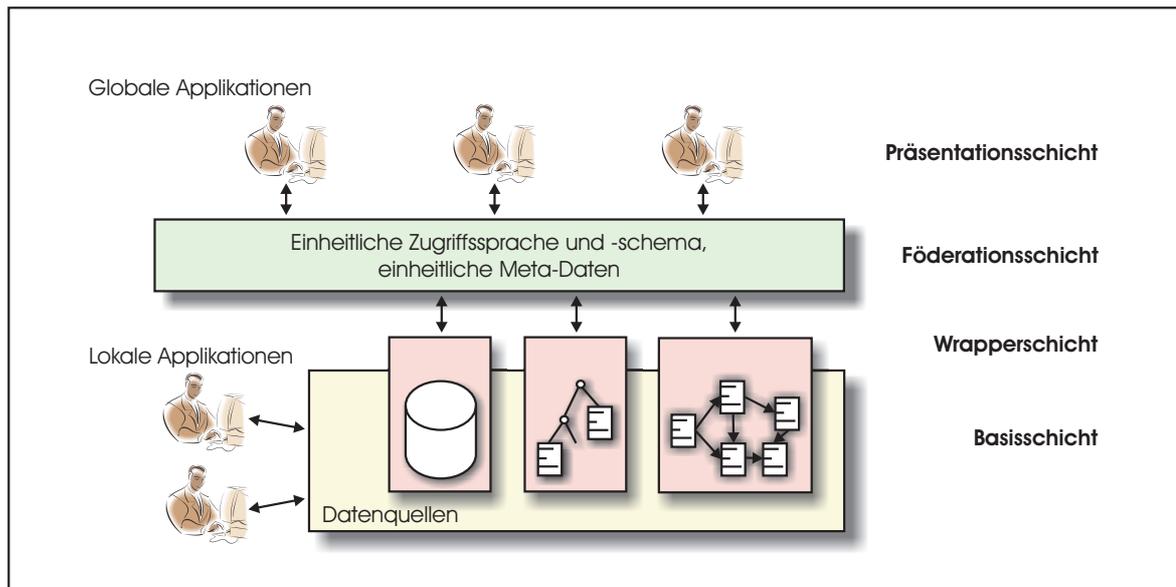


Abbildung 3-10: Drei-Ebenen-Architektur eines FIS
(vgl. [Busse et al. 1999], S. 7)

Ausgehend von dem Konzept der „Mediatoren“, das in dieser Form erstmals von Wiederhold ([Wiederhold 1992]) eingeführt wurde – Mediatoren können als Software-Komponenten bezeichnet werden, die zwischen den Benutzern und den physikalischen Datenquellen vermittelnd wirken –, entwickeln Busse et al. ([Busse et al. 1999]) das Modell der „mediator-based information systems“ (MBIS). MBIS definieren sie folgendermaßen:

“A mediator-based information system is a system that offers a homogeneous, virtual and read-only access mechanism to a dynamically changing collection of heterogeneous, autonomous and distributed information sources. [...] MBIS are typically lightweight systems that are developed top-down, addressing all types of heterogeneity. [...] The main components of a MBIS are wrappers, which encapsulate sources and remove technical and data model heterogeneity, and mediators, which resolve logical heterogeneity.” ([Busse et al. 1999], S. 32)

Deutlich wird an dieser Definition, dass hier bewusst die Einschränkung auf den ausschließlich lesenden Zugriff erfolgt, um mit der gegenüber FDBS gewachsenen Heterogenität umgehen zu können. Abbildung 3-11 zeigt die aus der allgemeinen Architektur eines FIS (vgl. Abbildung 3-10) abgeleitete Architektur eines MBIS.

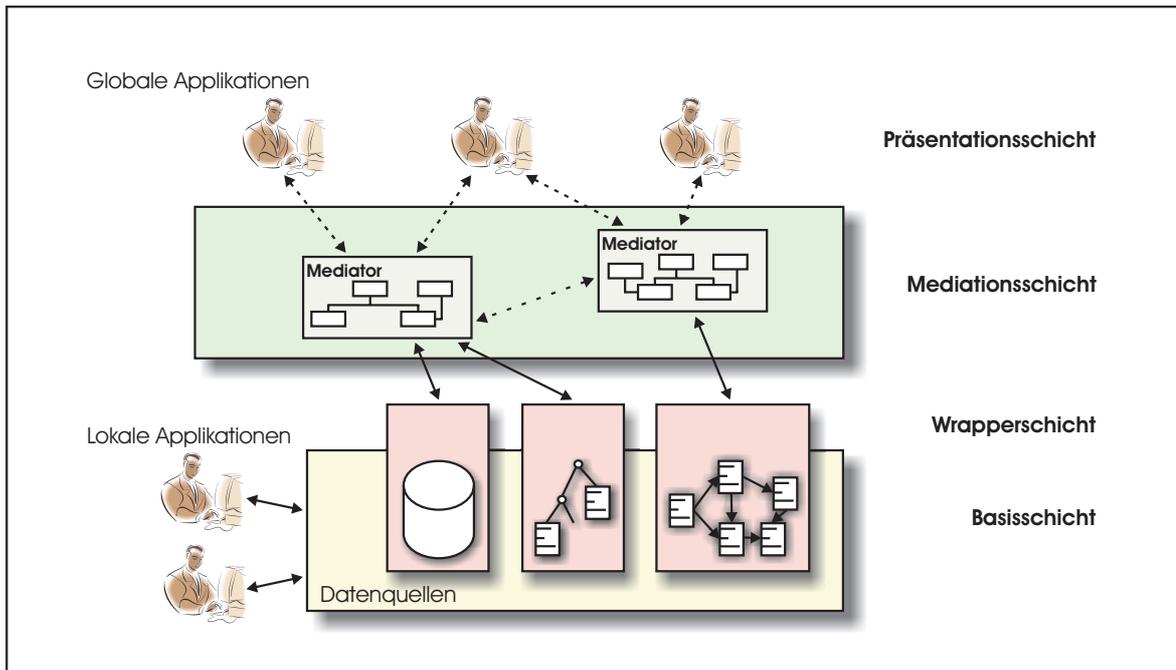


Abbildung 3-11: Architektur eines Mediator-basierten Informationssystems
(vgl. [Busse et al. 1999], S. 21)

In Anlehnung an die Klassifizierung föderierter Datenbanken (vgl. Abschnitt 3.3.1.3) entwickeln Busse et al. einen Kriterienkatalog von zehn Kriterien, um darauf aufbauend eine vorläufige Klassifikation föderierter Informationssysteme vorzuschlagen, die in Abbildung 3-12 wiedergegeben ist.

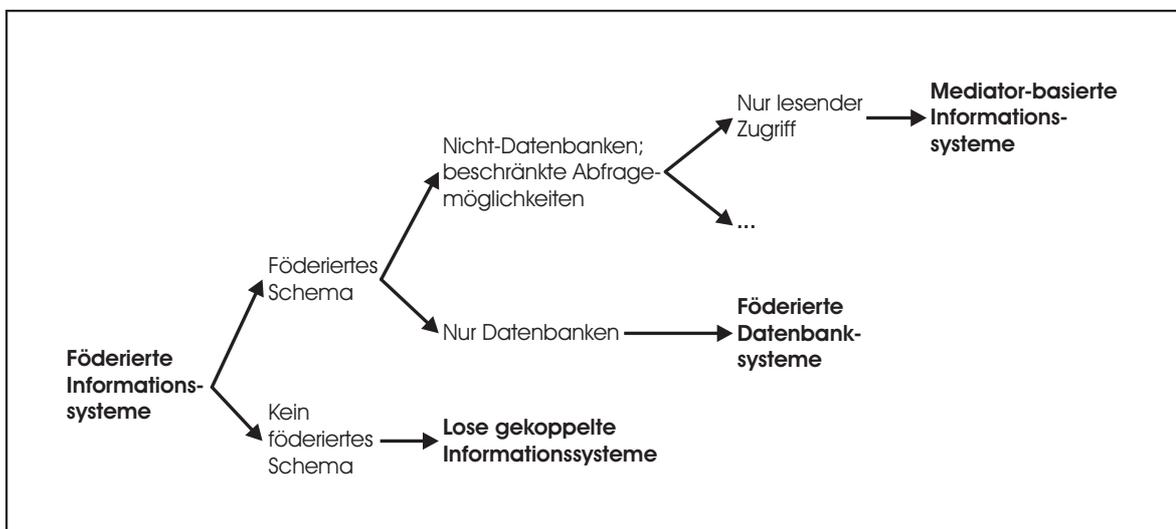


Abbildung 3-12: Klassifikation föderierter Informationssysteme
(vgl. [Busse et al. 1999], S. 18)

Die Klassifizierung ordnet die FDBS als eine Form der FIS ein. Die stilisierte Auslassung in der Abbildung legt gleichzeitig nahe, dass umfangreiche Forschungen notwendig sind, um die Erkenntnisse über die Schaffung von föderierten Informationssystemen voranzubringen. Dies wird beispielsweise auch durch die sehr unterschiedlichen Schwerpunkte der Beiträge für die

vergangenen Workshops zu „Engineering Federated Information Systems“ (EFIS) unterstrichen (vgl. z. B. [Conrad et al. 1999], [Hasselbring et al. 2000], [Conrad et al. 2002]).

3.3.3 Föderierte Benutzeridentitäten

Der Informationstechnologie-basierten Identität eines Benutzers oder Unternehmens kommt bei jeder Form der Transaktion eine zentrale Bedeutung zu. Sie stellt eine kritische Komponente innerhalb eines Unternehmens dar, die mit praktisch jedem Geschäftsprozess verwoben ist. Die Identität wird zur Zugriffssteuerung herangezogen, ist beispielsweise Grundlage von Customer Relationship Management (CRM)-Systemen und zur Etablierung und Aufrechterhaltung der Geschäftsbeziehung mit Kunden, Partnern und Zulieferern unabdingbar (vgl. [Liberty Alliance Project 2003a], S. 2).

Die Verwaltung von Identitäten ist mit zunehmender innerbetrieblicher Dezentralisierung und Vernetzung von Rechnerinfrastrukturen und Anwendungen ein immer dringlicheres Problem geworden. Diesem wurde innerhalb von Unternehmen teilweise erfolgreich durch die Etablierung zentraler Verzeichnisdienste mit standardisierten Schnittstellen und Zugriffsprotokollen (z. B. Lightweight Directory Access Protocol, LDAP) begegnet. Mit der Verbreitung des Internets innerhalb der letzten Dekade hat sich das Problem von einem innerbetrieblichen jedoch zusehends zu einem überbetrieblichen entwickelt. Die kooperierenden Systeme, Dienste und Anwendungen sind nicht mehr auf einen abgeschlossenen, von einer Organisation verwalteten Bereich beschränkt. Zur Nutzung der über Unternehmensgrenzen hinweg verteilten Dienste ist in der Regel eine Anmeldung an diesen notwendig. Jede Registrierung für einen Dienst erzeugt eine neue, von den anderen Identitäten unabhängige Identität.

Dieses ineffektive Identitätsmanagement führt nicht nur zu erhöhten Kosten, sondern stellt gleichzeitig eine Barriere für die Etablierung vertrauenswürdiger elektronischer Geschäftsbeziehungen dar. Mit der zunehmenden Virtualisierung der Geschäftsprozesse (vgl. auch Abschnitt 2.1.1) geht zusätzlich die Tendenz einher, die internen Systeme gezielt für die Partner zu öffnen, ohne dabei die Sicherheit und Skalierbarkeit zu beeinträchtigen (vgl. [Liberty Alliance Project 2003a], S. 3).

Aufgrund der zentralen Bedeutung des Managements und der Föderation von Identitäten wurden zum Zweck der Erarbeitung von Konzepten und zur Etablierung von Standards ausschließlich für diesen Bereich zwei Initiativen gegründet. Unter der Führung von IBM, Microsoft und der Mitwirkung von VeriSign, BEA und RSA Security wurde Mitte 2003 die

Spezifikation für die Web Service Federation Language (WS-Federation) vorgelegt (vgl. [Bajaj et al. 2003], [Della-Libera et al. 2003]). Das Liberty Alliance Project wurde 2001 gegründet und repräsentiert als offene Organisation internationale Vertreter eines breiten Spektrums an Industrien (vgl. [Liberty Alliance Project 2001]).

Durch die frühere Formierung des Liberty Alliance Projects ist die Entwicklung von Konzepten und deren Detaillierungsgrad weiter fortgeschritten, so dass sich die folgenden Ausführungen auf die Ansätze dieser Gruppe beschränken. In ähnlicher Form finden sich diese aber auch in der WS-Federation-Spezifikation wieder.

Grundlegender Gedanke ist die Schaffung einer föderierten Netzwerkidentität (Federated Network Identity) (vgl. [Wason et al. 2003], S. 4). Diese erlaubt es den Benutzern, persönliche Informationen verschiedener Identitäten miteinander zu verbinden, ohne die Informationen in einem zentralen System vorhalten zu müssen. Die Benutzer behalten dabei die vollständige Kontrolle darüber, wann und wie ihre Identitäten bzw. deren Informationen von verschiedenen Diensteanbietern gemeinsam genutzt bzw. zwischen diesen ausgetauscht werden. Der Begriff Föderierung umschreibt in diesem Zusammenhang die Technologien, die notwendig sind, um die Eigenschaften einer Identität zwischen voneinander unabhängigen Systemen transferierbar zu machen. Daraus lässt sich im Umkehrschluss ableiten, dass es sich bei einer föderierten Identität um eine transferierbare Identität handelt (vgl. [Liberty Alliance Project 2003a] und [Liberty Alliance Project 2003b]).

Um die Vision einer föderierten Identität in die Realität umzusetzen, ist es notwendig, zwischen allen beteiligten Parteien Vertrauen zu etablieren. Die Liberty Alliance sieht hierzu auf Seiten der Serviceanbieter die Schaffung so genannter „circles of trust“ vor. Zum Aufbau dieser circles of trust werden im Wesentlichen drei Modelle unterschieden (vgl. [Linn et al. 2003]):

- *Direktes, paarweise bestehendes Vertrauen* wird durch wechselseitige (rechtliche) Vereinbarungen und Verträge zwischen allen Beteiligten aufgebaut. Obwohl es die stärkste Form des Vertrauens der drei Modelle bietet, ist es nur begrenzt auf größere Gruppen skalierbar.
- *Gemakeltes Vertrauen* basiert nicht auf dem paarweise bestehendem Vertrauen zwischen allen Beteiligten, sondern darauf, dass es immer mindestens eine Instanz gibt, die als Intermediär dienen kann, um ein indirektes Vertrauen zwischen den Beteiligten herzustellen. Abhängig von den getroffenen Vereinbarungen zwischen dem Intermediär und dem Serviceanbieter sind zwei Fälle zu unterscheiden:

1. Die Vereinbarung mit dem Intermediär sieht explizit vor, dass die direkte Vereinbarung des Intermediärs mit dem Servicenachfrager transitiv genutzt werden kann, um die Vertrauensbeziehung zwischen Servicenachfrager und -anbieter zu begründen.

Durch die Forderung nach einer direkten Vereinbarung des Intermediärs mit dem Servicenachfrager werden die Stationen zwischen Servicenachfrager und -anbieter auf den Intermediär selbst beschränkt.

2. Die Vereinbarung mit dem Intermediär sieht vor, dass eine Vertrauensbeziehung zwischen dem Servicenachfrager und dem Intermediär transitiv auf den Serviceanbieter übertragen werden kann, ohne den Servicenachfrager vorab zu benennen. Die Vertrauensbeziehung zwischen Intermediär und Servicenachfrager kann selbst auf indirektem Vertrauen beruhen.

Dieses Vorgehen erfordert ein größeres Vertrauen des Serviceanbieters in den Intermediär, erlaubt dafür aber auch eine flexiblere, dynamischere Etablierung von Vertrauen mit einer größeren Anzahl an Beteiligten.

- *Gemeinschaftsvertrauen* setzt keine direkten oder indirekten Vereinbarungen zwischen den Beteiligten voraus. Grundlage ist vielmehr, dass alle Beteiligten nachweislich Mitglied in einer Gemeinschaft sind. Dadurch, dass der Gemeinschaft Vertrauen entgegengebracht wird, werden auch die Mitglieder der Gemeinschaft als vertrauenswürdig qualifiziert.

Der Überblick bzgl. der Zielsetzung, die das Liberty Alliance Project verfolgt, und die Darstellung der wesentlichen Aspekte der Modelle zur Bildung der circles of trust stellen nur einen kleinen Teil der entwickelten Infrastruktur und Definitionen dar. Für weitere Informationen sei neben den bereits genannten Quellen des Weiteren z. B. auf das Liberty-Alliance Project ([Liberty Alliance Project 2003c]), Landau et al. ([Landau et al. 2003]) und Watson et al. ([Wason et al. 2003]) verwiesen.

3.4 Zusammenfassung

Die Zielsetzung dieses Kapitels war die Schaffung eines grundlegenden Verständnisses für die wesentlichen Aspekte von Föderationen in verschiedenen Gebieten, vornehmlich im Bereich der IT. Die Ausführungen basieren im Wesentlichen auf einem intensiven Literaturstudium, auf dessen Grundlage die Extraktion der für den eigentlichen Forschungsgegenstand

relevanten Aspekte erfolgte. Diese sind in den folgenden Kapiteln für die Auseinandersetzung mit dem konkreten Thema der Föderation von Portalen heranzuziehen.

Verteiltheit, Heterogenität und Autonomie wurden als die Kerndimensionen identifiziert, die auf der einen Seite die Abgrenzung verschiedener Arten von Systemen zulassen und auf der anderen Seite unmittelbar Einfluss auf die Problemstellungen, Konzepte und Lösungsalternativen haben. Zentrale Aspekte wie Identitätsmanagement und Sicherheitsfragen wurden in unterschiedlichem Zusammenhang ebenso angesprochen wie Fragen der Schemaintegration und Replikation.

4 Konzeption einer Architektur zur Kopplung von Portalen

Aufbauend auf der dargestellten grundsätzlichen Problemstellung der vorliegenden Arbeit – der Förderierung von Portalen (Kapitel 2) und den damit verbundenen Rahmenbedingungen, sowie den spezifischen Fragestellungen, Konzepten und Lösungsansätzen für Förderierungen in anderen Forschungsbereichen, speziell der IT (Kapitel 3) – ist im Folgenden als Kern der Arbeit eine Lösungskonzeption für die Förderierung von Portalen zu erarbeiten. Dieses umfassende Vorhaben lässt sich in sechs aufeinander aufbauende Teilbereiche strukturieren: Als Basis erfolgt die Wahl der anzuwendenden Entwurfsmethode (Abschnitt 4.1). Die Identifikation der Basiskomponenten und -dienste, die verschiedene am Markt verfügbare Portale anbieten (Abschnitt 4.2), ermöglicht die Operationalisierung des Vorhabens (Abschnitt 4.3). Darauf aufbauend können die Anforderungen formuliert werden (Abschnitt 4.4), die schlussendlich in einem Kommunikations- und Synchronisationsmodell (Abschnitt 4.5) sowie in einem Daten- und Funktionsmodell (Abschnitt 4.6) für eine spätere informationstechnische Realisierung (Kapitel 5) umgesetzt werden.

4.1 Entwurfsmethoden

Die Erstellung eines Konzepts zur Kopplung von Portalen ist an sich ein komplexes Vorhaben und erfordert ein strukturiertes, planvolles, die gegebenen Rahmenbedingungen berücksichtigendes Vorgehen. Entwurfsmethoden sind weitgehend abstrakte Vorgehensweisen, die durch die Bereitstellung verschiedener bereits mehrfach erprobter Vorgehensmodelle, Richtlinien und Methoden einen Rahmen vorgeben, in dem das Vorhaben (vor)strukturiert durchgeführt werden kann. Die folgende Darstellung dreier im Kontext der Arbeit relevanter Entwurfsmethoden bildet die Grundlage für die Auswahl der für die Erstellung eines Konzepts zur Kopplung von Portalen einzusetzenden Entwurfsmethode.

4.1.1 Bottom-Up-, Top-Down- und Jojo-Entwurf

Die am häufigsten verwendete Methode, um heterogene und verteilte Informationssysteme miteinander zu integrieren, ist der *Bottom-Up-Ansatz*. Bei diesem werden die Informationen, die in den jeweiligen Systemen vorhanden sind, im Hinblick auf Überlappungen analysiert; auf dieser Basis erfolgt anschließend die Integration. Eine Folge des Bottom-Up-Ansatzes ist, dass sich die Integration der Informationssysteme nicht an den Bedürfnissen der globalen auf

sie zugreifenden Anwendungen orientiert, sondern dass die Struktur des zusammengesetzten gemeinsamen Datenmodells allein von den Datenmodellen der ursprünglichen Informationssysteme bestimmt wird. Die in Abschnitt 3.3.1.5 vorgestellte Fünf-Ebenen-Schemaarchitektur föderierter Datenbanksysteme ist von ihrem Basisverständnis her als Bottom-Up-Ansatz zu identifizieren.

Hasselbring ([Hasselbring 2000]) zeigt auf, dass die durch Integration auf Basis von Schnittmengen entstehenden globalen Modelle häufig durch Probleme in der Nutz- und Wartbarkeit gekennzeichnet sind. Gegenüber anderen, im Folgenden vorzustellenden Ansätzen zeichnet den Bottom-Up-Ansatz aus, dass die Komponentensysteme nicht angepasst werden müssen und die Transformationen, Filterungen und Abbildungen (vgl. Abschnitt 3.3.1.5) in den meisten Fällen einfache Operationen darstellen.

Als alternative Entwurfsmethode steht der *Top-Down-Ansatz* zur Verfügung, der auf der Unterscheidung zwischen den Phasen „domain engineering“ und „application engineering“ basiert (vgl. [Hasselbring 2000], S. 120 ff.). Beim domain engineering werden wiederverwendbare, bereichsspezifische Modelle und Architekturen erstellt, die im Anschluss daran beim application engineering verwendet werden, um konkrete Systeme auf deren Basis zu entwerfen und umzusetzen. Beide Bestandteile sind komplementär zueinander. Hasselbring fasst dies folgendermaßen zusammen:

“The domain model characterizes the *problem space*, while the *reference architecture* addresses the *solution space* (design). The *reference requirements* within the domain model define the (generic) functional requirements for applications in a domain.” ([Hasselbring 2000], S. 121)

Abbildung 4-1 stellt das Vorgehen zur Erstellung einer gemeinsamen Sicht auf Basis des Top-Down-Ansatzes dar. Beginnend mit der Definition der gemeinsamen Sicht, die sich an den Erfordernissen der globalen Anwendungen und relevanter bereichsspezifischer Standards orientiert, entstehen durch Filterung der globalen Sicht die Exportsichten. Diese werden durch Abbildung und Transformation in die Komponenten- und die lokalen Sichten überführt.

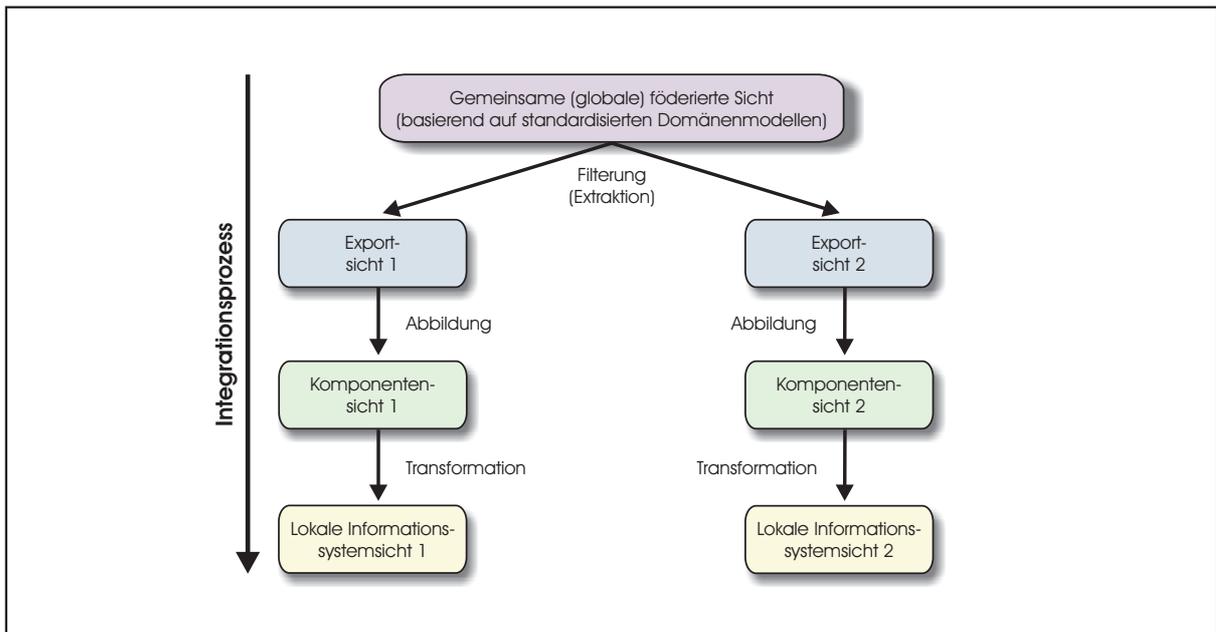


Abbildung 4-1: Top-Down-Ansatz zur Sichtenintegration
(in Anlehnung an [Hasselbring 2000], S. 123)

Zur Integration eines Komponentensystems ist es nicht mehr nötig, alle Systeme zu betrachten, um aus ihren lokalen Sichten eine globale Sicht zu bilden. Es reicht aus, nur noch die Komponentensicht des zu integrierenden Systems und die globale Sicht zu kennen und entsprechend ineinander zu überführen. Um diese Überführung möglichst einfach zu gestalten, sind auf der Ebene der globalen Sicht bereichsspezifische Standards zu verwenden. Damit ist es möglich, Komponentensysteme, die eine weitgehende Übereinstimmung mit den bereichsspezifischen Standards aufweisen, sehr schnell zu integrieren. Andere Komponentensysteme müssen einmalig dahingehend angepasst werden, dass sie Standardkonformität aufweisen, um dann ebenfalls schnell in die aktuellen, aber auch zukünftigen Integrationslösungen aufgenommen werden zu können.

Neben dem Bottom-Up- und dem Top-Down-Ansatz ist auch der häufig in der Praxis angewandte *Jojo-Ansatz* möglich. Dieser verbindet die Vorteile beider vorgenannter Ansätze miteinander. Als Bestandteil des Jojo-Ansatzes sorgt der Top-Down-Ansatz weiterhin für eine Entkopplung der Betrachtung der Komponentensysteme untereinander. Der Bottom-Up-Ansatz wird beim Jojo-Ansatz als eine Komponente zur initialen Entwicklung bereichsspezifischer Standards herangezogen. In späteren Phasen wird die Weiterentwicklung der globalen Sicht durch Überführung von Informationen aus den Komponentensichten, die ursprünglich nicht in der globalen Sicht enthalten waren, ermöglicht.

Ein wesentliches Element des Top-Down- und des Jojo-Ansatzes ist die Verfügbarkeit und Verwendung bereichsspezifischer Standards. Diese wurden hier ausschließlich unter dem

Aspekt des Entwurfs zur Integration heterogener und verteilter Informationssysteme betrachtet. Weitere Aspekte, die bei der Entwicklung und Verwendung von Standards für IT-Systeme relevant sind, finden sich z. B. bei der Delphi Group ([Delphi Group 2003]) und bei Buxmann ([Buxmann et al. 2000]).

4.1.2 Vorgehen zum Entwurf einer Architektur gekoppelter Portale

Auf Basis der im vorangegangenen Abschnitt dargestellten Entwurfsmethoden föderierter Sichten für heterogene und verteilte Informationssysteme ist vor dem eigentlichen Entwurf der Architektur gekoppelter Portale zu betrachten, welcher Ansatz Erfolg versprechend ist.

Im Bereich der Portale sind Standards bisher, wenn überhaupt, nur in Teilbereichen vorzufinden. Diese beschränken sich auf die im Herbst 2003 verabschiedeten Portlet-Standards Java Portlet API (vgl. [Java Community Process 2003]) und Web Service for Remote Portlets (WSRP) (vgl. [Thompson/Leue/Kropp 2003]).¹¹ Zusätzlich ist am Markt trotz mehrerer Konsolidierungswellen weiterhin eine große Anzahl an Anbietern von Portalsoftware vertreten, die teilweise gleiche Grundelemente in ihren Portalarchitekturen verwenden, aber im Detail betrachtet sehr verschieden und heterogen sind (vgl. Abschnitt 4.2). Aufgrund dieser Gegebenheiten ist es praktisch unmöglich, einen Bottom-Up-Ansatz zu verfolgen, bei dem ein gemeinsames globales Modell für die Kopplung von Portalen auf Basis der einzelnen Komponentenportale gebildet wird. Dieser Ansatz wäre bei weitem zu komplex und sein Ergebnis praktisch nicht nutzbar. Aufgrund der fehlenden bereichsspezifischen Standards ist ein Top-Down-Ansatz ebenfalls nicht unmittelbar sinnvoll anwendbar. Im Weiteren wird daher der Jojo-Ansatz angewendet.

Einerseits wird eine Analyse vorhandener Portalarchitekturen und ihrer wesentlichen Elemente vorgenommen. Das Ergebnis der Analyse (Bottom-Up) wird insofern abstrahiert, als gewisse Grundelemente und Grundfunktionen identifiziert werden, die im globalen Modell wiederzufinden sein müssen. Andererseits sind Anforderungen festzulegen, die von der Kopplung von Portalen als quasi globaler Anwendung gestellt werden (Top-Down).

Aus beiden Aspekten werden anschließend Vorschläge für Standards für den Bereich gekoppelter Portale entwickelt (domain engineering, vgl. Abschnitt 4.5 und 4.6), die über einen Top-Down-Ansatz zur Realisierung eines neuen bzw. zur Anpassung eines bestehenden Portalsystems angewendet werden können (application engineering, vgl. Kapitel 5).

¹¹ Zu einer Definition von Portlets vgl. Abschnitt 4.2.2.

4.2 Basiskomponenten und -dienste eines Portals

Um den Gegenstand der Kopplung von Portalen diskutieren zu können, ist es notwendig, die Basiskomponenten und Basisdienste eines Portals zu identifizieren. Hierzu wurden ausgewählte, am Markt verfügbare Portal-Frameworks (Apache Jetspeed 1 [Apache 2004], IBM WebSphere Portal [IBM 2004], Oracle Enterprise Portal [Oracle 2004], SAP Enterprise Portal [SAP 2004]) analysiert und basisgebende strukturelle Gemeinsamkeiten herausgearbeitet, die im Wesentlichen in Abbildung 4-2 dargestellt und in den folgenden Abschnitten diskutiert werden.

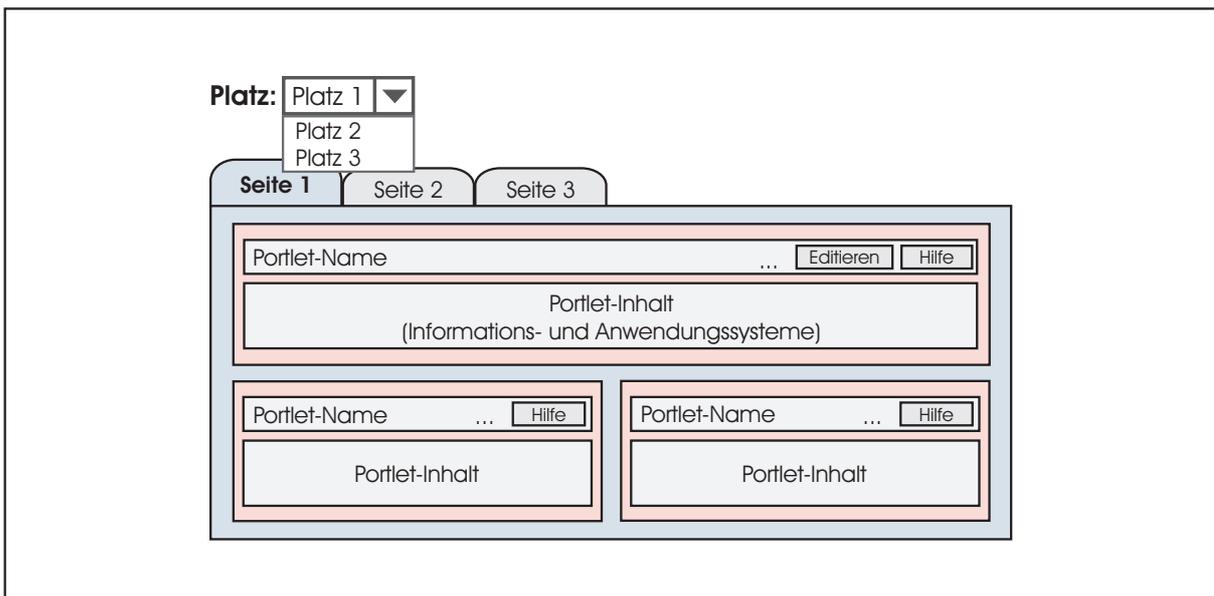


Abbildung 4-2: Gemeinsame Strukturelemente verschiedener Portalimplementierungen
(in Anlehnung an [Java Community Process 2003], S. 19)

4.2.1 Informations- und Anwendungssysteme

Informations- und Anwendungssysteme sind die funktionalen Bausteine der betrieblichen Informationstechnologie. In und mit ihnen werden die unternehmensrelevanten Daten, Informationen und Prozesse gespeichert, verwaltet und verarbeitet. Zu den wichtigsten Systemkategorien zählen unter anderem Enterprise Resource Planning-, Produktionsplanungs- und Produktionssteuerungs-, Groupware-Systeme sowie Desktop-Büro-Anwendungen.

Ein wesentliches bereits mehrfach dargestelltes Ziel, das Portale verfolgen, ist die Integration dieser verschiedenen, zumeist voneinander unabhängigen Systeme in einer personalisierten und konsistenten Oberfläche. Ein Portal stellt daher i. d. R. selbst keine eigenen Inhalte zur Verfügung, sondern ist oberhalb bereits bestehender Informations- und Anwendungssysteme

angesiedelt, deren Inhalte und Funktionen den Benutzern in einer aggregierten Darstellung zugänglich gemacht werden.

4.2.2 Portlets

Portlets stellen die kleinste Informationseinheit eines Portals dar und sind das Bindeglied zwischen den Informations- und Anwendungssystemen auf der einen und den übrigen Portal-diensten auf der anderen Seite. Sie bieten den Benutzern des Portals den personalisierten Zugang zu Informationen, Applikationen, Prozessen und Personen, indem die anzuzeigenden Informationen aus den dahinter liegenden Informations- und Anwendungssystemen abgerufen und im Portlet dargestellt werden. Portlets können nicht nur zur ausschließlichen Anzeige von Informationen, sondern auch als Benutzerschnittstelle zur Durchführung komplexer Prozesse und Aufgaben eingesetzt werden. Aus welcher Quelle die im Portlet angezeigten Informationen und Anwendungen stammen, ist für den Benutzer sekundär, da Portlets eine aufgaben- und nicht anwendungszentrierte Arbeitsweise fördern.

Portlets tragen einen Namen, der hauptsächlich von den Administratoren im Rahmen der Auswahl eines geeigneten Portlets und von den Benutzern beim Wiederauffinden eines gesuchten Portlets als Hauptidentifikationsmerkmal herangezogen wird. Daher sollten die Namen der Portlets die von ihnen angebotenen Informationen und Funktionen möglichst zweckmäßig und präzise beschreiben. Die Informationen selbst werden in einem grafisch abgegrenzten Bereich dargestellt. Portlets können verschiedene Modi unterstützen wie Ansicht, Konfigurieren, Editieren, Hilfe etc., verschiedene Status wie Normal, Maximiert und Minimiert sowie unterschiedliche Ausgabegeräte wie Web-Browser, Mobiltelefon, Handheld etc. Abhängig von den unterstützten Modi und dem Quellsystem sowie der Basiskonfiguration der Portlets beinhalten Portlets durch die Veränderbarkeit von Parametern und Optionen eine inhärente Anpassungsfähigkeit (engl. customization) durch den Administrator und/oder den Benutzer. Jeder Benutzer erhält eine individuelle Sicht auf den Informationsbestand eines Informationssystems oder einer Anwendung. Ob ein Benutzer Zugriff auf ein Portlet hat und wenn ja welche Art des Zugriffs, wird durch Zuweisung von Rechten auf einzelne Portlets durch Administratoren gesteuert.

Der Begriff des Portlets beschreibt in Bezug auf die technische Realisierung zwei miteinander verbundene Entitäten. Auf der einen Seite existiert die Definition eines Portlets, die dessen Basiskonfiguration, z. B. die Zugriffsparameter auf ein Informationssystem, festlegt. Auf der anderen Seite können auf Basis dieser Definition entsprechend konkrete Instanzen eines Port-

lets erzeugt werden, die den Benutzern im spezifischen Kontext angezeigt werden. Die Verwendung des allgemeinen Begriffs Portlet hat sich durchgesetzt. Im Rahmen dieser Arbeit schließt der Begriff entweder beide Bedeutungsvarianten ein, oder aber die jeweilige Bedeutung wird explizit genannt bzw. geht aus dem Kontext hervor.

Aus der Vielzahl der in der Literatur genannten Definitionen für den Begriff des Portlets werden im Folgenden beispielhaft drei herausgestellt, die unterschiedliche Facetten betonen und die oben gemachten Ausführungen unterstützen:

- “Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems.” ([Java Community Process 2003], S. 13)
- “The term ‘portlet’ refers to a small portal application, usually depicted as a small box in the web page. Portlets are reusable components that provide access to applications, web-based content, and other resources.” ([IBM 2003], S. 7)
- “A portlet is a live area of HTML or XML/XSL which represents an information source in a standardized, consistent, and secure manner. Portlets summarize, promote, or provide basic access to an information source for a group of users who find business value in the information.” ([Oracle 2002])

4.2.3 Seiten

Einzelne Portlets können die Funktion eines Portals als Single Point of Access in ein umfassendes Informations- und Anwendungsangebot nicht erfüllen. In der Regel bilden sie ausschließlich den Zugang zu einem Informations- oder Anwendungssystem ab. Es ist daher erforderlich, Portlets innerhalb eines Portals gruppieren und zu neuen Informationseinheiten zusammenfassen zu können.

(Portal-)Seiten (engl. pages) lösen dieses Problem, indem sie als Behälter für die Aufnahme von Portlets fungieren. „The portal aggregates portlet windows into a complete document, the portal page.“ ([Java Community Process 2003], S. 19). Administratoren und/oder Benutzer des Portals platzieren, abhängig von ihren Rechten, auf ihnen die Portlets, die für die entsprechende Aufgabe, Rolle etc. benötigt werden. Die Seiten bringen die vormals voneinander unabhängigen Portlets und damit die Informations- und Anwendungssysteme in einen neuen Kontext.

Bei Seiten handelt es sich in den betrachteten Realisierungen um zweidimensionale Flächen, deren Aufteilung durch ein tabellenartiges Raster aus so genannten Zeilen- und Spalten-Containern bestimmt wird; eine Überlagerung von Portlets ist bei aktuellen Portalsystemen i. d. R. nicht möglich. Die Portlets lassen sich entsprechend dem Layout flexibel in die jeweiligen Spalten-Container einfügen; können jedoch nicht völlig frei auf der Seite positioniert werden.

Zur Festlegung des Layouts sind grundsätzlich drei Ansätze zu identifizieren:

- Auswahl aus einer Anzahl vordefinierter Layouts, z. B. ein-, zwei-, dreispaltig.
- Freie Festlegung des Layouts der Seite durch Hinzufügen, Löschen, Verschieben und Verändern von Zeilen- und Spalten-Containern.
- Eine Kombination aus beiden Ansätzen, bei der vordefinierte Standardlayouts direkt verwendet, diese flexibel angepasst oder eigene Layouts erstellt werden können.

Soweit dies aus den verfügbaren Informationen nachvollziehbar ist, stellt die Auswahl aus vordefinierten Standardlayouts zumeist eine Spezialisierung der freien Festlegung des Layouts dar. Umgekehrt lassen sich die Standardlayouts durch die Mechanismen zur Festlegung eines freien Layouts erzeugen. Die visuelle Darstellung der Navigation zwischen den Seiten ist portalspezifisch und kann auch innerhalb eines Portals variieren. Neben der in Abbildung 4-2 gezeigten Verwendung von „Karteireitern“ ist alternativ die Darstellung in einer Gliederung vorzufinden. Einige Portal-Frameworks bieten die Möglichkeit, dies auch flexibel an andere Anforderungen anzupassen.

4.2.4 Plätze

Plätze (engl. places) sind die höchste Aggregationsebene eines Portals. Sie dienen der weiteren Strukturierung und fassen zusammengehörige Seiten, z. B. basierend auf Projekten, Aufgaben oder Organisationseinheiten, zusammen. Diese Gruppierung von Portalseiten ist in praktisch allen Portal-Frameworks vorhanden, wird jedoch von den Anbietern unterschiedlich bezeichnet: „Design palettes“ (BEA WebLogic Portal), „template framework“ (Oracle’s 9iAS Portal), „page groups“ (Plumtree Portal), „theme editors“ (SAP’s Portal) und „places“ (WebSphere Portal) (vgl. [Drakos 2003]).

Ein weiterer wesentlicher Aspekt von Plätzen ist die Schaffung von Teamumgebungen, in denen lokale sowie verteilte oder virtuelle Teams an gemeinsamen Aufgaben und Projekten arbeiten können (vgl. Abschnitt 2.2.2.1). Dies wird dadurch ermöglicht, dass ein Platz nicht nur von einem Benutzer verwendet werden kann, sondern von mehreren und so ein „Shared Information Workspace“ entsteht. In den untersuchten Implementierungen lassen sich zwei grundsätzliche Arten von Plätzen identifizieren:

1. Persönliche Plätze (Private Places)
2. Gemeinsam genutzte Plätze (Shared Places)

- Zugriffsgeschützte, gemeinsam genutzte Plätze
- Offene, gemeinsam genutzte Plätze

Bei *persönlichen Plätzen* handelt es sich um spezielle Plätze, auf die jeweils nur ein Anwender Zugriff hat. Jedem Anwender stehen ein oder mehrere persönliche Plätze zur Verfügung. Diese dienen der Organisation der persönlichen Arbeitsumgebung unabhängig von speziellen Projekten oder ähnlichen Kriterien und können zusätzlich z. B. private Portlets wie die persönliche E-Mail, den persönlichen Kalender etc. enthalten, wohingegen in gemeinsam genutzten Plätzen i. d. R. Gruppen-/Projektkalender etc. eingesetzt werden. Die Seiten und Portlets sind dabei exklusiv dem jeweiligen Benutzer zugeordnet.

Gemeinsam genutzte Plätze bieten mehreren Benutzern Zugriff auf die in ihnen enthaltenen Seiten und Portlets. Sie lassen sich durch die Art der Zugriffskontrolle in *zugriffsgeschützt* und *offen* unterteilen. Offene, gemeinsam genutzte Plätze stehen allen Benutzern zur Verfügung, bei zugriffsgeschützten, gemeinsam genutzten Plätzen beschränkt hingegen eine Zugriffskontrollliste (ZKL) den Benutzerkreis auf berechnigte Benutzer.

Obwohl die Benutzer bei gemeinsam genutzten Plätzen grundsätzlich die gleichen Seiten und Portlets zur Verfügung haben, nutzen sie nur deren Meta-Struktur, also die Beschreibung der Seiten des Platzes und die in den Seiten enthaltenen Portlets, nicht aber deren konkreten Zustand gemeinsam. Die einzelnen Benutzer können so unabhängig voneinander mit dem Portal arbeiten, werden nicht durch andere Benutzer unbeabsichtigt beeinflusst, und auch unter Sicherheitsaspekten findet eine Trennung zwischen den Benutzern statt.

Dieser Aspekt soll für Portlets durch zwei Beispiele verdeutlicht werden: Die Navigation, das Layout und die Anordnung der Portlets ist für alle Benutzer grundsätzlich gleich. Würde der konkrete Zustand der Portlets aber ebenfalls gemeinsam genutzt werden, dann würde der durch einen Benutzer veranlasste Wechsel eines Portlets in den Editiermodus auch zum Wechsel des Portlet-Modus für alle anderen Benutzer führen. Gleiches würde für die Bearbeitung einer Transaktion in einem Portlet gelten. Hier könnten andere Benutzer die Transaktion eines Benutzers einsehen und beeinflussen. Ein Portlet könnte also selbst bei Ausklammerung von Sicherheitsgesichtspunkten nie gleichzeitig von mehreren Benutzern verwendet werden.

Deshalb wird gefordert, dass es für gemeinsam genutzte Plätze logisch-konzeptionell ein zentrales Referenzmodell gibt, das für alle Benutzer gleich ist. Von diesem können entsprechend benutzerspezifisch einzelne Instanzen der Portalelemente erzeugt werden, die für eine

Trennung der Benutzer sorgen. Die Ausführungen im weiteren Verlauf des Kapitels basieren auf dieser Forderung. Da es sich jedoch ausschließlich um ein logisch-konzeptionelles Referenzmodell handelt, wird keine Aussage darüber getroffen, wie konkrete Implementierungen dies tatsächlich umsetzen.

Die visuelle Darstellung der Navigation zwischen Plätzen ist analog zur Navigation zwischen Seiten portalspezifisch und kann ebenfalls innerhalb eines Portals variieren. Neben der in Abbildung 4-2 gezeigten Verwendung einer Auswahlliste ist alternativ die Darstellung als eine weitere Ebene von „Karteireitern“ oder als Elemente in einer Gliederung vorzufinden. Einige Portal-Frameworks bieten die Möglichkeit, dies flexibel an weitergehende Anforderungen anzupassen.

Abbildung 4-3 visualisiert den Zusammenhang zwischen den Strukturelementen Platz, Seite – mit Zeilen- und Spalten-Containern – und Portlets nochmals zusammenfassend in Form einer an die Syntax der Universal Modelling Language (UML) angelehnten Darstellung eines Klassendiagramms.

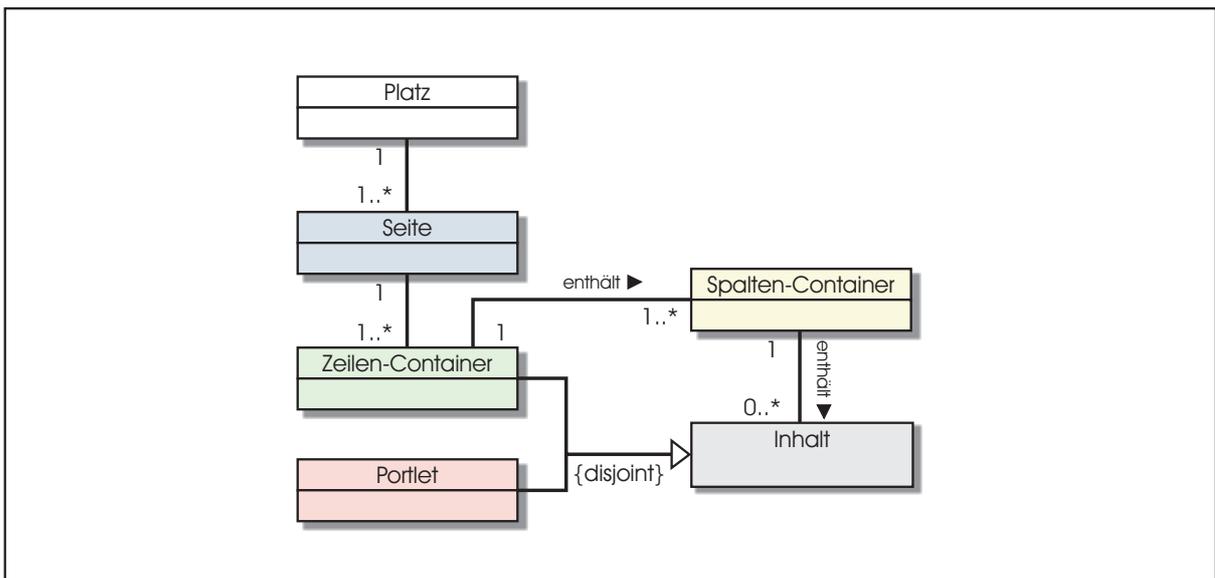


Abbildung 4-3: Strukturelle Darstellung der Beziehungen der Portalelemente

4.2.5 Weitere Dienste

Um den Nutzen von Portalen weiter zu erhöhen, sind neben den Kernelementen zur Strukturierung eines Portals zahlreiche weitere Dienste in die Portal-Frameworks integriert. Aufgrund der Vielzahl an möglichen und von verschiedenen Anbietern implementierten Dienste werden im Folgenden nur die *Personalisierung*, die *Suche* und die *Echtzeitkollaborations-*

funktionen näher betrachtet. Diese sind unter dem Gesichtspunkt der Kopplung interessant und in praktisch allen Portal-Frameworks anzutreffen.

Personalisierung

Die Hauptaufgabe der Personalisierung (engl. personalization) ist die automatisierte Bereitstellung von auf den Nutzer abgestimmten Informationen und Funktionen. Die Grundlage von Personalisierungsfunktionen sind Profile, die Informationen über Rollen, Aufgaben, Interessen, Nutzungsverhalten und weitere auswertbare Aspekte des Benutzers enthalten. Die Profilinformationen können mittels automatischer Verfahren gewonnen werden, z. B. aus der Analyse der Nutzung des Portals, aus der Position des Benutzers im Unternehmensorganigramm, aus vorhandenen persönlichen Daten der Personalabteilung oder auch aus durch den Benutzer selbst gepflegten Profildaten. Je genauer die Profilinformationen sind, desto besser können die im Portal präsentierten Inhalte auf den Benutzer abgestimmt werden.

Die Personalisierung kann im Wesentlichen auf der Ebene der Inhalte von Portlets oder auf der Ebene ganzer Seiten oder Plätze erfolgen. Zwei Beispiele verdeutlichen dies:

- In Abhängigkeit von der Region, der ein Benutzer zugeordnet ist, können ihm in einem Portlet direkt die Umsatzzahlen dieser Region angezeigt werden.
- Ist ein Mitarbeiter im Bereich Marketing tätig, dann werden ihm bereichsspezifisch relevante Plätze und Seiten zur Verfügung gestellt, aber er erhält keinen Zugriff auf Plätze und Seiten, die ausschließlich für Mitarbeiter aus anderen Bereichen vorgesehen sind.

Suche

Die Suche erlaubt es den Benutzern, auf die Inhalte eines Portals unabhängig von den durch die Struktur des Portals vorgegebenen Navigationspfaden zuzugreifen. Hierzu werden in das Portal integrierte oder externe Suchmaschinen eingesetzt, welche die Suche auf den im Portal vorhandenen Quellen durchführen und über spezielle Portlets angesprochen werden. Zu unterscheiden sind die indexbasierte und die brokerbasierte Suche:

- *Indexbasierte Suche*: Bei der indexbasierten Suche indiziert die Suchmaschine selbst alle zu durchsuchenden Quellen. Sie greift bei der Suche direkt auf den eigenen Index zu und bildet daraus ein vom Ausgabeformat her einheitliches Suchergebnis.
- *Brokerbasierte Suche*: Bei der brokerbasierten Suche handelt es sich um eine Meta-Suchmaschine. Diese erzeugt nicht selbst Indizes über die zu durchsuchenden Quellen, um diese bei der eigentlichen Suchanfrage zu durchsuchen. Stattdessen greift sie über standardisierte

oder proprietäre Schnittstellen auf vorhandene Suchmaschinen zu, die jeweils für die Indizierung einer oder mehrerer Informationsquellen verantwortlich sind. Aus den einzelnen Suchergebnissen bildet die Meta-Suchmaschine anschließend das konsolidierte Suchergebnis.

Echtzeitkollaborationsfunktionen

In den aktuellen Portalprodukten finden sich integrierte Funktionen von „Online Awareness“ und „Chat-Funktionen“ bis hin zu „e-Meetings“ mit Audio- und Videoübertragung, „Whiteboard“ und „Application Sharing“.

Diese Funktionalitäten sollen die Kommunikation und Zusammenarbeit zwischen den verteilten Mitarbeitern verbessern und gleichzeitig zur Kostensenkung beitragen, z. B. durch Reduktion der notwendigen Reisen. Zur Realisierung kommen auch hier entweder direkt in das Portal-Framework integrierte Serverfunktionen oder externe Produkte zum Einsatz. Die Integration erfolgt bisher ausschließlich über proprietäre, portal- bzw. produktspezifische Programmierung. Wird bereits ein anderes Produkt eingesetzt, das dieselben Aufgaben erfüllt, wäre es sinnvoll, dieses beibehalten zu können und die im Portal integrierte Implementierung zu ersetzen. Dies ist aufgrund fehlender Standards zur Einbindung in die Portale jedoch bisher – wenn überhaupt – nur mit großem Aufwand möglich.

4.2.6 Zusammenfassung

Die Analyse verschiedener Portal-Frameworks, die sich auf die von den Anbietern zur Verfügung gestellten Dokumentationen, das Studium vorhandener Programmcodes, ein teilweise durchgeführtes Reverse-Engineering¹² und die praktische Anwendung stützt, hat ein uneinheitliches Bild ergeben.

Die Basisstrukturen und -dienste sind in praktisch allen untersuchten Portal-Frameworks vorhanden, auch wenn diese teilweise mit unterschiedlichen Namen belegt werden. Im Detail weisen sie aber eklatante Unterschiede in der funktionalen Mächtigkeit und Realisierung auf. Einige Beispiele verdeutlichen dies nochmals zusammenfassend:

- Vererbungsbeziehungen zwischen Elementen (Plätzen, Seiten, Portlets) erlauben teilweise die Definition von Abhängigkeiten, die andernorts nicht abgebildet werden können.

¹² Dies beschränkte sich auf die Analyse der erzeugten HTML-Ausgabe, die Analyse des Aufbaus, der Verknüpfung und der Inhalte der Tabellen der verwendeten RDB und ähnliche Ansätze, die auch bei kommerziellen Produkten zulässig sind.

- Die Hierarchisierung von Seiten ist bei einigen Anbietern unbeschränkt, bei anderen ist sie dagegen auf eine Ebene und damit auf eine sequenzielle Abfolge von Seiten limitiert.
- Rechte können teilweise bis auf die Ebene einzelner Spalten einer Seite, in anderen Fällen aber nur auf die Ebene von Plätzen vergeben werden.
- Die Granularität der auf den einzelnen Elementen zu vergebenden Rechte ist sehr unterschiedlich. So wird z. B. teilweise zwischen verschiedenen Aufgaben der Benutzeradministration unterschieden, in anderen Fällen ist hierfür immer die Rolle des Administrators zuständig.

Die Analyse hat die in Abschnitt 4.1.2 gemachte Aussage bestätigt, dass, obwohl die Grundansätze eng miteinander verwandt sind, oberhalb der Ebene der Portlets bisher weder Standards noch Quasistandards verfügbar sind.

4.3 Operationalisierung gekoppelter Portale

Bislang wurde ausgehend von der in Abschnitt 2.2.2.3 gegebenen Problembeschreibung und den in der Literatur gemachten Vorschlägen die Föderierung von Portalen als Untersuchungsgegenstand der vorliegenden Arbeit identifiziert. Kapitel 3 hat einen Überblick darüber gegeben, was unter dem Begriff der Föderation speziell im IT-Bereich verstanden wird. Um Vorschläge für bereichsspezifische Standards machen und eine Architektur für gekoppelte Portale konzipieren zu können, ist es erforderlich, den Begriff der Kopplung bzw. Föderation von Portalen zu konkretisieren und für diesen Zweck zu operationalisieren.

4.3.1 Begriffsbestimmung

Unter dem Begriff gekoppelter Portale wird, angelehnt an die in Abschnitt 3.3 formulierte allgemeine Zielsetzung der Föderation in der Informationstechnologie, der koordinierte Austausch und die ggf. gemeinsame Nutzung von über Portalgrenzen hinweg verteilten Informationen und Anwendungen durch Einsatz von Portalen verstanden. Wesentliches Ziel ist die Wiederherstellung des durch die Proliferation – von sowohl internen als auch externen Portalen – verloren gegangenen Single Point of Access, also die Rückführung des Portalparadigmas auf sein Kernversprechen.

Verteilung, Heterogenität und Autonomie sind die im Verlauf des Kapitels darzustellenden Merkmale, welche die Ausprägung der Kopplung auch in Hinblick auf eine tatsächliche Föde-

ration sowohl aus betriebswirtschaftlicher wie aus informationstechnischer Sicht maßgeblich bestimmen.

4.3.2 Kontinuum der Portalkopplung

Ausgehend von den in Abschnitt 4.2 vorgestellten Basiskomponenten eines Portals wurde ein Kontinuum der Portalkopplung (vgl. Abbildung 4-4) entwickelt, dessen Ausprägungen geeignet sind, die verschiedenartige Zusammenarbeit innerhalb und zwischen Unternehmen zu unterstützen. Das Kontinuum lässt sich auch derart interpretieren, dass es sich um mögliche Stufen der fortschreitenden Umsetzung einer Portalkopplung handelt. Die ebenfalls in Abschnitt 4.2.5 thematisierten Portaldienste sind hierzu orthogonal zu verstehen.

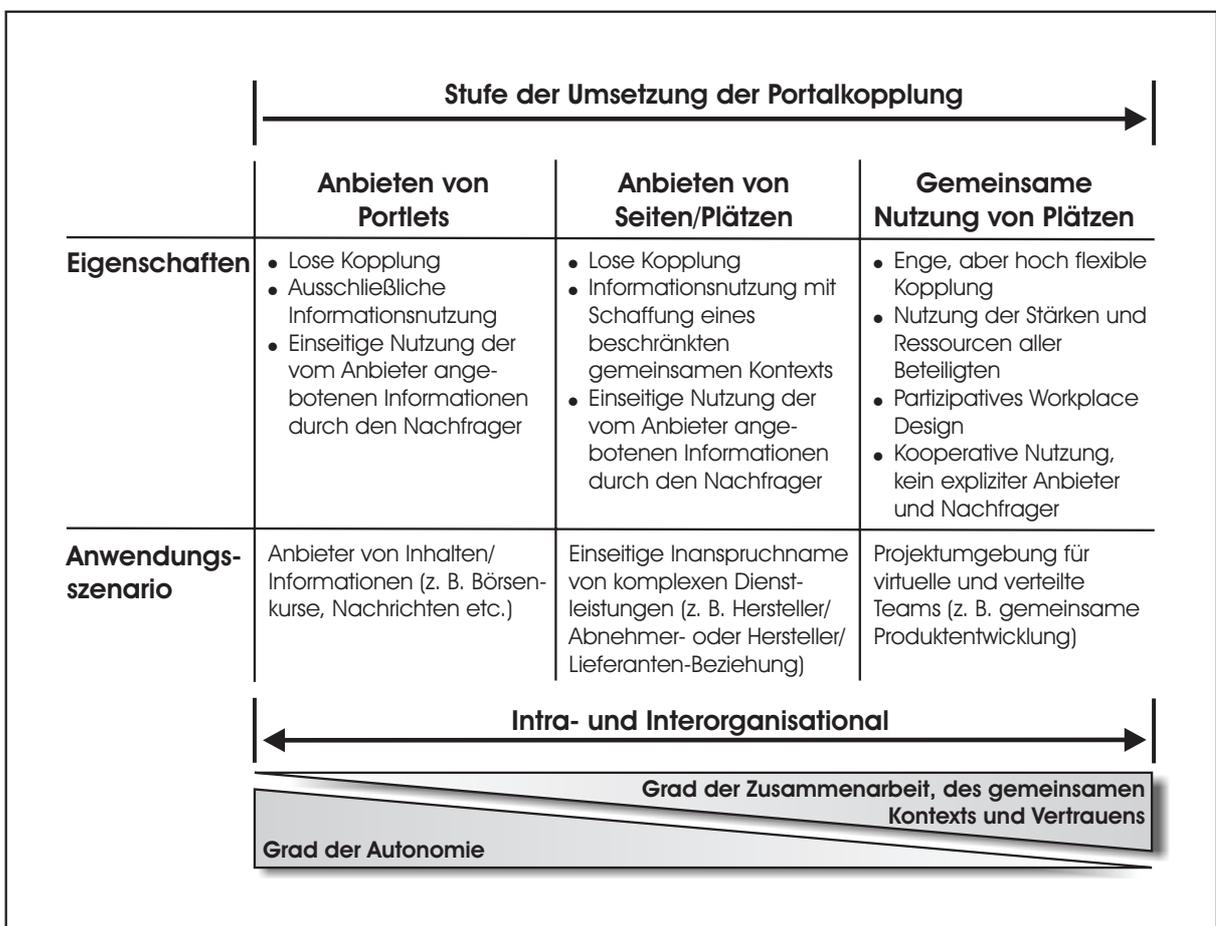


Abbildung 4-4: Kontinuum der Portalkopplung

Die einfachste Form zur Kopplung von Portalen besteht im ausschließlichen *Anbieten von Portlets*. Portlets, die bereits lokal verfügbar sind, werden hierbei auch für externe Portale verfügbar gemacht. Sie sind in geeigneter Form zu publizieren, so dass sie von Kooperationspartnern gefunden und als Portlets in ihr Portal eingebunden werden können. Hierbei handelt

es sich um eine lose und einseitige Kopplung, die ausschließlich der Integration externer Informationsquellen dient.

Für den Anbieter des Portlets ergibt sich das Problem, dass er keinen Einfluss darauf hat, in welchem Kontext das Portlet auf der Nachfragerseite eingesetzt wird. Er hat über die konkrete Verwendung keine Kontrolle, sondern stellt das Portlet ausschließlich für die Nutzung zur Verfügung, ohne dass ein gemeinsamer Kontext entsteht. Der Nachfrager hat einerseits die Möglichkeit, Portlets mit externen Informationen in sein eigenes Portal zu integrieren, um so den Single Point of Access wieder herzustellen. Andererseits ist es aber seine Aufgabe und sein Aufwand, evtl. in einem Kontext zueinander stehende Portlets eines Anbieters selbst zu identifizieren und gemeinsam zu verwenden.

Diese Form der Portalkopplung ist besonders für Inhaltenanbieter z. B. von Börsenkursen, Nachrichten verschiedener Bereiche etc. interessant, da ihre Portlets selten einen besonderen Kontext benötigen bzw. untereinander haben. Sie ist aber zugleich auch Grundlage für die beiden weiteren Formen der Portalkopplung.

Das *Anbieten von Seiten oder Plätzen* berücksichtigt besonders den Aspekt der Kontextlosigkeit einzelner Portlets, der sowohl für den Anbieter als auch für den Nachfrager problematisch sein kann. Hierbei gruppiert der Anbieter zusammengehörige Portlets zu Seiten bzw. Plätzen. Diese Entitäten werden anschließend publiziert und für externe Kooperationspartner zugreifbar gemacht.

Dies ermöglicht auf Seiten des Nachfragers ebenfalls die Wiederherstellung des Single Point of Access und die Reduktion der Anzahl an Portalen, auf die von den Benutzern direkt zugegriffen werden muss. Da die Definition der Seiten bzw. der Plätze ausschließlich vom Anbieter vorgenommen wird, ist es dem Nachfrager jedoch nicht möglich, auf diese Einfluss zu nehmen, d. h., er kann die Zusammenstellung und Anordnung der Portlets nicht verändern, um sie seinen Bedürfnissen anzupassen. Gleichzeitig ermöglicht es aber auch einen eingeschränkten gemeinsamen Kontext zwischen Anbieter und Nachfrager, da beide im Rahmen der Nutzung der publizierten Seiten oder Plätzen eine einheitliche Umgebung vorfinden.

Ein Szenario, in dem diese Form der Kopplung von Portalen zur Anwendung kommen kann, ist eine Hersteller/Abnehmer-Beziehung. Bisher musste sich der Abnehmer beim Hersteller an dessen Portal anmelden, um dort miteinander in Beziehung stehende Portlets in einer oder mehreren Seiten mit allgemeinen Vertriebsinformationen, Produktankündigungen, Datenblättern, Bestell- und Auskunftssystemen etc. vorzufinden. Diese Seite oder Seiten können nun

vom Hersteller für einen bestimmten Benutzerkreis publiziert werden. Der Abnehmer kann sie, z. B. direkt für seine Einkaufsabteilung, in das eigene Portal integrieren und sorgt so für die zentrale Bereitstellung der notwendigen Informationen und Anwendungen.

Die am weitesten fortgeschrittene Form der Kopplung von Portalen ist die *gemeinsame Nutzung von Plätzen*. Hierbei handelt es sich um den höchsten Grad der Zusammenarbeit und des gemeinsamen Kontexts, einhergehend mit einer weiterhin flexiblen, aber engeren Verbindung der Kooperationspartner.

(Portal-)Plätze werden als portal- und ggf. unternehmensübergreifende Informationsräume gekoppelter Portale verstanden. Dies manifestiert sich darin, dass jeder Teilnehmer, sofern er die notwendigen Rechte besitzt, Seiten und Portlets und damit entsprechende Informationen, Anwendungen und Prozesse einem gemeinsam genutzten Platz hinzufügen kann. Er bringt die Elemente ein, die seinen Anteil an der Kooperation betreffen. Insgesamt entsteht ein partizipatives Workplace Design mit dem Ziel der Zusammenführung der Stärken aller Beteiligten und der Schaffung einer gemeinsamen Arbeitsumgebung.

Eine explizite Unterscheidung in Anbieter und Nachfrager ist nicht zweckmäßig, da prinzipiell jeder beide Rollen gleichzeitig einnimmt. Dadurch, dass alle Informationen allen in einer gemeinsamen aufgaben- und kooperationsbezogenen Arbeitsumgebung zur Verfügung stehen, entsteht ein gemeinsamer Kontext, der die Zusammenarbeit wesentlich erleichtert. Gleichwohl arbeitet jeder Teilnehmer ausschließlich in seinem eigenen Portal, dem Single Point of Access, der jeweils den Zugriff auf die gemeinsame Arbeitsumgebung für die Benutzer transparent ermöglicht.

Wesentliche Einsatzfelder sind Projektumgebungen für intra- und interorganisationale virtuelle und verteilte Teams sowie allgemein engere inner- und zwischenbetriebliche Kooperationen zwischen Unternehmensteilen bzw. vollständigen Unternehmen, bei denen z. B. eine gemeinsame Produktentwicklung oder andere informations- und koordinationsintensive Aufgaben verfolgt werden.

4.3.3 Taxonomie gekoppelter Portale

In Analogie zur Taxonomie von Özsu und Valduriez für Multi- und föderierte Datenbankmanagementsysteme (vgl. Abschnitt 3.3.1.3) lässt sich eine Taxonomie gekoppelter Portale anhand der Dimensionen Verteilung, Heterogenität und Autonomie festlegen, die den Raum potenzieller Realisierungsformen aufspannt (vgl. Abbildung 4-5). Auch in diesem Fall gilt,

dass es sich nicht um dichotome Merkmalsausprägungen handelt, wie von der Benennung der Extremausprägungen suggeriert, sondern um in weiten Bereichen kontinuierliche Merkmale. Das im vorherigen Abschnitt eingeführte Kontinuum der Portalkopplung ist hierzu orthogonal zu sehen.

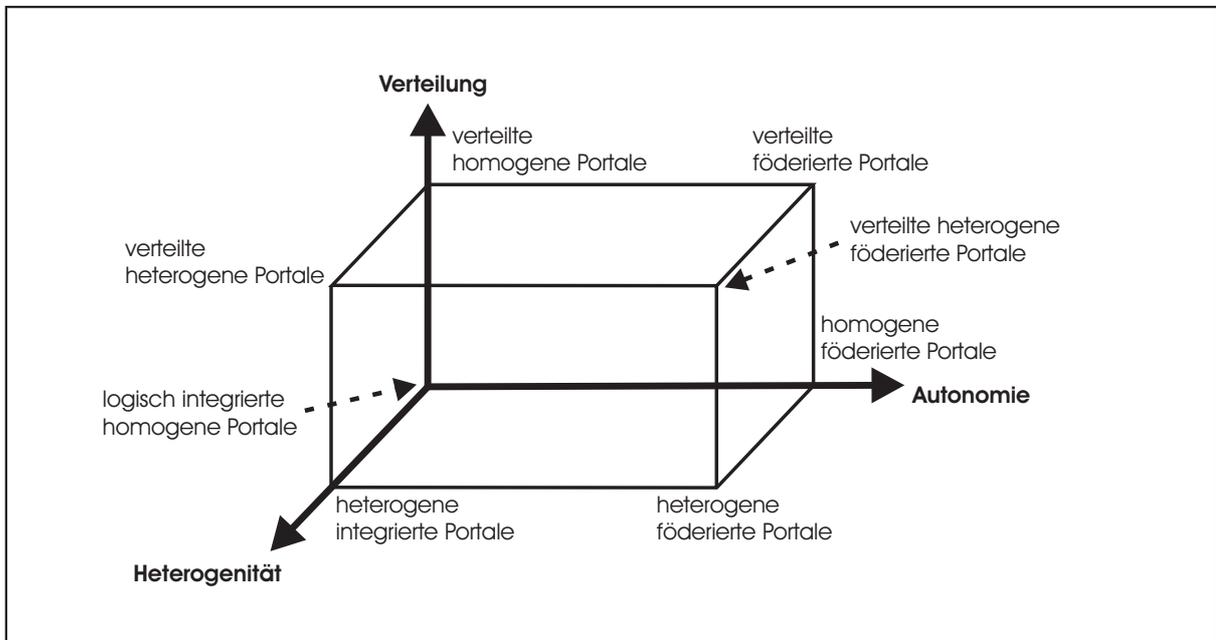


Abbildung 4-5: Taxonomie von Portalkopplung unter Berücksichtigung der Dimensionen Verteilung, Heterogenität und Autonomie

Abhängig von dem Szenario, in dem gekoppelte Portale eingesetzt werden sollen, können die Merkmalsausprägungen variieren. Sind auf dem gleichen Portal-Framework aufsetzende Abteilungs- oder Rollenportale (vgl. Abschnitt 2.2.1.2) eines Unternehmens miteinander zu verbinden, so sind diese dem Bereich verteilter und homogener Portale zuzurechnen, da in diesem Bereich die Autonomie der einzelnen Portale erwartungsgemäß gering ist. Bei der Kopplung von Portalen unterschiedlicher Unternehmen ist zusätzlich zur Verteilung von einer weitgehenden Autonomie der Portale auszugehen. Basieren die zu koppelnden Portale auf demselben Portal-Framework, ist die Kopplung dem Bereich verteilter föderierter Portale, andernfalls dem Bereich verteilter heterogener föderierter Portale zuzuordnen.

Die Taxonomie und genannten Beispiele verdeutlichen, dass es sich bei föderierten Portalen nur um eine von zahlreichen möglichen Ausprägungen gekoppelter Portale handelt. Die in Abschnitt 2.2.2.3 zitierten Autoren haben eine weitreichende Pauschalisierung des Begriffs der Föderation von Portalen vorgenommen, der sich jedoch im Rückblick als unreflektiert und daher unzulässig erweist. Sie subsumieren unter diesem Begriff sehr unterschiedliche Szenarien aus dem Raum *gekoppelter* Portale, der durch die Dimensionen Verteilung, Heterogenität und Autonomie aufgespannt wird.

4.3.4 Zusammenfassung

Die Ausführungen haben gezeigt, dass bei der Konzeptionierung von gekoppelten Portalen unterschiedliche Aspekte zu berücksichtigen sind. Die vorhandene zu koppelnde Infrastruktur und die Rahmenbedingungen der Kooperation der Beteiligten bestimmen die Dimensionen Verteilung, Heterogenität und Autonomie. Gemeinsam mit der Stufe der Kooperation, auf der sich diese aus Sicht des Kontinuums der Portalkopplung befindet, sind sie unmittelbare Determinanten des zu erstellenden Konzepts.

Aufmerksam gemacht wurde auch auf die in der Literatur unreflektierte Verwendung des dort als leere Hülse gebrauchten Begriffs der Föderierung von Portalen. Der intentionale Gebrauch des Begriffs föderierter Portale für das gesamte Spektrum der Portalkopplung ist insofern nur dann zulässig, wenn dieser als Maximalausprägung der Dimension Autonomie verstanden wird, in der alle anderen Formen der Portalkopplung als Spezialisierung bzw. Einschränkung der Autonomie und als Variationen der Ausprägungen der anderen Dimensionen eingeschlossen sind.

In diesem Sinne werden die im Folgenden dargestellten Anforderungen unter den verschiedenen Aspekten betrachtet und die Konzeption mit dem Ziel verfolgt, in jeder Dimension einen möglichst großen Freiheitsgrad des Einsatzes zu ermöglichen, der sowohl tatsächlich als föderierte Portale zu bezeichnende, aber auch alle anderen Formen gekoppelter Portale zulässt.

4.4 Anforderungen

Dem gewählten Jojo-Ansatz (vgl. Abschnitt 4.1) folgend werden im Anschluss die domänen-spezifischen Referenzanforderungen formuliert, die als generische und übergeordnete Anforderungen an die durchzuführende Konzeption zu verstehen sind. Sie schließen sowohl betriebswirtschaftliche als auch technische Anforderungen mit ein.

4.4.1 Betriebswirtschaftliche Anforderungen

Aus betriebswirtschaftlicher Sicht kommt der Kopplung von Portalen in und zwischen Unternehmen eine Unterstützungsfunktion zu, die vor allem den Informations- und Wissensfluss verbessert, die Kommunikation, Koordination und Kooperation fördert und somit indirekt dazu beiträgt, den Geschäftserfolg z. B. durch Kostensenkung, Qualitätsverbesserung und Effizienzsteigerung positiv zu beeinflussen. Dem potenziellen Nutzen stehen aber auch

Kosten gegenüber, die entweder direkt vom Portal oder durch die Kopplung selbst verursacht werden. Mögliche Kosten sind z. B. Kosten für die Entwicklung bzw. den Kauf, die Einführung und den Betrieb. Unter indirekten, nicht direkt vom Portal verursachten Kosten werden solche Kosten verstanden, die dadurch entstehen, dass sich die Rahmenbedingungen für die Portalkopplung ändern und daraus folgend eine Anpassung der Portalkopplung notwendig wird, die ihrerseits Kosten verursacht. Zu den Rahmenbedingungen zählen z. B. der veränderte Grad an Vertrauen zwischen den Partnern, die Aufnahme eines neuen oder das Ausscheiden eines bestehenden Partners und die Änderung der Art der Kooperation.

Als allgemeines Ziel zur Unterstützung des Geschäftserfolgs und zur Reduktion der indirekten Kosten kann formuliert werden, dass – unter der Annahme, dass die Kosten insgesamt geringer sind als die monetär bewerteten erzielten positiven Effekte – die Portalkopplung den jeweiligen Kooperationsanforderungen entsprechen und dennoch flexibel auf Veränderungen der Rahmenbedingungen reagieren können muss. Der Aspekt der indirekten Kosten einer Portalkopplung wird im Hinblick auf die Spezifität und Flexibilität präzisiert (Abschnitt 4.4.1.1) und die betriebswirtschaftliche Betrachtung durch einen weiteren Aspekt der Sichtbarkeit der Interna einer Organisation (Abschnitt 4.4.1.2) ergänzt.

4.4.1.1 Spezifität und Flexibilität der Bindung

Unter der *Spezifität* einer Kopplung von Portalen wird, wie in Abschnitt 2.1.1.3 allgemein dargestellt, verstanden, inwieweit die Investitionen, die ggf. für das Portal selbst und im Besonderen für die Kopplung getätigt wurden, ausschließlich für diese Kopplung nutzbar und an diese gebunden sind („idiosyncratic investments“, „sunk-costs“ vgl. z. B. [Williamson 1993]) oder ob diese auch für andere Kopplungen eingesetzt werden können.

Flexibilität kann als komplementär zur Spezifität einer Kopplung von Portalen interpretiert werden. Sie beschreibt die Anpassungsfähigkeit der gekoppelten Portale an unterschiedliche Rahmenbedingungen. Je höher die Flexibilität eines Portals zur Kopplung mit anderen Portalen ist, desto einfacher, schneller und kostengünstiger lässt sich eine Kopplung zwischen Portalen auf- oder abbauen. Dies wirkt sich unmittelbar auf die Höhe der Eintritts- und Austrittsbarriere und auf die möglichen Kooperationsformen an sich aus, besonders im Hinblick auf deren Fristigkeit.

Der Einsatz von Technologien zur Kopplung von Portalen kann daher sowohl hemmend als auch unterstützend auf die Flexibilität einer Kooperation wirken. Sobald individuell angepasste, aufwändige Verfahren (hohe Spezifität, geringe Flexibilität) zur Kopplung etabliert

werden, können diese nur durch zusätzliche Kosten verändert oder ausgetauscht werden. Zunächst muss i. d. R. die Amortisation der nur mit dieser Organisation nutzbaren Investition sichergestellt werden; dadurch entstehen Abhängigkeiten zwischen den beteiligten Organisationen (vgl. [Rupprecht-Däullary 1994], S. 129 f.). Diesem Widerspruch, auch als *Flexibilitätsparadoxon* bekannt (vgl. [Picot/Reichwald/Wigand 2001], S. 425 f.), kann durch den Einsatz von Portalen mit standardisierten und zugleich verbreiteten Schnittstellen (geringe Spezifität, hohe Flexibilität) entgegengewirkt werden.

Durch Verringerung der Spezifität und Erhöhung der Flexibilität der Portalkopplung können, mit Unterstützung der IT, Formen der Kooperation realisiert werden, die vorher nicht wirtschaftlich waren. Damit kommt auch aus betriebswirtschaftlicher Sicht der Etablierung und Verwendung von bereichsspezifischen Standards (vgl. Abschnitt 4.1.2) zur Kopplung von Portalen eine fundamentale Bedeutung zu.

4.4.1.2 Sichtbarkeit der internen Organisation

Portale setzen auf vorhandenen Organisationsstrukturen eines Unternehmens auf. Dabei nutzen sie besonders direkt in elektronischen (Organisations-)Verzeichnissen abgebildete oder in Informationssystemen jeglicher Art implizit und explizit kodifizierte Personendaten, Organisationsbeziehungen und weitere Unternehmensinterna. Aus Sicht der Organisationen, die an einer Portalkopplung teilnehmen, können abhängig von der Art der Kooperationsbeziehung und besonders dem Vertrauen, das zwischen den Beteiligten vorhanden ist, unterschiedlich starke Schutzbedürfnisse interner Strukturen, Prozesse, Informationen, Personen etc. gegeben sein. Dies wird mit dem Begriff der *Sichtbarkeit der internen Organisation* belegt.

Für die Portalkopplung besonders relevant sind Richtlinien die festlegen, inwieweit Benutzer- und Organisationsdaten zur Erbringung gemeinsamer Dienste übertragen werden dürfen oder nicht. Enge zwischenbetriebliche Kooperationen, in denen in verteilten virtuellen Teams gearbeitet wird, haben eine ähnlich hohe Sichtbarkeit zumindest eines Teils der internen Organisation, wie dies bei innerbetrieblicher Portalkopplung der Fall ist. Eine gelegentliche Hersteller/Abnehmer-Beziehung erfordert hingegen keine Sichtbarkeit der internen Organisation der Partner der Kopplung.

Ein weiterer Aspekt bezieht sich unmittelbar auf die Informationen, die dem Partner zur Verfügung gestellt werden. Abgesehen von der grundsätzlichen Entscheidung, dem Partner gewisse Informationen zugänglich zu machen, kann die Notwendigkeit bestehen, abhängig von der Art der Kooperation, eine Filterung auf der Ebene einzelner Informationseinheiten, z. B.

Dokumente vorzunehmen. Ein Beispiel soll dies verdeutlichen: Der Partner darf ein Dokument einsehen; abhängig davon, in welcher Beziehung die Partner jedoch zueinander stehen, müssen die zugänglichen Informationen anonymisiert, gewisse Bestandteile ausgefiltert oder anderweitig aufbereitet werden, um dem Schutzbedürfnis der Organisation zu entsprechen.

Diese zwei Aspekte sind bei der technischen Konzeption zu berücksichtigen, um auch hier durch eine hohe Flexibilität die Einsatzmöglichkeiten zu vergrößern und die Spezifität zu verringern.

4.4.2 Technische Anforderungen

Neben den betriebswirtschaftlich geprägten Anforderungen an die Funktionalität einer Portalkopplung sind gleichermaßen technische Aspekte zu betrachten. Beide Bereiche stehen dabei in enger Verbindung und Wechselwirkung. Aspekte der Verteilung, Heterogenität und Autonomie haben unmittelbaren Einfluss auf die Spezifität und Flexibilität der Portalkopplung. Im Umkehrschluss wirken sich geänderte betriebswirtschaftliche Anforderungen bzgl. dieser Einflussfaktoren auf die entsprechenden Gestaltungsmöglichkeiten der technischen Dimensionen aus. Sicherheitsfragen auf technischer Ebene sind gleichermaßen mit Fragen der Sichtbarkeit der internen Organisation verknüpft. Vor diesem Hintergrund findet im Folgenden die Auseinandersetzung mit den Bereichen Verteilung (Abschnitt 4.4.2.1), Heterogenität und Autonomie (Abschnitt 4.4.2.2) sowie Sicherheit (Abschnitt 4.4.2.4) statt. Diese werden ergänzt um die Betrachtung der Kopplungsarchitektur (Abschnitt 4.4.2.3).

4.4.2.1 Aspekte der Verteilung

Der Aspekt der Verteilung wurde in Abschnitt 3.3.1.2 im Rahmen von DBMS als Verteilung der tatsächlichen Daten und Verteilung der Schemainformationen diskutiert. Für gekoppelte Portale lässt sich eine ähnliche Unterscheidung treffen. Es können die Meta-Daten, die den Aufbau und die Anordnung von Plätzen, Seiten und Portlets beschreiben, und die konkreten Inhalte der Portlets unterschieden werden.

Wesentliches Differenzierungsmerkmal zu DBMS ist, dass zur Generierung der Inhalte der Portlets ausschließlich jenes Portal in der Lage ist, das Zugriff auf das zu integrierende, jedoch nicht verteilte oder verteilbare Informations- und Anwendungssystem hat. Dies ist i. d. R. ausschließlich das das Portlet anbietende Portal. Eine Möglichkeit bestünde darin, den einmal generierten Inhalt eines Portlets zwischen den gekoppelten Portalen zu verteilen, um

dann jeweils lokal auf diesen zugreifen zu können. Dies ist aufgrund der dynamischen Inhalte jedoch nicht als sinnvoll zu erachten. Oberstes Ziel ist es, die Benutzer immer mit aktuellen Informationen zu versorgen, was nur durch das das Portlet anbietende Portal möglich ist. Die Verteilung der Meta-Daten über die Plätze, Seiten und Portlets ist hingegen grundsätzlich möglich, da sie sich hinreichend selbst beschreiben und keine Abhängigkeiten zu externen Systemen aufweisen.

Angelehnt an die in Abschnitt 3.3.1.4 vorgestellten Anforderungen an die Transparenz der Verteilung von Daten, so wie Date sie formuliert, kann anhand der folgenden Darstellungen abgeleitet werden, ob es sinnvoll ist, eine Verteilung der Meta-Daten vorzunehmen.

Die *Unabhängigkeit von einem zentralen System* ist nur dann gegeben, wenn die an einer Kopplung beteiligten Portale auch dann funktionsfähig sind, wenn ein oder mehrere Mitglieder der Kopplung ausfallen. Aufgrund der nicht gegebenen Verteilbarkeit der Inhalte der Portlets stehen beim Ausfall eines Portals die Inhalte der von ihm angebotenen Portlets nicht zur Verfügung. Darüber hinaus muss die Funktionsfähigkeit der anderen an der Kopplung beteiligten Portale aber gesichert sein.

Das Konzept der *Fragmentierung* beschreibt das Ziel, die Daten dort abzulegen, wo am häufigsten auf sie zugegriffen wird. Die Meta-Daten über den Aufbau und die Anordnung von Plätzen, Seiten und Portlets werden von allen beteiligten Portalen gleichermaßen wiederkehrend benötigt, um die Anzeige für die Benutzer zu generieren. Diesem Umstand kann die *Replikation* Rechnung tragen, indem Kopien der Meta-Daten bei den auf sie zugreifenden beteiligten Portalen vorgehalten werden können. Die Replikation kann gegenüber der ausschließlich zentralen Speicherung sowohl die Abhängigkeit von einem zentralen System als auch einen bei jedem Zugriff eines Benutzers wiederkehrenden hohen Kommunikationsaufwand zwischen den Portalen vermeiden. Die Replikation führt aber zu dem in Abschnitt 3.3.1.4 diskutierten Problem der Verteilung von Aktualisierungen, das für gekoppelte Portale im Abschnitt 4.5.2 behandelt wird.

Aus der Betrachtung beider Kriterien lässt sich ableiten, dass bei nicht gegebener Verteilbarkeit der eigentlichen Inhalte der Portlets, die Verteilung der Meta-Daten über Plätze, Seiten und Portlets zwingend erforderlich ist, um die Verfügbarkeit der Portale selbst sowie die Stabilität und Performance einer Portalkopplung sicherzustellen.

4.4.2.2 Aspekte der Heterogenität und Autonomie

Wie die Analysen in Abschnitt 4.2 ergeben haben, weisen am Markt verfügbare Portal-Frameworks eine große Heterogenität auf, die auf die Autonomie bei deren Entwurf zurückzuführen ist. Anders als bei konventionellen RDBMS stehen hier nicht der direkte Zugriff und die Nutzung einer durch ein standardisiertes Basismodell definierten Datenrepräsentation im Vordergrund. Die verschiedenen Portalelemente werden logisch über Objekte gekapselt und weisen inhärente und differenzierte Interaktionsmodelle auf. Abgesehen von Standardisierungsbemühungen im Bereich von Portlets (vgl. z. B. [Hildreth 2002], [Java Community Process 2003], [Thompson/Leue/Kropp 2003]) ist eine semantische Heterogenität vorherrschend, die durch verschiedene Objekt-, Interaktions- und Sicherheitsmodelle begleitet wird.

Obwohl in der Regel für die jeweiligen Portal-Frameworks umfangreiche Schnittstellendefinitionen veröffentlicht sind, beziehen sich diese nur auf Teilaspekte und schließen aufgrund der internen Komplexität die Interaktion zwischen Plätzen, Seiten und Portlets nicht mit ein. Selbst wenn hier entsprechende Schnittstellen verfügbar wären, ist aufgrund der in Abschnitt 4.5 dargestellten komplexen Anforderungen, die eine Portalkopplung stellt, nicht davon auszugehen, dass die vorhandenen Möglichkeiten ausreichend sind, um diese erfüllen zu können. Dies führt dazu, dass nicht die Portale einsetzenden Unternehmen selbst auf allen Stufen des Kontinuums der Portalkopplung dazu in der Lage sind, geeignete Maßnahmen zur Überbrückung der Heterogenität und zur Kopplung bestehender Portalsysteme zu ergreifen. Geeignete Standardisierungsbemühungen müssen von den Herstellern ausgehen. Dabei sind durch notwendige Anpassungen, Einschränkungen der Entwurfsautonomie zu erwarten.

Eine an die Schemaarchitekturen aus Abschnitt 3.3.1.5 angelehnte und in Abbildung 4-6 dargestellte Architektur für gekoppelte Portale zeigt die von einem Standardisierungsgremium festzulegenden bereichsspezifischen Standards (mintgrün) und die von den Herstellern logisch vorzusehenden Komponenten (hellgelb), um ihre Portal-Frameworks diesen Standards anzupassen.

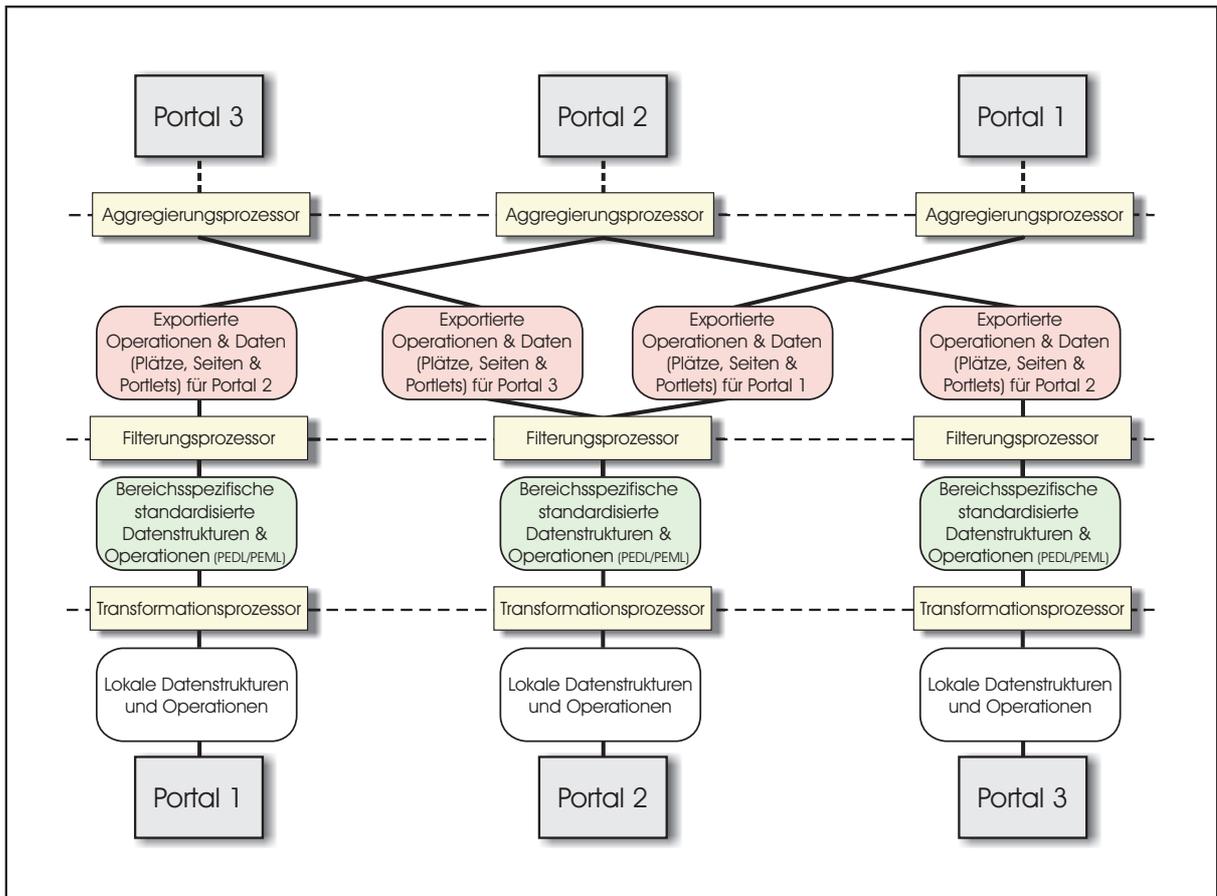


Abbildung 4-6: Komponenten der logischen Architektur zur Portalkopplung

Ausgangspunkt bei der Verwendung eines Top-Down-Ansatzes ist die für jedes Portal logisch vorhandene einheitliche Sicht auf alle für Portalkopplungen zur Verfügung stehenden Portale. Diese wird für jedes Portal mithilfe eines *Aggregierungsprozessors* erzeugt, der Aufgaben zur Unterstützung sowohl der Verteilungs- als auch der Replizierungstransparenz wahrnimmt. Grundlage für den Aggregierungsprozessor sind die Ergebnisse des Filterungsprozessors. Der *Filterungsprozessor* arbeitet auf dem standardisierten Daten- und Objektmodell und bildet daraus nachfragerspezifisch eine Menge von Operationen, die aufgerufen werden dürfen, und eine Menge von Portlets, auf die Zugriff besteht. Der Filterungsprozessor übernimmt damit im Wesentlichen Aufgaben der Zugriffsbeschränkung und ermöglicht damit die flexible Gestaltung der Kommunikations-, Ausführungs-, Autorisierungs- und Verbindungsautonomie (vgl. Abschnitt 3.3.1.2). Das für alle Portale standardisierte Daten- und Objektmodell (Portal Element Definition Language, PEDL) mit auf diesem spezifizierten Operationen (Portal Element Manipulation Language, PEML) wird durch einen *Transformationsprozessor* auf das für jedes Portal eigene lokale Daten- und Objektmodell abgebildet. Dieser letzte Schritt trägt der durch die Entwurfsautonomie entstehenden Heterogenität Rechnung.

Anknüpfend an die Ausführungen in Abschnitt 4.1 sind die PEDL und PEML im Rahmen des domain engineering (vgl. auch Abschnitt 4.6) festzulegen und durch die Implementierung der Prozessorfunktionen im Rahmen des application engineering in neu zu entwickelnde oder bereits bestehende Portal-Frameworks zu integrieren.

4.4.2.3 Kopplungsarchitektur

Aufbauend auf den in den vorherigen Abschnitten gemachten Ausführungen hinsichtlich der Unabhängigkeit von einem zentralen Portalsystem und den Implikationen, die aus der Heterogenität der Portal-Frameworks folgen, ist die Frage zu beantworten, in welcher Form ein Förderierungsdienst zu koppelnder Portale realisiert werden kann.

Bei der Betrachtung föderierter DBS (vgl. Abschnitt 3.3.1.1) wurden als Alternativen ein zentraler oder ein auf die CDBS verteilter Förderierungsdienst identifiziert. Als vorherrschende Variante ist im Bereich föderierter DBS der zentrale Förderierungsdienst anzutreffen. Dieser bietet einige dort diskutierte Vorteile. Neben dem Vorteil, dass die Kommunikation bei n beteiligten Systemen nur n Verbindungen benötigt, wurde speziell herausgestellt, dass die Föderation zentral definiert und vor allem verwaltet werden kann, ohne dass dies von jedem Anwender selbst geleistet werden muss. Dieser Ansatz ist besonders dann geeignet, wenn die Föderation in einer aus Sicht der zu föderierenden Systeme übergeordneten, zentralen Organisationseinheit angesiedelt ist, in deren Verantwortlichkeit die Einrichtung, Verwaltung und der Betrieb liegt. Zudem werden föderierte DBS fast ausschließlich innerhalb eines Unternehmens für dauerhafte, weitgehend zeitlich stabile Föderationen aufgebaut und überschreiten nur in Ausnahmefällen die Unternehmensgrenzen.

Im Fall der Kopplung von Portalen mit der Zielsetzung, sowohl interne, aber vor allem interne und externe Portale miteinander zu verbinden, ist der Einsatz eines zentralen Förderierungsdiensts schwer realisierbar. Dieser Ansatz würde erfordern, für jede, evtl. ad hoc und befristet stattfindende Kopplung bei einem der Partner einen speziellen Förderierungsdienst einzurichten. Dies würde in besonderem Maße die Unabhängigkeit von einem zentralen System verletzen, die Flexibilität im Auf- und Abbau einer Kopplung verringern sowie für einen Partner eine besondere Stellung, aber auch besondere Verpflichtungen und zusätzlichen Aufwand bedeuten.

Um eine möglichst hohe Flexibilität und Autonomie im Hinblick auf die Ausgestaltung möglicher Portalkopplungen zu erreichen, wird ein dezentraler Förderierungsdienst vorgeschlagen, der die notwendigen Funktionen pro Portal zur Verfügung stellt und bei dem die

Portale mithilfe von Punkt-zu-Punkt-Verbindungen flexibel miteinander kommunizieren (vgl. Abbildung 4-7). Obwohl die Punkt-zu-Punkt-Kommunikation bei n Teilnehmern $n*(n-1)/2$ Verbindungen erfordert, trägt dies speziell bei der interorganisationalen Kopplung unternehmenskritischer Infrastrukturen vor allem dem Autonomiebedürfnis der beteiligten Organisationen in größerem Maße Rechnung. Eine intensivere Betrachtung möglicher Topologien erfolgt in den Abschnitten 4.5.1 und 4.5.3.

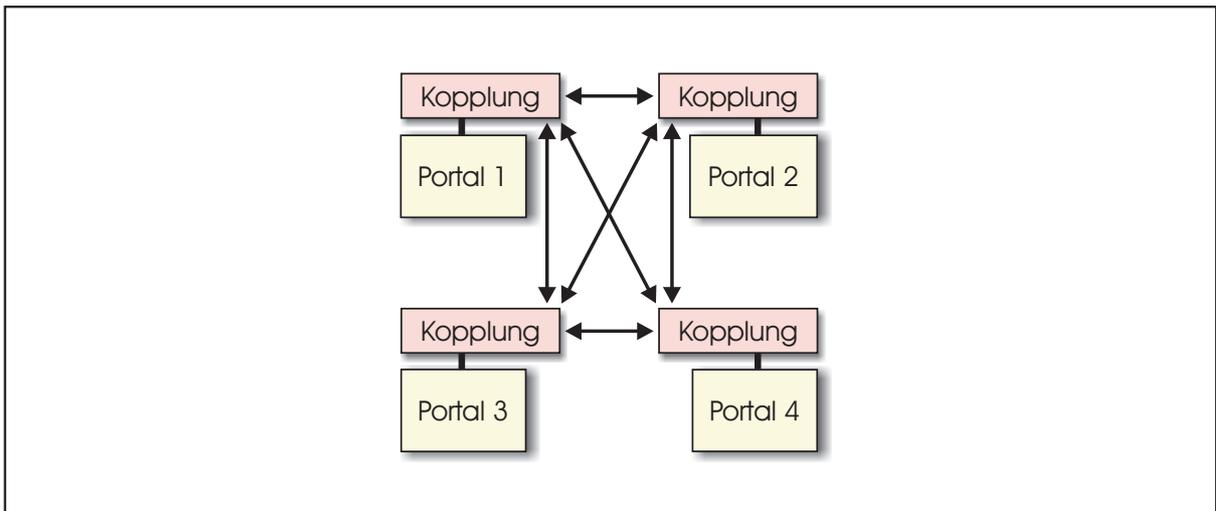


Abbildung 4-7: Architektur der Fördererungsdienste gekoppelter Portale

4.4.2.4 Sicherheit in gekoppelten Portalsystemen

Informations- und Anwendungssysteme verwalten in Unternehmen häufig unternehmenskritische Daten und Informationen, die vor dem Zugriff Unbefugter zu schützen sind. Durch die Portalkopplung werden diese Systeme für den Zugriff berechtigter externer Partner geöffnet, wodurch sich die Gefahr des unerlaubten Zugriffs massiv erhöht. Umso wichtiger ist es, geeignete Maßnahmen zu ergreifen, um die Portalkopplung abzusichern.

Sicherheit lässt sich in die Teilaufgaben Authentifizierung und Autorisierung sowie Vertraulichkeit und Integrität unterteilen. Bevor einem Benutzer oder System Zugriff auf geschützte Informationen gewährt wird, ist im Rahmen der Authentifizierung seine Identität zweifelsfrei festzustellen. Auf Basis der Identität ist im nächsten Schritt zu entscheiden, ob der Benutzer oder das System die Autorisierung besitzt, die entsprechende Aktion durchzuführen. Die Identität wird, wie dies in Verzeichnisdiensten üblich ist, über einen hierarchischen Namen eindeutig festgelegt. Die zwischen den Kommunikationspartnern ablaufende Kommunikation ist des Weiteren davor zu schützen, dass Außenstehende sie belauschen (Vertraulichkeit) oder manipulieren (Integrität) können. Diese allgemeinen Aussagen sind auf den konkreten Fall zu übertragen, um sie in ein Gesamtkonzept integrieren zu können.

Zur Schaffung von Vertraulichkeit und Wahrung der Integrität sind entsprechende standardisierte Methoden und Konzepte verfügbar, die unabhängig vom Einsatz in einer Portalkopplung den Kommunikationskanal zwischen den Endpunkten der Kommunikation absichern und auf die daher nicht weiter eingegangen wird (vgl. hierzu z. B. [Fischer/Rensin/Rödig 2000]). Die Authentifizierung und Autorisierung erfordert dagegen in diesem Kontext eine detailliertere Betrachtung. Zu unterscheiden sind hierbei drei Ebenen, die jeweils verschiedene Maßnahmen notwendig machen. Die erste und zweite Ebene betreffen die Portlets und gelten daher gleichermaßen für alle Stufen der Portalkopplung im Sinne des Kontinuums der Portalkopplung (vgl. Abschnitt 4.3.2). Die dritte Ebene bezieht sich dagegen ausschließlich auf die Verwaltung der Zugriffsrechte auf anzubietenden Seiten oder Plätzen und gemeinsam genutzten Plätzen.

4.4.2.4.1 Anbieten von Portlets

Auf der Ebene der *Portlet-Definitionen* ist festzulegen, wer auf die in ihnen enthaltenen, für eine externe Nutzung notwendigen Daten Zugriff hat, um auf deren Basis konkrete Portlet-Instanzen erzeugen und manipulieren zu können. Möchte ein Portal seinen Benutzern die Möglichkeit bieten, konkrete Portlet-Instanzen auf Basis der von anderen Portalen zur Verfügung gestellten Portlet-Definitionen zu erzeugen, muss es die für den entfernten Zugriff benötigten Daten der Portlet-Definition vorher abrufen. Der Abruf bei den ihm bekannten Portalen wird in einem asynchronen Prozess, unabhängig von konkreten Benutzern, erfolgen. Die getroffene Annahme basiert auf der Abwägung der Umsetzbarkeit des mit der Benutzeraktion synchronen Abrufs, besonders im Hinblick auf dessen mangelnde Performance und Skalierbarkeit bei einer wachsenden Anzahl an bekannten Portalen. Das Vorgehen des asynchronen Abrufs legt nahe, dass das anbietende Portal keine Rechtevergabe auf der Ebene einzelner Portalbenutzer externer Portale durchführen kann. Stattdessen wird vorgeschlagen, einen generischen Benutzer für das nachfragende Portal zu vereinbaren, mit dem der Abruf der verfügbaren Portlet-Definitionen erfolgt. Dem Anbieter gibt das die Möglichkeit, basierend auf dem generischen Benutzer entsprechende Zugriffsbeschränkungen zu definieren. Der Nachfrager kann im Anschluss selbst für das eigene lokale Portal entscheiden, welche seiner – ihm bekannten – Benutzer Zugriff auf einzelne Portlet-Definitionen und damit Portlets bekommen sollen. Gleichermäßen kann er, falls dies vom Anbieter zugelassen wird, die Portlet-Definition weiteren externen Portalen zugänglich machen. Für den Anbieter ist die weitere Verbreitung der Portlet-Definition an dieser Stelle transparent.

Ein Teil der Authentifizierung und Autorisierung ist damit auf den Nachfrager übertragen worden. Dieses Vorgehen macht eine zentrale Benutzerverwaltung oder alternativ die komplexe Mehrfachpflege von Benutzerdaten unnötig und vermeidet gleichzeitig die Notwendigkeit der weitgehenden Sichtbarkeit der internen Organisation des Nachfragers beim Anbieter der Portlets (vgl. Abschnitt 4.4.1.2).

Die Erzeugung einer auf einer Portlet-Definition basierenden *Portlet-Instanz* und die weitere Ausführung von Aktionen erfordert gleichfalls die Authentifizierung des Ausführenden und anschließend dessen Autorisierung zur Durchführung der Aktion. Vom Nachfrager wird diese im Kontext eines konkreten Benutzers durchgeführt werden, so dass dieser entsprechend direkt herangezogen werden könnte. Dieses Vorgehen würde mit den beschriebenen Nachteilen einhergehen, die der beim Zugriff auf die Portlet-Definitionen verfolgte Ansatz vermieden hat. Sinnvoll erscheint es daher, ebenfalls einen generischen, ein Portal identifizierenden Benutzer zur Authentifizierung und grundsätzlichen Autorisierung zu verwenden.

Bei der konkreten Durchführung von Aktionen kann das Sicherheitsbedürfnis größer sein, als beim ausschließlichen Abruf der Portlet-Definitionen, so dass der bisherige Ansatz nicht in allen Fällen ausreichend ist. Es ist zusätzlich ein Mechanismus vorzusehen, die Identität des Benutzers zu übermitteln, der die Aktion tatsächlich durchführt. Auf Seiten des Anbieters des Portlets findet keine Authentifizierung der Identität statt. Da diese von einem authentifizierten Portal übermittelt wird, ist sie – bedingt durch das gegenseitige Vertrauen – als bereits bestätigt zu interpretieren. Auf Basis der Identität können anschließend weitere Autorisierungsprüfungen vorgenommen werden, die in Abhängigkeit von der lokalen Implementierung erfolgen und daher nicht direkt Bestandteil einer Spezifikation sind. Als Empfehlung sollte der Benutzername jedoch nicht direkt verwendet werden, sondern allgemeinere Bestandteile einer im hierarchischen Namen vorhandenen Organisationsstruktur. Beispielsweise lässt sich der Zugriff hierdurch auf die Mitarbeiter einer Abteilung oder eines Standorts eines Partners begrenzen, ohne dass die einzelnen Mitarbeiter selbst bekannt sein und einzeln aufgeführt werden müssen. Dies ermöglicht in der Abwägung der beiderseitigen Interessen einen Kompromiss zwischen dem Sicherheitsbedürfnis des Anbieters und dem Bedürfnis des Nachfragers, die Interna seiner Organisation zu schützen.

Die vorgeschlagene Architektur für die Sicherheit beim Anbieten von Portlets ermöglicht im Sinne des Abschnitts 3.3.1.6 die Verwendung mittlerer und schwacher Autorisierungsautonomie. Abhängig von den konkreten Sicherheitsanforderungen ist jeweils pro Portlet-Definition festzulegen, welche Form der Authentifizierung und Autorisierung durch den

Nachfrager notwendig ist und welche konkreten Regeln einzuhalten sind, z. B. ob Subject Switching (vgl. Abschnitt 3.3.1.6) erlaubt ist oder nicht.

Eine alternative Möglichkeit besteht darin, die Verzeichnisdienste der beteiligten Partner untereinander zu verbinden. Dies widerspricht aber in besonderem Maße der Zielsetzung, keine Sichtbarkeit der internen Organisation zu fordern (vgl. Abschnitt 4.4.1.2), und wird daher nicht berücksichtigt. Auch die im Abschnitt 3.3.3 vorgestellte tatsächliche Föderierung von Benutzeridentitäten ist nicht Bestandteil der Architektur. Dies wäre nur möglich und notwendig, wenn jeder konkrete Benutzer, der eine Portlet-Instanz erzeugen und manipulieren möchte, selbst ein Benutzerkonto beim Anbieter des Portlets unterhalten würde. Der hierfür notwendige Aufwand und die für den Benutzer nicht gegebene Transparenz der Nutzung von durch einen anderen Anbieter zur Verfügung gestellten Portlets schließt dieses Vorgehen im konkreten Anwendungsbereich aus. Die im Abschnitt 3.3.3 diskutierten Techniken zur Schaffung und Transferierung von Vertrauen zwischen den beteiligten Organisationen können gleichwohl auf den aktuellen Anwendungsbereich übertragen werden.

Indem sich das nachfragende Portal direkt gegenüber dem anbietenden Portal authentifiziert, existiert ein direktes, paarweise bestehendes und wechselseitiges Vertrauen. Wird eine Portlet-Definition jedoch von einem Nachfrager weiteren Portalen zugänglich gemacht, die anschließend über diesen entsprechende Aktionen durchführen, dann ist dies nur mithilfe gemakelten Vertrauens möglich. Legt das anbietende Portal für die Ausführung von Aktionen auf Portlet-Instanzen in der zugehörigen Portlet-Definition konkrete Zugriffsbeschränkungen fest, lässt sich dies als der erste im Abschnitt 3.3.3 beschriebene Fall gemakelten Vertrauens identifizieren. Im Unterschied zu der dort formulierten Beschränkung der Stationen zwischen dem Service-Nachfrager und dem Service-Anbieter auf eine, ist bei diesem Vorschlag keine Beschränkung notwendig. Da die Autorisierung der Aktion erst vom Anbieter vorgenommen wird und nicht bereits vom Intermediär, ist es möglich, dass mehrere Portale eine Vermittlerrolle einnehmen. Entscheidend ist, dass der eigentliche Nachfrager beim Anbieter entsprechend autorisiert ist. Ist seitens des anbietenden Portals keine Zugriffsbeschränkung vorhanden, so korrespondiert dies mit dem zweiten Fall des gemakelten Vertrauens, bei dem der Anbieter vollständig der durch den Intermediär vorgenommenen Autorisierung vertraut.

4.4.2.4.2 Anbieten von Seiten und Plätzen und gemeinsame Nutzung von Plätzen

Das Anbieten von Seiten und Plätzen und die gemeinsame Nutzung von Plätzen weisen unter Sicherheitsgesichtspunkten ähnliche Eigenschaften auf, so dass diese im Folgenden gemeinsam betrachtet werden. Bezüglich der in den Seiten und damit auch Plätzen enthaltenen Portlets gelten die im vorherigen Abschnitt gemachten Ausführungen.

Bei der Synchronisation der jeweiligen Meta-Daten ist zu erwarten, dass, obwohl diese explizit durch einen berechtigten Benutzer angestoßen werden kann, sie in der Regel jedoch zeitgesteuert – und damit losgelöst von einem konkreten Portalbenutzer – durch das Portal selbst erfolgen wird. Für die Synchronisation legt dies nahe, dass keine Rechtevergabe auf der Ebene einzelner Portalbenutzer externer Portale erfolgt. Vielmehr ist ebenfalls ein generischer Benutzer für das nachfragende Portal zu vereinbaren, der bereits beim Anbieten von Portlets eingeführt wurde. Unter dessen Identität wird die Synchronisation durchgeführt und es wird ihm grundsätzlich Zugriff auf die entsprechenden Portalelemente gewährt.

Neben den Sicherheitsaspekten bei der Synchronisation ist für anzubietende Seiten und Plätze festzulegen, welche externen Benutzer diese ausschließlich lesend nutzen können. Für gemeinsam genutzte Plätze muss die Zugriffsentscheidung auf die Berücksichtigung weitergehender Rechte an den Plätzen ausgedehnt werden. Da angebotene Seiten alleinstehend nicht einsetzbar sind, sind grundsätzlich implementationsabhängige Mechanismen vorzusehen, mit deren Hilfe die Seiten verwaltet werden können. Aus diesem Grund wird vorgeschlagen, die Zuordnung expliziter Rechte für konkrete Benutzer vollständig auf den jeweiligen Nachfrager auszulagern. In welcher Form diese Zuordnung erfolgt und durch welche Benutzer diese beim Nachfrager vorgenommen werden kann, ist implementationsabhängig und liegt außerhalb einer Spezifikation. Für das Anbieten von Seiten ist es daher hinreichend, einen generischen Benutzer des externen Portals zu verwenden, so dass keine Sichtbarkeit der internen Organisation des Nachfragers für den Anbieter notwendig ist (vgl. Abschnitt 4.4.1.2).

Plätze sind als oberste Aggregationsebene eines Portals direkt verwendbar, erfordern zu deren Nutzung aber entsprechende Zugriffsrechte. Daher wird empfohlen, neben der Festlegung eines generischen Benutzers, der grundsätzlich bestimmt, dass ein externes Portal einen angebotenen Platz nutzen kann, diesem einen konkreten Benutzer des externen Portals zuzuordnen, der als initialer Administrationsbenutzer bei diesem dient. Der Anbieter legt für den initialen Administrationsbenutzer die Rechte des externen Portals auf dem Platz fest. Die weitere Rechtevergabe wird an das externe Portal delegiert. Dieses Vorgehen erlaubt die unmittelbare

Integration des Platzes in die bestehenden Strukturen. Es erfordert dazu zwar die Absprache der Partner und die Festlegung und Übermittlung des Namens des initialen Administrationsbenutzers, vermeidet dafür jedoch weitgehend die Notwendigkeit zur Sichtbarkeit der internen Organisation des Nachfragers. Es findet eine Übertragung der entsprechenden Administration seiner Benutzer an das externe Portal selbst statt. Eine Alternative besteht darin, entweder grundsätzlich eine entsprechend zu diesem Zweck vereinbarte funktionale Rolle zu verwenden oder auf die Zuweisung eines konkreten Administrationsbenutzers beim Anbieter zu verzichten, die dann implementationsabhängig beim Nachfrager nachzuholen ist.

Durch die weitgehende Entkopplung der Rechtevergabe zwischen Anbieter und Nachfrager wird es möglich, eine ggf. vorhandene Heterogenität der Zugriffskontrollkonzepte (vgl. Abschnitt 3.3.1.6) zu überwinden, da diese vollständig voneinander abgeschirmt werden. Dies führt gleichzeitig aber auch dazu, dass der Anbieter ein Stück seiner Kontrolle abgibt und darauf angewiesen ist, dass der Nachfrager die Informationen nur der intendierten (internen) Benutzergruppe zugänglich macht. Trotz dieses Umstandes ist nach Abwägung aller genannten Aspekte nicht vorgesehen, optional die vollständige Kontrolle beim Anbieter zu belassen.

4.5 Kommunikations- und Synchronisationsarchitektur gekoppelter Portale

Nachdem in den vorangegangenen Unterkapiteln die notwendigen Grundlagen gelegt und entsprechende Anforderungen formuliert wurden, kann im Folgenden die eigentliche Konzeption einer Kommunikations- und Synchronisationsarchitektur gekoppelter Portale durchgeführt werden. Diese folgt grundsätzlich dem Jojo-Ansatz und bezieht die dargestellten Anforderungen sowohl aus betriebswirtschaftlicher wie aus technischer Sicht mit ein. Aufgrund der gegebenen Komplexität umfasst die Konzeption ausschließlich die Basiskomponenten eines Portals: Plätze, Seiten inklusive deren Layout und Portlets. Die in Abschnitt 4.2.5 vorgestellten weiteren Dienste sind nicht Bestandteil des Konzepts. Auf sie wird im Ausblick (Abschnitt 7.1.5) eingegangen.

4.5.1 Topologie

Die Topologie, in der gekoppelte Portale miteinander verbunden sind und interagieren, lässt sich angelehnt an bekannte Muster aus der Netzwerkarchitektur (vgl. Abbildung 4-8) konzeptionalisieren. Wesentliche Merkmale der unterschiedlichen Muster sind die Anzahl der Statio-

nen, über welche die Knoten miteinander verbunden sind, und die Auswirkungen, die der Ausfall eines oder mehrerer Knoten für die Kommunikation der anderen Knoten untereinander hat.

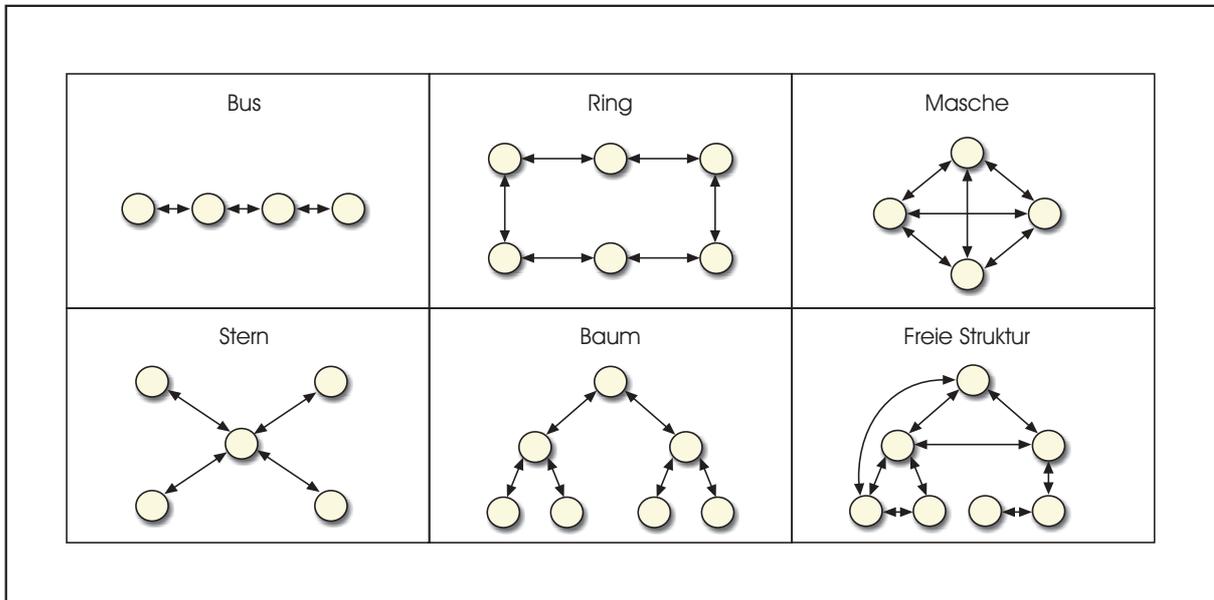


Abbildung 4-8: Netzwerktopologien

Die Struktur der Masche, bei der zwischen allen Knoten direkte Kommunikationsverbindungen bestehen, verspricht sowohl die kürzesten Latenzzeiten als auch die stabilste Form der Vernetzung, erfordert dafür aber auch $n \cdot (n-1) / 2$ Verbindungen, die definiert und administriert werden müssen und diese Struktur wenig skalierbar machen. Bus-, Ring-, Stern- und Baumstrukturen weisen jeweils spezifische Vor- und Nachteile im Hinblick auf die beiden Merkmale auf, die gegeneinander abgewogen werden müssen.

Gewachsene, nicht von einer zentralen Instanz koordinierte und sich immer neu formierende Netze aus miteinander gekoppelten Portalen lassen keine klassischen Netzwerkstrukturen entstehen. Selbst wenn diese angestrebt würden, ist deren dauerhafte Sicherstellung unter wirtschaftlichen Gesichtspunkten nicht immer zweckmäßig. In diesem Umfeld ist daher von sich stetig ändernden freien Strukturen auszugehen, die sich aktuellen Notwendigkeiten anpassen, bei denen alte Verbindungen aufgelöst und neue etabliert werden. Die Masche stellt auch hier die ideale Kombination aus Robustheit und Performance dar. Dadurch, dass einzelne Portale als Transferknoten fungieren, die ihre Kommunikationsverbindung den mit ihnen verbundenen Portalen zur Verfügung stellen, können jedoch auch weniger vollständige, evtl. wirtschaftlichere freie Strukturen zur Anwendung kommen.

Die Kommunikationsverbindung bezeichnet bei gekoppelten Portalen nicht die physikalische Netzwerkverbindung, sondern die logische portalspezifische Verbindung der Förderungs-

dienste zwischen zwei Portalen (vgl. Abschnitt 4.4.2.3). Haben zwei Portale, die an einer mehrere Portale umfassenden Portalkopplung teilnehmen, keine direkte Verbindung, sind hierfür drei Gründe möglich:

- Die beiden Portale sind einander bisher unbekannt.
- Der Aufwand zum Aufbau und der Pflege der direkten Verbindung ist für die vorhandene oder erwartete Häufigkeit der Kommunikationsbeziehung zu hoch.
- Die beiden Portale gewähren einander keine Zugriffsrechte auf die Portalressourcen des jeweils anderen.

Abhängig von der vorliegenden Situation ist es im ersten und zweiten Fall möglich, die Ressourcen des jeweils anderen über ein oder mehrere auf einer Route zwischen den beiden Portalen liegende Portale zu nutzen. Im dritten Fall würde dies zu einer unerwünschten Verbindung führen, was durch entsprechende Richtlinien unterbunden werden muss.

4.5.2 Verteilung

Nachdem in Abschnitt 4.4.2.1 die Vorteilhaftigkeit der Verteilung der Meta-Daten im Verbund der gekoppelten Portale festgestellt wurde, wird unter den gegebenen Randbedingungen ein Konzept entworfen, wie dies realisiert werden kann. Dabei bietet sich ein Rückgriff auf die in Abschnitt 3.3.1.4 thematisierte Replikation als Ansatz an, um Kopien relationaler Daten über verschiedene verteilte Standorte zu synchronisieren. Die dort diskutierten Probleme und Ansätze werden aufgegriffen, auf die konkrete Situation angewandt und entsprechend erweitert.

4.5.2.1 Synchroner vs. asynchroner Replikation

Buretta ([Buretta 1997], S. 44 ff.) beschreibt die beiden Grundmodelle synchrone und asynchrone Replikation, deren wesentliches Unterscheidungsmerkmal darin zu sehen ist, dass nur bei synchroner Replikation jederzeit die Konsistenz des Modells der gekoppelten Portale garantiert werden kann. Das Modell der Portalkopplung umfasst die Meta-Daten über die Struktur von Plätzen, Seiten und Portlets, nicht jedoch die eigentlichen in den Portlets angezeigten Inhalte. Diese können ausschließlich in Echtzeit bezogen werden (vgl. Abschnitt 4.4.2.1) und sind nicht Bestandteil der zu replizierenden Daten. Eine asynchrone Replikation hätte unter diesen Gegebenheiten zur Folge, dass die Benutzer nicht zu jedem Zeitpunkt an jedem an der Kopplung teilnehmenden Portal dieselbe Repräsentation der gemeinsam genutz-

ten Portalelemente zur Verfügung haben. Die innerhalb der Portlets angezeigten Informationen sind aber gleichwohl aktuell. In Abhängigkeit von der jeweiligen Veränderlichkeit der Portalelemente und dem Maß der geforderten Synchronität kann die Zeitspanne zwischen zwei Synchronisierungen kürzer oder länger gewählt werden.

In Abwägung der mit synchroner Replikation verbundenen Nachteile (vgl. Abschnitt 3.3.1.4) und der ausschließlich für die Meta-Daten des Portals auftretenden und hierfür vertretbaren temporären Inkonsistenz, ist die asynchrone Replikation als Ansatz für die Erstellung eines Konzepts zur Modellierung der Kopplung von Portalen vorzuziehen.

4.5.2.2 Identifikation und Verteilung von Aktualisierungen

Im Fall der asynchronen Replikation erfolgt die Verteilung der Aktualisierungen immer losgelöst von der Aktualisierung selbst. Dies macht es notwendig, die zur Wiederherstellung der Synchronität notwendigen Operationen entweder bis zur jeweiligen Synchronisierung vorzuhalten oder alternative Möglichkeiten zur Verfügung zu stellen, um diese rekonstruieren zu können.

Für relationale Datenbanken, bei denen in Datenbeständen von häufig zigtausenden bis Millionen strukturierter Datensätze nur einige wenige verändert werden, wird zumeist der Ansatz verfolgt, ein Journal zu verwalten, das es erlaubt, die seit der letzten Synchronisierung durchgeführten Änderungen zu sammeln und diese bei der nächsten Synchronisierung entsprechend auf das andere System anzuwenden. Zudem handelt es sich ausschließlich um insofern einfache Operationen, als sie in ihrer Dekomposition jeweils ausschließlich auf einzelnen Tupeln durchzuführen sind.

Für komplexere Datenmodelle stößt dieser Ansatz an seine Grenzen. Kawell et al. ([Kawell et al. 1992]) beschreiben einen Replikationsansatz für das dokumentenbasierte, mit verteilten Dokumentendatenbanken arbeitende Notes-System. Als Ziele beim Design des Replikationsmechanismus stellen sie heraus, dass dieser automatisch im Hintergrund ablaufen soll. Ohne den Zugriff auf das System zu beschränken, soll er die Konvergenz der Daten zu einem konsistenten Zustand ermöglichen und alle Datenbanken und Dokumente gleichberechtigt behandeln, ohne „Master-Kopien“ von Objekten vorauszusetzen.

Das Notes-System verwendet kein Synchronisationsjournal, sondern ein auf verschiedenen Zeitstempeln basierendes Modell, um Veränderungen gegenüber der letzten Synchronisation zwischen zwei Repliken derselben Datenbank feststellen zu können. Jedes Dokument kann zu

jeder Zeit in einer beliebigen Anzahl von Repliken in unterschiedlichen Versionen enthalten sein. Pro individuelle Replik kann ein Dokument jedoch maximal einmal existieren; neuere ersetzen ältere Versionen. Zur eindeutigen Identifikation eines Dokuments über alle Repliken hinweg schlagen die Autoren einen eindeutigen, nach der Erzeugung beim Anlegen des Dokuments unveränderlichen Schlüssel vor. Zusammen mit einem Zeitstempel und einem sequenziellen Zähler, die beide bei jeder Veränderung des Dokuments aktualisiert werden, lässt sich entscheiden, welches Dokument zweier Repliken das aktuellere darstellt und beizubehalten ist. Der von ihnen beschriebene Replikationsalgorithmus garantiert, dass nach einer Replikation in beiden Datenbanken die identischen Dokumente vorhanden sind.

Eine Besonderheit stellt die Behandlung von Löschungen dar. Würde die Löschung eines Dokuments zu dessen vollständiger Entfernung führen, könnte bei der Replikation nicht festgestellt werden, dass das Dokument gelöscht wurde. Vielmehr würde angenommen, dass das Dokument in der Datenbank noch nicht enthalten ist, es sich also um ein neues Dokument handelt, das dieser hinzuzufügen ist. Zur Vermeidung dieser Situation ist eine Löschkennzeichnung (engl. deletion-stub) für das Dokument in der Datenbank vorzuhalten, die als Zeitstempel den Zeitpunkt der Löschung beinhaltet. Durch den Vergleich des Zeitstempels der Löschkennzeichnung mit dem Zeitstempel des Dokuments in der anderen Datenbank kann entschieden werden, ob das Dokument gelöscht werden muss. Die Löschkennzeichnungen sind bei der Replikation auch in die Datenbanken zu verteilen, in denen das ursprüngliche Dokument selbst zuvor gar nicht enthalten war. Sie sollten so lange vorgehalten werden, bis die Löschung an alle Repliken verteilt wurde. Da dies nicht feststellbar ist, erfolgt die endgültige Löschung nach Ablauf einer den jeweiligen Anforderungen anzupassenden Zeitspanne.

Um die Anzahl zu replizierender Dokumente gering zu halten, verfolgt das Notes-System eine inkrementelle Replikationsstrategie. Es werden immer nur die nach der letzten Replikation neu hinzugefügten oder veränderten Dokumente repliziert. Weitergehende technische Ausführungen zu Lotus Notes und den Replikationsmechanismen finden sich z. B. bei Kawell et al. ([Kawell et al. 1992]) und Lotus ([Lotus 2000]).

Die bei einer Kopplung von Portalen zu synchronisierenden Meta-Daten lassen sich als komplexe interdependente Datenstrukturen identifizieren, die von komplexen ebenso interdependenten Operationen manipuliert werden. Auf Grundlage der vorherigen Ausführungen scheidet ein journalbasierter Ansatz aus. In Anlehnung an das im Notes-System verwendete Replikationskonzept sind folgende Anforderungen an das Synchronisationskonzept gekoppelter Portale zu formulieren:

- Zur eindeutigen Identifikation der an einer Portalkopplung aktuell und zukünftig teilnehmenden Portale ist jedes Portal mit einer weltweit eindeutigen und unveränderlichen Kennung, die als Portal-ID bezeichnet wird, zu versehen.
- Jedes zu synchronisierende Portalelement ist über alle an einer Kopplung teilnehmenden Portale hinweg eindeutig über einen nach der Erzeugung unveränderlichen Schlüssel identifizierbar.
- Jedes zu synchronisierende Portalelement ist mit einem Zeitstempel zu versehen, der den Zeitpunkt der letzten Änderung des Elements repräsentiert. Dieser ermöglicht es, festzustellen, welches die aktuellere Repräsentation des Elements ist.
- Gelöschte Portalelemente sind nicht sofort vollständig zu entfernen; stattdessen ist eine Löschmarkierung beizubehalten, die den gleichen Schlüssel trägt wie das zu löschende Element und deren Zeitstempel den Zeitpunkt kennzeichnet, zu dem dieses gelöscht wurde. Löschmarkierungen sind bei einer Synchronisation entsprechend den Portalelementen zu übertragen.
- Das Synchronisationsmodell hat bei Durchführung endlich vieler Synchronisationen zu gewährleisten, dass die zu synchronisierenden Portalelemente konvergieren und einen logisch gleichen Zustand erreichen. Dies gilt unter der Annahme, dass alle Veränderungsaktivitäten bei allen miteinander gekoppelten Portalen eingestellt wurden.

4.5.2.3 Vollständig dezentraler vs. dezentraler Ansatz mit Koordinationsinstanz

Zur Konzeptionalisierung bei replizierter Datenhaltung wurden im Abschnitt 3.3.1.4 das Master/Slave- und das Update-anywhere-Modell eingeführt. Kawell et al. ([Kawell et al. 1992], S. 229 f.) fordern für ihr Notes-System, dass alle Datenbanken und Dokumente gleichberechtigt zu behandeln sind, ohne „Master-Kopien“ von Objekten vorauszusetzen. Das entspricht dem Update-anywhere-Modell, bei dem Änderungen zu jeder Zeit an jeder Quelle vorgenommen werden können. Dem steht das Master/Slave-Modell gegenüber, bei dem jedes einzelne Datenelement eine primäre Quelle besitzt und nur diese geändert werden darf. Darüber hinaus sehen die Autoren in ihrer Architektur explizit keine zentrale Instanz vor, die als Verzeichnis alle Repliken einer Datenbank verwalten würde. Das Erstellen einer neuen Replik stellt ein lokales Ereignis dar, das keine Auswirkungen auf andere Repliken hat, so dass es nicht notwendig ist, andere Repliken über zentral koordinierte Nachrichten über das Hinzukommen oder auch Löschen einer Replik zu informieren.

Die Ausführungen machen deutlich, dass zwei Ebenen zu unterscheiden sind. Auf der einen Seite wird ein Modell für die verteilte Datenhaltung betrachtet, die andere Seite betrifft das Management der Verteilung der Repliken einer Datenbank. Der vom Notes-System verfolgte Ansatz ist zusammengefasst als vollständig dezentral zu kennzeichnen.

Für gekoppelte Portale ist dies im Sinne des Kontinuums der Portalkopplung (vgl. Abschnitt 4.3.2) für die ersten beiden Stufen bereits durch deren Funktion und Struktur explizit festgelegt und daher ausschließlich für die dritte Stufe, die gemeinsame Nutzung von Plätzen, zu betrachten. Eine direkte Übertragung auf gekoppelte Portale scheint für diese unter technischen wie betriebswirtschaftlichen Gesichtspunkten nicht ohne weiteres möglich und sinnvoll. Unter der Zielsetzung, ein partizipatives Workplace Design zu ermöglichen, ist es zwingend erforderlich, allen an der Kopplung beteiligten Portalen die Möglichkeit zu geben, zu jeder Zeit die Portalelemente eines Platzes zu verändern, auch wenn dies über eine gezielte Rechtevergabe eingeschränkt werden kann. Bezüglich der Veränderungsmöglichkeit (unter Beachtung der Rechtevergabe) und der Priorität von Veränderungen kommt keinem Portal eine herausgehobene Stellung zu, alle sind gleichberechtigt.

Der Verzicht auf eine für den Platz zentrale Managementinstanz ist hingegen nicht sinnvoll. Im Wesentlichen lassen sich hierfür zwei Argumente anführen. Aus betriebswirtschaftlicher Sicht geht die Initiierung einer Kooperation und im Folgenden die Einrichtung eines gemeinsam genutzten Platzes in der Regel von einem Partner aus, der eine gewisse Führungs- und Koordinationsrolle einnimmt. In Kooperationen mit einer größeren Anzahl an Mitgliedern wird des Weiteren häufig ein Unternehmen explizit mit Koordinatorfunktionen ausgestattet. Eine völlige Dezentralisierung der Managementfunktionen ist grundsätzlich nicht in der Lage, die vorhandene Rolle eines Koordinators sinnvoll abzubilden. Zusätzlich birgt sie die Gefahr der nicht gewünschten und nicht kontrollierbaren Weitergabe sensibler Unternehmensinformationen, indem ein gemeinsam genutzter Platz dezentral und für die anderen Partner nicht erkennbar weiteren, außen stehenden Unternehmen zugänglich gemacht wird.

Ein weiterer Aspekt ist technischer Natur. Im Gegensatz zum Notes-System sind die Daten eines gemeinsam genutzten Platzes aus den in Abschnitt 4.4.2.1 genannten Gründen nicht vollständig verteilbar. Zumindest logisch existiert zu jedem Portlet, das in einen gemeinsam genutzten Platz eingebracht wird, beim Anbieter des Portlets eine als Referenz-Portlet bezeichnete Instanz des Portlets (vgl. die Ausführungen hierzu in Abschnitt 4.2.4). Von diesem werden alle weiteren Portlet-Instanzen der gekoppelten Portale abgeleitet und jeweils mit ihm synchronisiert. Im Umkehrschluss bedeutet dies, dass sich alle bzgl. eines Platzes gekoppelten

Portale diese Instanz des Referenz-Portlets teilen und ausschließlich diese auf das Referenz-Portlet zugreifen dürfen. Daraus folgt, dass entsprechend bekannt sein muss, welche Portale an der Kopplung teilnehmen. Die gemeinsame Nutzung des Referenz-Portlets ist mit Ausnahme des Löschens eines Portlets unproblematisch. Das Löschen eines Portlets führt aber zu der Frage, ob und wann das Referenz-Portlet, auf dem das zu löschende Portlet basiert, selbst gelöscht werden kann. Würde es unmittelbar gelöscht, dann wären davon auch alle anderen gekoppelten Portale betroffen, selbst wenn diese noch keine Synchronisation vorgenommen haben, die sie von der Löschung unterrichtet. Grundsätzlich müsste mit der Löschung des Referenz-Portlets so lange gewartet werden, bis alle beteiligten Portale von der Löschung informiert wurden. Möglich wäre dies indirekt, auch ohne eine zentrale Koordinationsinstanz realisierbar, indem das Referenz-Portlet einen Zähler der Referenzen aufweist. Das Referenz-Portlet dürfte nur dann gelöscht werden, wenn der Zähler null erreicht hat. Neben der in verteilten Systemen potenziellen Unzuverlässigkeit des Ansatzes legen auch die vorhergehenden Ausführungen die Etablierung eines zentralen Koordinators nahe.

Zusammenfassend wird für eine Architektur gekoppelter Portale eine dezentrale Veränderbarkeit eines Platzes bei gleichzeitig zentraler Koordinationsinstanz zum Management der Kopplung eines Platzes vorgeschlagen. Damit werden sowohl die betriebswirtschaftlichen wie auch die technischen Rahmenbedingungen berücksichtigt und trotzdem bleibt eine weitgehende Flexibilität erhalten. Die sich hieraus ergebenden Rollen und Verantwortlichkeiten sind Gegenstand der Betrachtung des folgenden Abschnitts.

4.5.3 Rollen

Bevor genauer auf die zu definierenden Rollen in gemeinsam genutzten Plätzen eingegangen wird, werden die Rollen betrachtet, die auf der ersten Stufe – auch als Grundlage für die weiteren Stufen – und auf der zweiten Stufe des Kontinuums zur Portalkopplung (vgl. Abschnitt 4.3.2) definiert werden.

Portale, die eigene Portlets zur Nutzung durch andere Portale anbieten, werden allgemein als Provider-Portale bzw. spezifischer als Portlet-Provider bezeichnet. Portale, die Portlets, die sie von einem Portlet-Provider beziehen, weiteren Portalen zur Verfügung stellen, werden hingegen allgemein als Intermediär-Portale bzw. spezifischer als Intermediär-Portlet-Provider und der Vorgang selbst als Portlet-Routing bezeichnet. Das Portal, das schlussendlich das Portlet innerhalb einer Seite darstellt, wird allgemein als Consumer-Portal bzw. wiederum spezifischer als Portlet-Consumer bezeichnet, und zwar unabhängig davon, ob der Zugriff

direkt über den Portlet-Provider oder über ein oder mehrere Intermediär-Portlet-Provider erfolgt. Eine entsprechende Visualisierung dieser Rollen stellt Abbildung 4-9 dar. Mit der Einführung von Intermediär-Portlet-Providern und dem damit verbundenen Portlet-Routing können die in Abschnitt 4.5.1 geforderten freien Strukturen für den Einsatz verteilter Portlets realisiert werden.

Bei Portlets, die von einem anderen Portal angeboten werden, wird im Folgenden von Remote-Portlets und entsprechend von *entfernt* angebotenen, abgerufenen etc. Portlets gesprochen. Dies kennzeichnet von der Verwendung ausschließlich eine logische Beziehung, bei der das Portlet von einem anderen als dem eigenen Portal zur Verfügung gestellt wird, kann aber auch mit einer räumlichen Trennung einhergehen.

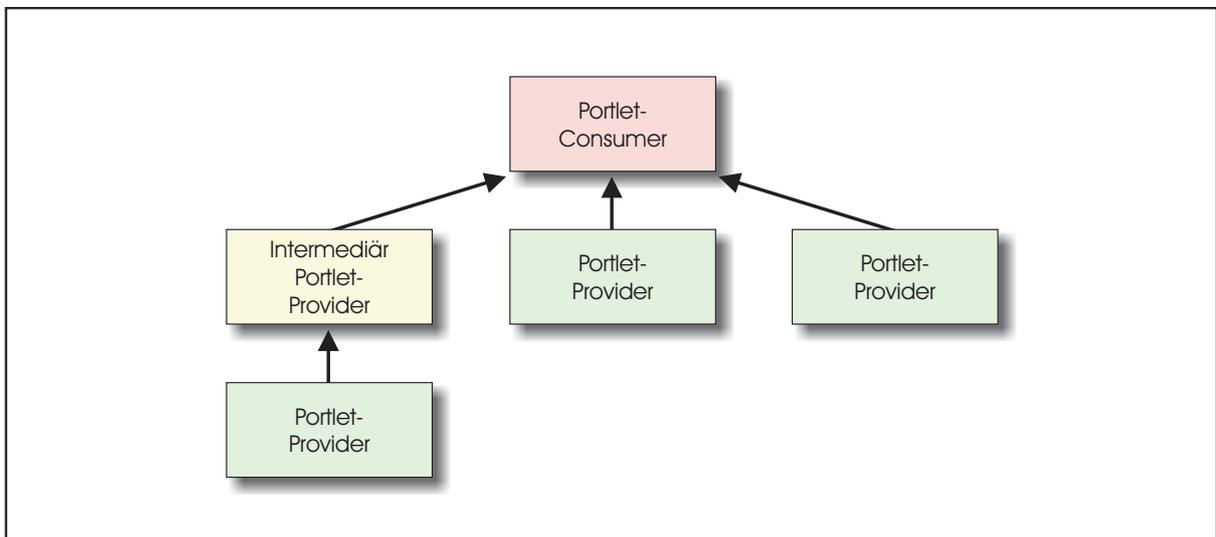


Abbildung 4-9: Rollen beim Anbieten von Portlets

Das Portal, das lokal definierte Seiten oder Plätze für andere Portale anbietet, wird dem obigen Schema entsprechend ebenfalls als Provider-Portal bzw. spezifischer als Seiten- oder Platz-Provider bezeichnet. Es ist bei der momentan intendierten Verwendung von angebotenen Seiten und Plätzen (vgl. Abschnitt 4.3.2) nicht sinnvoll und daher auch nicht vorgesehen, in dieser Stufe zur Verfügung gestellte Seiten oder Plätze indirekt weiteren Portalen zur Verfügung zu stellen, so dass als zweite Rolle nur das Consumer-Portal bzw. spezifischer der Seiten- oder Platz-Consumer vorhanden ist (vgl. Abbildung 4-10). Für die in den zur Verfügung gestellten Seiten oder Plätzen enthaltenen Portlets gelten jeweils die zuvor dargestellten Rollen.

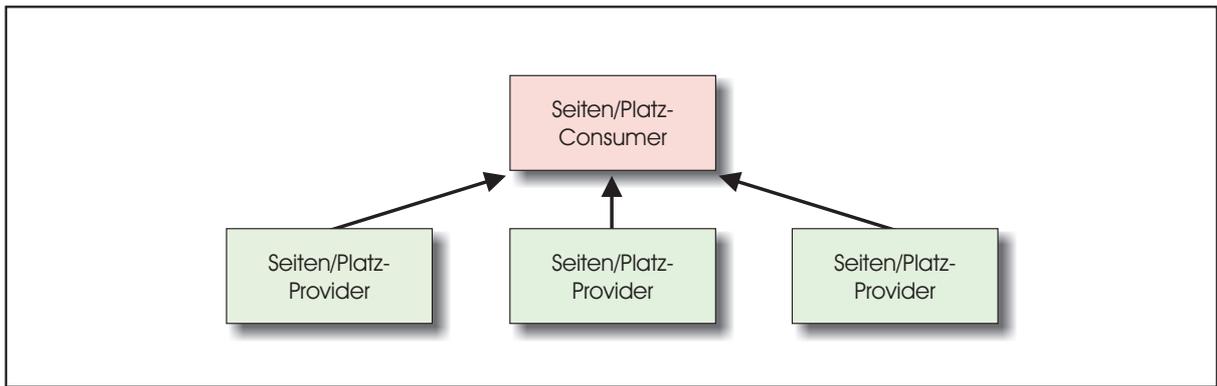


Abbildung 4-10: Rollen beim Anbieten von Seiten und Plätzen

Bei der gemeinsamen Nutzung von Plätzen sind nach Maßgabe des vorherigen Abschnitts grundlegend die zwei Rollen Coordinator- und Consumer-Portal zu unterscheiden. Da diese Rollenverteilung ausschließlich platzbasiert erfolgt, kann jedes Portal jede der beiden Rollen gleichzeitig und mehrfach einnehmen. Der entsprechende Platz des Coordinator-Portals wird als Coordinator-Place und der damit korrespondierende Platz eines Consumer-Portals als Consumer-Place bezeichnet (vgl. Abbildung 4-11).

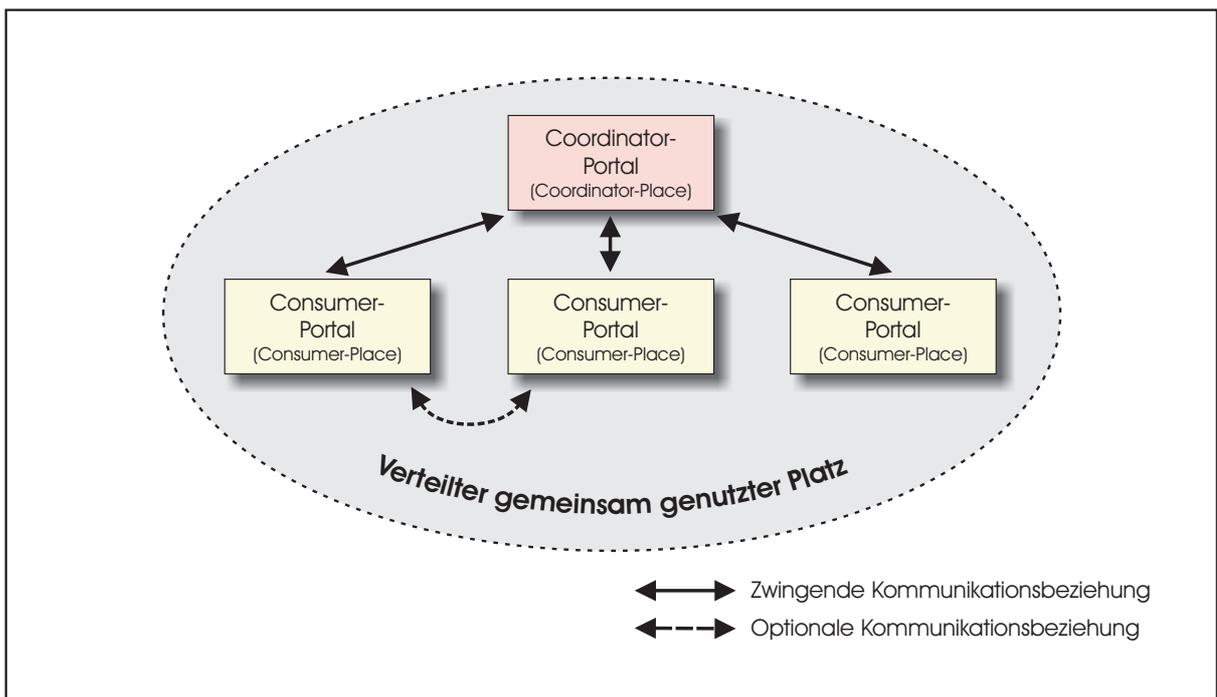


Abbildung 4-11: Rollen bei gemeinsamer Nutzung eines Platzes

Als Konzept zur Umsetzung der Rollenbasierung wird vorgeschlagen, dass ein lokaler gemeinsam genutzter Platz eines Portals durch das explizite Hinzufügen eines weiteren Portals zu einem verteilt gemeinsam genutzten Platz wird. Das Portal, das den Platz initial zur Verfügung stellt, wird automatisch zum Coordinator-Portal und der Platz zum Coordinator-Place. Alle weiteren Portale werden Consumer-Portale und deren korrespondierender Platz jeweils

ein Consumer-Place. Durch den Austritt des letzten Consumer-Portals wird der Platz für das Coordinator-Portal in einen lokalen gemeinsam genutzten Platz zurück überführt. Die Löschung des Platzes beendet grundsätzlich die Verfügbarkeit des Platzes. Dieser Lebenszyklus ist, angelehnt an die UML-Notation eines State-Charts, in Abbildung 4-12 wiedergeben.

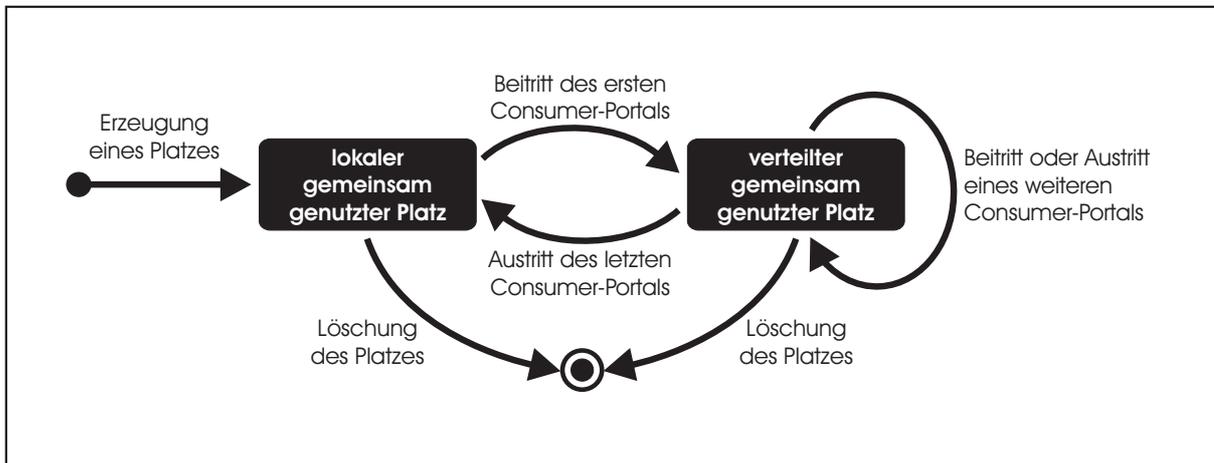


Abbildung 4-12: Lebenszyklus eines gemeinsam genutzten Platzes

Um den Anforderungen aus Abschnitt 4.5.2.3 bzgl. des Managements eines verteilt gemeinsam genutzten Platzes nachzukommen, bleibt es dem Coordinator-Portal vorbehalten, weiteren Portalen den Zugriff auf den Platz einzuräumen, die Zugriffsmöglichkeit wieder zu entziehen, festzulegen, welche Rechte einzelne Consumer-Portale auf dem Platz besitzen, und den Platz endgültig zu löschen. Durch die Rechtevergabe ist es weiterhin möglich, dass mehrere Portale am partizipativem Workplace Design mitwirken; eine ungewollte Verteilung und Ausbreitung, aber auch Löschung des Platzes kann hingegen gezielt verhindert werden. Die Löschung eines Platzes bei einem Consumer-Portal führt ausschließlich zur Beendigung der Teilnahme an der gemeinsamen Nutzung des Platzes, aber nicht zu dessen vollständiger Löschung für alle Teilnehmer.

Durch weitere Regeln ist es darüber hinaus möglich, die Probleme, die durch die Löschung von Portlet-Instanzen auftreten, zu lösen. Das Coordinator-Portal hat die Aufgabe, final darüber zu entscheiden, wann eine Portlet-Instanz endgültig gelöscht wird und damit auch die gemeinsame Instanz des Referenz-Portlets beim Portlet-Provider. Solange eine neu hinzugefügte Portlet-Instanz im Coordinator-Place noch nicht enthalten ist, obliegt die Kontrolle über das Portlet dem erzeugenden Portal, da das Coordinator-Portal diese noch nicht ausüben kann. Sobald die Synchronisation mit dem Coordinator-Portal stattgefunden hat, übergibt das Consumer-Portal diese Aufgabe an das Coordinator-Portal. Um diese Regel gezielt umsetzen zu können, ist es notwendig, dass das Consumer- mit dem Coordinator-Portal synchronisiert wird. Bei der Synchronisation zwischen zwei Consumer-Portalen und anschließender Syn-

chronisation des zweiten Consumer-Portals, das die Portlet-Instanz nicht selbst eingebracht hat, ist eine Übertragung entweder nicht möglich oder führt zumindest zu temporären nicht tolerablen Inkonsistenzen. Selbst wenn sich das erste Consumer-Portal vor der Synchronisation mit dem zweiten Consumer-Portal mit dem Coordinator-Portal synchronisiert hatte, birgt die Synchronisation zwischen den Consumer-Portalen das Risiko von Inkonsistenzen. Dies ist auf die gleiche Problematik bei dem anderen Consumer-Portal zurückzuführen. Aus diesem Grund wird festgelegt, dass die Synchronisation ausschließlich zwischen einem Consumer- und dem Coordinator-Portal erfolgen darf, niemals aber zwischen zwei Consumer-Portalen.

Die gesamte Architektur macht es notwendig, dass die Consumer-Portale zwingend eine direkte Kommunikationsverbindung zum Coordinator-Portal unterhalten; nur so ist das Consumer-Portal dem Coordinator-Portal bekannt und kann in den Kreis der an der verteilten gemeinsamen Nutzung des Platzes beteiligten Portale aufgenommen werden. Weitere Kommunikationsverbindungen zwischen den Consumer-Portalen sind entsprechend den Ausführungen in Abschnitt 4.5.1 wünschenswert, um die Stabilität und Geschwindigkeit beim Zugriff auf die entfernten Portlets zu erhöhen. Für die eigentliche Synchronisation sind sie jedoch nicht notwendig.

4.5.4 Synchronisationsmodell

Kernelement der Portalkopplung ist das Synchronisationsmodell, mit dem die (asynchrone) Synchronisation realisiert wird (vgl. Abschnitt 4.5.2). Im Sinne des Kontinuums der Portalkopplung (vgl. Abschnitt 4.3.2) ist eine Synchronisation zwischen Portalen im Kern nur beim Anbieten von Seiten und Plätzen sowie bei der gemeinsamen Nutzung von Plätzen notwendig. Nur in diesen Fällen findet ein Austausch der Meta-Daten von Portalelementen statt. Für das Anbieten von Portlets ist die Synchronisation der Portlet-Definitionen zu betrachten.

Die folgenden Abschnitte stellen zuerst das Synchronisationsmodell, das in verschiedenen Ausprägungen für die Stufen des Kontinuums der Portalkopplung verwendet wird, vor. Das Synchronisationsmodell basiert auf den in Abschnitt 4.4 definierten Anforderungen und wird anhand ausführlicher Beispiele visualisiert und algorithmisch dargestellt, bleibt aber unabhängig von einer konkreten Implementierung. Das Modell definiert die verbindliche Festlegung von Regeln, die von an einer Portalkopplung teilnehmenden Portalen zu erfüllen sind. Dies gewährleistet bei der Synchronisation die Erreichung konsistenter Zustände. Zusätzlich werden Regeln definiert, die für die synchrone Nutzung von entfernt zur Verfügung gestellten Portlets notwendig sind. Ein eigener Abschnitt beschäftigt sich mit erforderlichen Erweiterun-

gen des Synchronisationsmodells für die gemeinsame Nutzung von Plätzen, die zur Auflösung von bei der Synchronisation auftretenden Konflikten notwendig sind.

4.5.4.1 Definition von Zeitstempeln als Basis der Synchronisation

Im Rahmen der Ausführungen zur Identifikation und Verteilung von Aktualisierungen (vgl. Abschnitt 4.5.2.2) wurde als eine Anforderung formuliert, Portalelemente mit Zeitstempeln der letzten Änderung zu versehen. Ergänzend zu der allgemeinen Anforderung sind zur Durchführung des im Folgenden beschriebenen Synchronisationsalgorithmus präzisere Festlegungen notwendig.

Grundsätzlich hat jedes Portalelement mindestens einen Zeitstempel zu verwalten. Portalelemente, die aus zwei distinkten Eigenschaftsbereichen bestehen, weisen logisch für jeden dieser Bereiche einen eigenen Zeitstempel auf. Dies trifft sowohl auf den Platz zu, bei dem die Eigenschaften wie Name und Kategorie von seiner Struktur zu unterscheiden sind, als auch auf die Seiten, mit Unterscheidung des Namens und der Position von deren strukturellem Aufbau. Ob in konkreten Implementierungen die logisch vorhandenen zwei Zeitstempel tatsächlich unabhängig voneinander verwaltet werden oder ob der eine Zeitstempel für beide Aufgaben herangezogen wird, ist für das Synchronisationsmodell transparent. Die Auswirkung der bei einer Implementierung tatsächlich getroffenen Entscheidung auf die Synchronisation wird im Abschnitt 4.5.4.4 genauer betrachtet.

Die Basisforderung, dass der Zeitstempel den Zeitpunkt der letzten Änderung (inklusive der Löschung) des entsprechenden Elements wiedergibt, ist für die verschachtelte Struktur aus Portalelementen nicht präzise genug. Auf jeder Ebene gelten daher die folgenden Regeln:

- *Portlet-Instanzen*: Der Zeitpunkt, zu dem das Portlet zuletzt verändert wurde, ergibt sich entweder durch dessen explizite oder implizite Veränderung. Eine explizite Veränderung liegt z. B. vor, wenn das Portlet von einem Benutzer an eine andere Position verschoben wurde. Implizit ist dies möglich, indem beispielsweise ein anderes Portlet im selben Spalten-Container gelöscht wurde und das Portlet dadurch indirekt eine andere Position innerhalb des Spalten-Containers erhalten hat. Da es sich bei Portlets um die letzten Elemente der Hierarchie aus Portalelementen handelt, existieren keine weiteren Auslöser zur Veränderung des Zeitstempels.
- *Zeilen- und Spalten-Container*: Grundsätzlich gelten für Zeilen- und Spalten-Container die gleichen Aussagen bzgl. expliziter und impliziter Veränderung und Aktualisierung des

Zeitstempels wie für Portlets. Zwei Erweiterungen sind zu betrachten. Spalten-Container können Portlets enthalten, die ihrerseits verändert werden können, und die Struktur einer Seite besteht aus einer Verschachtelung von Zeilen- und Spalten-Containern. Im ersten Fall wird davon ausgegangen, dass sich die Änderung von Portlets nicht direkt auf die sie enthaltene Spalten-Container auswirkt, so dass deren Zeitstempel nicht anzupassen sind. Im zweiten Fall handelt es sich bei der Änderung eines Containers um ein lokales Ereignis, das sich nur auf den eigenen und die Zeitstempel seiner Geschwister-Container auswirkt, nicht jedoch auf seinen Eltern-Container.

- *Seite*: Jede Veränderung des Namens einer Seite und/oder die explizite oder implizite Veränderung ihrer Position führt zur Aktualisierung des entsprechenden Zeitstempels. Die Veränderung der Struktur der Seite – unabhängig davon, ob Portlets und/oder Zeilen- und Spalten-Container gleich welcher Ebene verändert wurden – führt zur Aktualisierung des hierfür verwendeten Zeitstempels. Veränderungen auf hierarchisch tieferer Ebene werden folglich auf die Ebene der Seite weitergegeben und bei dieser durch einen aktualisierten Zeitstempel vermerkt.
- *Platz*: Die Veränderung der Eigenschaften eines Platzes, wie die Änderung des Namens, aber auch der für die Kopplung relevanten Zugriffsrechte, hat die Aktualisierung des entsprechenden Zeitstempels zur Folge. Analog zu den Seiten führt zusätzlich die Veränderung eines untergeordneten Elements zur Weitergabe dieser Tatsache mit dem Ergebnis der Aktualisierung des zugehörigen Zeitstempels auf der Ebene des Platzes.

4.5.4.2 Synchronisation auf den verschiedenen Stufen der Portalkopplung

4.5.4.2.1 Anbieten von Portlets

Beim ausschließlichen Anbieten von Portlets sind zwischen dem Portlet-Consumer und dem Portlet-Provider keine zu synchronisierenden Meta-Daten der Portlet-Instanzen auszutauschen. Da kein gemeinsamer Kontext besteht, sind die Positionsdaten des Portlets (Container und Position innerhalb des Containers) beim Portlet-Consumer für das Provider-Portal ohne Bedeutung. Stattdessen sind für Portlets in drei Bereichen Festlegungen zu treffen, die eine konsistente Integration ermöglichen.

Synchronisation der Portlet-Definitionen

Möchte ein Portal (Portlet-Provider) anderen Portalen (Portlet-Consumer) Portlets für den entfernten Zugriff zur Verfügung stellen, muss es die zum Abruf notwendigen Daten der Portlet-Definition veröffentlichen. Anschließend kann jeder Portlet-Consumer die Daten der für ihn zugreifbaren entfernten Portlets in einem zyklischen Prozess von allen ihm bekannten Portlet-Providern abrufen (vgl. Abschnitt 4.4.2.4.1). Hierbei sind drei Situationen zu unterscheiden:

- Wird eine neue Portlet-Definition abgerufen, so ist diese implementationsabhängig beim Portlet-Consumer vorzuhalten.
- Wird eine Portlet-Definition abgerufen, die bereits beim Portlet-Consumer bekannt ist, so ist die lokal vorgehaltene Version ggf. zu aktualisieren.
- Wird eine beim Portlet-Consumer bekannte Portlet-Definition nicht mehr vom Portlet-Provider angeboten, so ist diese beim Portlet-Consumer zu löschen.

Behandlung und Synchronisation der Eigenschaften von Portlet-Instanzen

Wie in Abschnitt 4.2.4 erläutert, kann es zumindest logisch-konzeptionell zu Portlets, die in einem (lokal) gemeinsam genutzten Platz enthalten sind, nicht nur eine Instanz geben, die von allen Benutzern gemeinsam benutzt wird. Um eine personalisierte Sicht auf die Inhalte zu gewährleisten und unbeabsichtigte Wechselwirkungen sowie Sicherheitsprobleme zwischen Benutzern auszuschließen, wird für jeden Benutzer auf Basis eines Referenz-Portlets ein eigenes Portlet erzeugt (geklont), das mit dem Referenz-Portlet verbunden ist. Das geklonte Portlet hat denselben eindeutigen Schlüssel wie das Referenz-Portlet. Eine Unterscheidung erfolgt ausschließlich anhand einer Kennung, die das Portlet einem Benutzer zuordnet oder es als Referenz-Portlet ausweist. Der aktuelle Navigationszustand und die persönlichen Einstellungen sind damit für jeden Benutzer des Portlets unabhängig von anderen Benutzern. Wird das Portlet hingegen von einem berechtigten Benutzer über seinen Konfigurationsmodus konfiguriert, so sind Änderungen der Konfiguration für alle auf der Referenz basierenden Portlets einheitlich zu übernehmen. Dies hat transparent für den Portlet-Consumer auf Seiten des Portlet-Providers zu erfolgen, der diese Semantik garantiert.

Regeln zur Generierung und Integration des Inhalts entfernter Portlets

Die Generierung der Inhalte entfernt zur Verfügung gestellter Portlets erfolgt beim Portlet-Provider, die Integration der Inhalte in eine Portalseite jedoch beim Portlet-Consumer. Zur Sicherstellung der Interoperabilität zwischen verschiedenen Portal-Frameworks sind in drei Bereichen konkrete Festlegungen notwendig:

- *Erzeugung von Ausgabefragmenten*: Abhängig vom Ausgabegerät, für das die Darstellung erfolgt, hat das entfernte Portlet eine entsprechende Ausgabe zu erzeugen, die als „Markup“ bezeichnet wird. Das Standard-Markup für Webbrowser ist HTML, für Mobiltelefone WML etc. Der Inhalt des Portlets stellt auf der beim Portlet-Consumer aggregierten Seite jedoch ausschließlich einen Bestandteil einer vollständigen Seite dar. Dies führt dazu, dass der Portlet-Provider kein vollständiges Dokument als Markup generieren darf, sondern ausschließlich ein Markup-Fragment, das in ein bestehendes Dokument eingebettet werden kann. Gleichermäßen ist zu beachten, dass neben dem Portlet selbst weitere Portlets auf der gleichen Seite angezeigt werden können. Bei der Verwendung von JavaScript muss berücksichtigt werden, dass keine Namenskonflikte durch gleiche Funktions- und Variablenamen auftreten. Dies kann beispielsweise durch die Qualifizierung der Namen durch den eindeutigen Schlüssel des Portlets erfolgen, liegt aber außerhalb einer Spezifikation für gekoppelte Portale, da es sich um kein spezifisches Problem gekoppelter Portale handelt.
- *Formatierung und visuelle Integration*: Wird eine Integration eines entfernten Portlets vorgenommen, so soll der Umstand, dass es sich nicht um ein lokales Portlet handelt, für den lokalen Benutzer unerheblich sein. Ein wesentliches Element hierzu stellt die vollständige optische Integration des entfernten Portlets dar. Um Darstellungsunterschiede zwischen verschiedenen Portalen zu vermeiden, sollte das erzeugte Markup des Portlet-Providers mithilfe von für alle Portale standardisierten Formatierungsklassen formatiert werden. Dies ermöglicht u. a. die prinzipielle Auszeichnung eines Bereichs als Überschrift, vermeidet jedoch beim Portlet-Provider die explizite Festlegung einer Schriftart, -farbe und -größe. Die Festlegung kann durch Definition der Formatierungsklasse beim Portlet-Consumer für diesen einheitlich vorgenommen werden.
- *Behandlung von Aktionen und Verknüpfungen auf Ressourcen*: Das Markup der Portlets besteht aus Definitionen zur Erzeugung einer textuellen Ausgabe sowie weiterer Elemente. Dazu zählen eingebettete Ressourcen wie Bilder oder Filme und Verknüpfungen zur Navigation und Ausführung von Aktionen. Im lokalen Fall können diese direkt erzeugt werden. Bei der Erzeugung für die entfernte Verwendung ist jedoch eine Indirektionsstufe, der Portlet-Consumer, zu berücksichtigen. Eingebettete Ressourcen sind häufig von außen nicht direkt zugreifbar, sondern müssen vermittelt durch das Consumer- und Provider-Portal abgerufen werden. Die Ausführung einer durch eine Verknüpfung im Portlet definierten Aktion muss ebenfalls über den Portlet-Consumer an den Portlet-Provider weitergegeben werden, da sie nicht direkt beim Portlet-Provider ausgelöst werden kann. In beiden Fällen

ist es daher notwendig, den beim Portlet-Provider erzeugten Markup entsprechend anzupassen, so dass diese Indirektion berücksichtigt wird. Dies geschieht, indem die Referenzierung der eingebetteten Ressourcen und Verknüpfungen vom Portlet-Provider modifiziert wird. Dazu stellt der Portlet-Consumer die notwendigen Informationen zur Verfügung. Dieser Vorgang wird als „URL-Rewriting“ bezeichnet.

4.5.4.2 Anbieten von Seiten oder Plätzen

Das Anbieten von Seiten oder Plätzen ist entweder als Erweiterung des Anbietens von Portlets oder als Spezialisierung der gemeinsamen Nutzung von Plätzen konzeptionalisierbar. In ersterem Fall würde eine Seite oder ein Platz als eine geschlossene Entität interpretiert, die als solche in das Consumer-Portal zu integrieren wäre. Die Zusammenfassung einer Vielzahl von Portalelementen zu einem Aggregat auf Seite des Provider-Portals verhindert die granulare Einflussnahme auf die visuelle Darstellung beim Consumer-Portal. Auf Grund dieser erheblichen Einschränkung wird der zweite Fall, die Spezialisierung der gemeinsamen Nutzung von Plätzen angenommen. Dies gilt insofern, als es sich um ein gegenüber der gemeinsamen Nutzung von Plätzen einseitiges Verhältnis handelt. Auf der Seite des Consumer-Portals ist keine Änderung der Meta-Struktur der Seite und/oder des Platzes möglich. Diese können nur durch das Provider-Portal vorgenommen werden. Veränderungen sind in geeigneter Weise vom Provider-Portal an das Consumer-Portal zu kommunizieren, das diese lokal anwendet und einen konsistenten Zustand herstellt. Dadurch, dass die Seite bzw. der Platz beim Consumer-Portal als Komposition aus einzeln definierten Portalelementen vorliegt, kann das Consumer-Portal weitgehenden Einfluss auf die visuelle Darstellung nehmen. Konflikte können durch die Beschränkung zur Durchführung von Änderungen auf das Provider-Portal nicht auftreten.

Die im Folgenden zum Synchronisationsmodell bei gemeinsamer Nutzung von Plätzen gemachten Ausführungen gelten daher analog für das Anbieten von Seiten oder Plätzen. Durch die Einseitigkeit der Veränderungsmöglichkeit ist immer nur eine Phase zum Erkennen und Übernehmen von Modifikationen zu durchlaufen. Die im Bereich des Konfliktmanagements notwendigen Erweiterungen des Synchronisationsmodells sind für diesen Anwendungsbereich aus oben erläuterten Gründen nicht relevant.

4.5.4.2.3 Gemeinsame Nutzung von Plätzen

Das Synchronisationsmodell für die gemeinsame Nutzung von Plätzen umfasst alle zu einem Platz gehörenden Basiselemente eines Portals. Dies sind der Platz mit seinen Eigenschaften selbst, Informationen über die Seiten und deren Eigenschaften sowie die in ihnen enthaltenen Portlets. Auf Ebene des Platzes und jeder Seite sind die Eigenschaften, z. B. der Name des Platzes oder der Seite, von deren strukturellem Aufbau, z. B. welche Seite befindet sich an welcher Position innerhalb des Platzes oder welche Seite hat welche interne Struktur aus Zeilen- und Spalten-Containern, zu unterscheiden. Diese sind jeweils voneinander unabhängig und daher auch separat zu betrachten und zu behandeln.

Ein vollständiger Synchronisationszyklus (vgl. Abbildung 4-13), der als Ergebnis die Synchronität eines Platzes zwischen Coordinator- und Consumer-Place hat, setzt sich aus zwei Hauptphasen zusammen. In der ersten Hauptphase, die sich jeweils in verschiedene Subphasen für die Basiselemente gliedert, wendet der Coordinator-Place die Repräsentation des Consumer-Place auf die eigene an. Daraus ergibt sich die zu diesem Zeitpunkt verbindliche Repräsentation des Platzes. Im zweiten Schritt wendet der Consumer-Place diese auf die eigene an, so dass nach Abschluss der zweiten Hauptphase die beiden Repräsentationen im Sinne der gemeinsamen Nutzung des Platzes identisch sind. Allerdings ist sie nicht als vollständig identisch zu bezeichnen, da, wie im Folgenden erläutert, in einzelnen Bereichen beabsichtigte Unterschiede bestehen bleiben können.

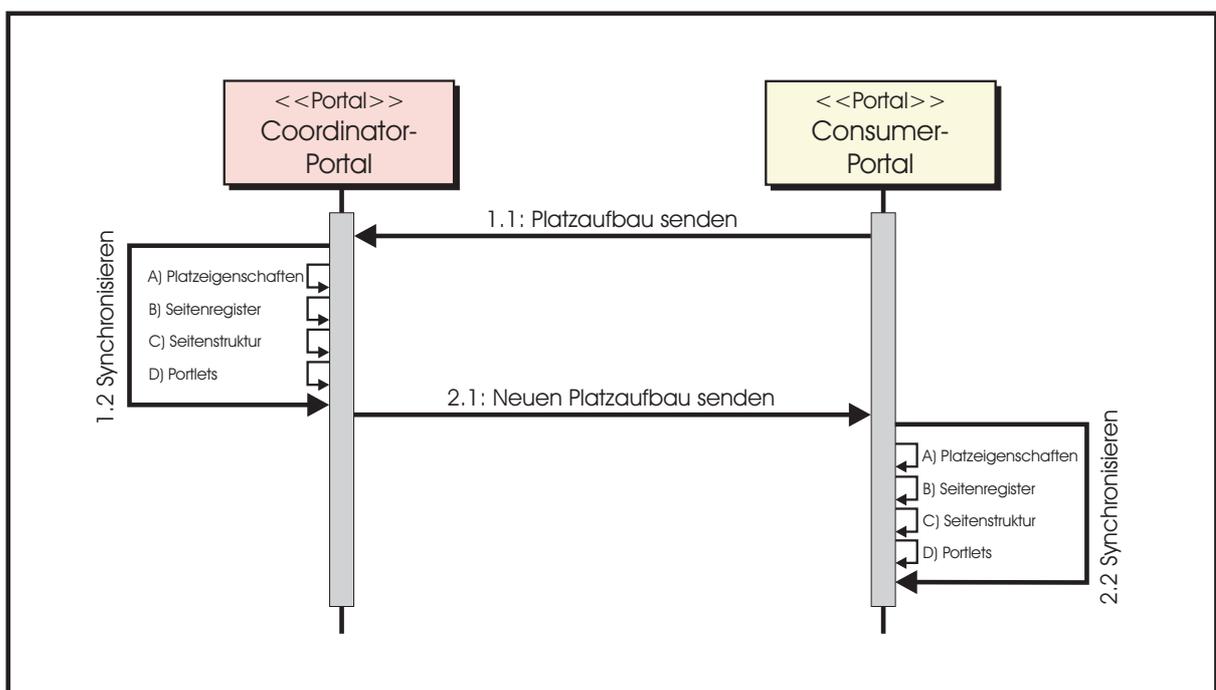


Abbildung 4-13: Haupt- und Subphasen des Synchronisationszyklus

Synchronisation von Eigenschaften und Rechten

Das Synchronisationsmodell legt zum Umgang mit den Eigenschaften „Name eines Platzes“ und „Namen der Seiten“ keine verbindliche Vorschrift fest. Es wird lediglich empfohlen, die Synchronisation der Namen der Seiten immer, die Synchronisation des Namens eines Platzes hingegen nur optional durchzuführen. Dies ermöglicht es unterschiedlichen Organisationen, auf der einen Seite unterschiedliche interne Benennungen von Plätzen vorzunehmen, auf der anderen Seite wird der gemeinsame Kontext nicht durch unterschiedliche Namen für die gleichen Seiten beeinträchtigt.

Wie im Abschnitt 4.4.2.4 dargestellt, wird die eigentliche Administration der Zugriffsrechte auf gemeinsam genutzten Plätzen jeweils im lokalen Platz und nicht zentral vorgenommen. Dem Eintrag, der die Mitgliedschaft eines Consumer-Portals definiert, kann vom Coordinator-Place ausschließlich das Recht zugewiesen oder entzogen werden, den Platz editieren zu können. Das Consumer-Portal ist dafür verantwortlich, dieses generische Recht korrekt auf sein lokales Rechtesystem abzubilden. Es hat dafür Sorge zu tragen, dass Benutzer, die Zugriff auf den Platz haben, nur dann das Recht zum Editieren des Platzes erhalten, wenn der spezielle Benutzer dieses ebenfalls besitzt. Das bedeutet im Umkehrschluss, dass, falls dem speziellen Benutzer das Recht zum Editieren des Platzes durch den Coordinator-Place entzogen und dieser mit dem Consumer-Place synchronisiert wird, nach Abschluss der Synchronisation kein Benutzer des Consumer-Place mehr das Recht besitzen darf, den Platz zu editieren. Da der spezielle Benutzer vom Coordinator-Place administriert wird, dürfen die Rechte nie und der Name bei expliziter Festlegung durch den Coordinator-Place im Consumer-Place nicht verändert oder der Eintrag gelöscht werden können. Das Coordinator-Portal hat bei der Synchronisation zu garantieren, dass keine vom Consumer-Portal unberechtigt durchgeführten Änderungen übernommen werden.

Synchronisation der Struktur

Die Synchronisation der Struktur eines Platzes umfasst logisch zwei Bestandteile, wobei sich der zweite in zwei weitere Bestandteile untergliedern lässt. Zum einen ist das Seitenregister als Zusammenfassung der in einem Platz enthaltenen Seiten abzugleichen. Es sind getrennt voneinander die Anordnung und Benennung der Seiten zu betrachten. Zum anderen ist je Seite die Struktur der Seite abzugleichen, wobei diese sich aus dem Layout aus Zeilen- und Spalten-Containern und aus der Anordnung der Portlets zusammensetzt.

Das Modell zur *Synchronisation des Seitenregisters* umfasst acht zu berücksichtigende Fälle mit den aus ihnen abzuleitenden Regeln. Es basiert aufgrund der größeren Allgemeingültigkeit auf der Annahme, dass die lineare Anordnung der Seiten durch eine fortlaufende Positionsangabe definiert ist. Alternativ wäre auch eine Listenstruktur möglich, bei der sich die Position durch einen Vorgänger und einen Nachfolger definiert. Im Folgenden wird die Phase 1 der Synchronisation (vgl. Abbildung 4-13), der Abgleich des Coordinator-Place mit dem Consumer-Place, betrachtet. Unter Vertauschung der Rollen gilt dies analog auch für die Phase 2, die Synchronisation des Consumer-Place mit dem neuen verbindlichen Zustand des Coordinator-Place.

Zu berücksichtigende Fälle und Regeln:

1. *Die Seite existiert sowohl im Consumer-Place als auch im Coordinator-Place und ist unverändert.*

In diesem Fall sind die beiden Seiten bereits synchron, so dass kein Abgleich vorgenommen werden muss.

2. *Die Seite existiert sowohl im Consumer-Place als auch im Coordinator-Place, es hat aber eine Veränderung stattgefunden.*

Durch den Vergleich der Zeitstempel der letzten Veränderung der Seite im Coordinator- und im Consumer-Place ergeben sich zwei Alternativen:

- a) Wurde die Seite des Coordinator-Place zuletzt geändert, dann ist für diesen nichts zu tun, da er bzgl. dieser Seite bereits den aktuellen Zustand aufweist.
- b) Wurde die Seite des Consumer-Place zuletzt geändert, dann ist diese Änderung für die Seite im Coordinator-Place zu übernehmen. Dies betrifft sowohl die Bezeichnung und die Position als auch ggf. die Struktur der Seite.

3. *Die Seite existiert im Coordinator-Place, aber nicht im Consumer-Place und wurde in diesem auch nicht gelöscht.*

In diesem Fall ist für den Coordinator-Place nichts zu tun, da er bzgl. dieser Seite bereits den aktuellen Zustand aufweist.

4. *Die Seite existiert im Consumer-Place, aber nicht im Coordinator-Place und wurde in diesem auch nicht gelöscht.*

Im Coordinator-Place wird im Standardfall an entsprechender Stelle eine neue Seite hinzugefügt, die sowohl die gleiche Bezeichnung und Position als auch die gleiche Struktur aus

Zeilen und Spalten sowie Portlets aufweist. Eine Abwandlung der Regel ist für den Konfliktfall notwendig, der in Abschnitt 4.5.4.3.1 beschrieben wird.

5. *Die Seite wurde im Coordinator-Place gelöscht, ist aber im Consumer-Place noch vorhanden.*

Im Standardfall ist für den Coordinator-Place nichts zu tun. Die Seite bleibt gelöscht und die Löschkennzeichnung im Coordinator-Place erhalten. Eine Abwandlung der Regel für den Konfliktfall wird ebenfalls in Abschnitt 4.5.4.3.1 beschrieben.

6. *Die Seite wurde im Consumer-Place gelöscht, ist aber im Coordinator-Place noch vorhanden.*

Im Standardfall ist die Seite im Coordinator-Place zu löschen und eine Löschkennzeichnung für die Seite anzulegen. Im Consumer-Place sind nach erfolgreicher Synchronisation aus weiter unten genannten Gründen die Löschkennzeichnungen nicht weiter vorzuhalten. Eine Abwandlung der Regel für den Konfliktfall ist ebenfalls Bestandteil von Abschnitt 4.5.4.3.1.

7. *Die Seite wurde sowohl im Coordinator-Place als auch im Consumer-Place gelöscht.*

In diesem Fall ist der Zustand der beiden Seiten bereits synchron, so dass kein Abgleich vorgenommen werden muss.

8. *Die Seite wurde im Consumer-Place hinzugefügt, aber vor einer Synchronisation bereits wieder gelöscht.*

Für den Coordinator-Place ist keine Aktion notwendig. Da diese Seite nie im Coordinator-Place bekannt war, kann sie auch in keinem anderen Consumer-Place enthalten sein, so dass auch keine Löschkennzeichnung gespeichert werden muss.

Der vollständige Synchronisationszyklus, der das Seitenregister (ohne Betrachtung der Struktur der Seiten und ohne Auftreten von Konflikten) in einen gemeinsamen Zustand überführt, ist in Abbildung 4-14 bis Abbildung 4-16 dargestellt und schematisch in Tabelle 4-1 und Tabelle 4-2 wiedergegeben.

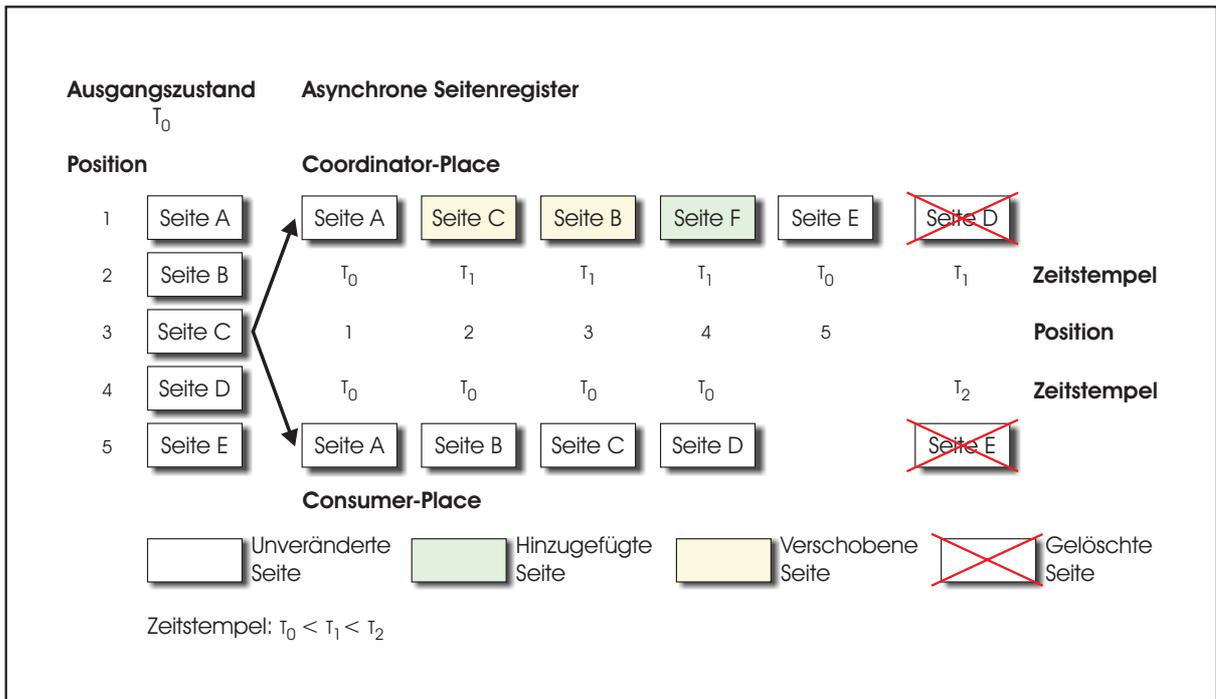


Abbildung 4-14: Synchroner Ausgangszustand und abgeleitete asynchrone Seitenregister

Ausgehend von einem für den gemeinsam genutzten Platz synchronen Seitenregister (vgl. Abbildung 4-14, Ausgangszustand) wird dieses in beiden Portalen unabhängig voneinander verändert (vgl. Abbildung 4-14, Asynchrone Seitenregister). Mit diesem Zustand beginnt die zweiphasige Synchronisation, bei der in der ersten Phase die Seiten des Coordinator-Place mit den Änderungen der Seiten des Consumer-Place aktualisiert werden. Das algorithmische Vorgehen ist für das Beispiel in Tabelle 4-1 dargestellt.

						Phase 1	
Coordinator-Place		Consumer-Place		Fall	Aktion	Ergebnis	
Seite	Änderungszeit		Änderungszeit	Seite			
A	T_0	=	T_0	A	1	Keine Veränderung	-
C	T_1	>	T_0	C	2 a)	Keine Veränderung	-
B	T_1	>	T_0	B	2 a)	Keine Veränderung	-
F	T_1	-	---	---	3	Keine Veränderung	-
D (gelöscht)	T_1	>	T_0	D	5	Keine Veränderung	-
E	T_0	<	T_2	E (gelöscht)	6	Löschen	-

Tabelle 4-1: Phase 1 der Synchronisation des Seitenregisters

Nach dem Abschluss der Phase 1 der Synchronisation befindet sich das Seitenregister des Coordinator-Place in dem in Abbildung 4-15 gezeigten konsistenten Endzustand.

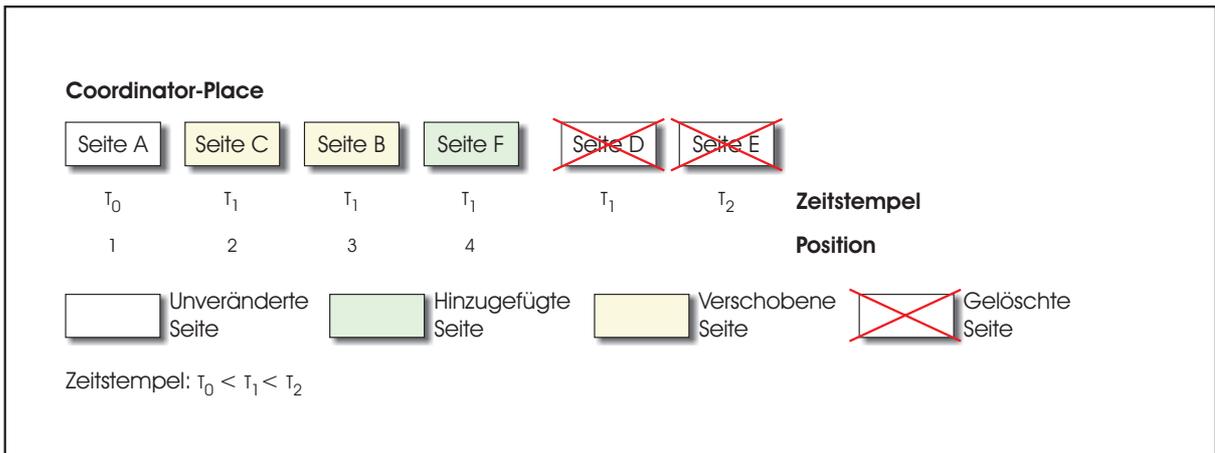


Abbildung 4-15: Seitenregister des Coordinator-Place nach Abschluss der Phase 1 der Synchronisation

In der Phase 2 wird im Anschluss der konsistente Endzustand des Seitenregisters des Coordinator-Place auf das Seitenregister des Consumer-Place angewendet (vgl. Tabelle 4-2).

					Phase 2		
Consumer-Place		Coordinator-Place			Fall	Aktion	Ergebnis
Seite	Änderungszeit		Änderungszeit	Seite			
A	T_0	=	T_0	A	1	Keine Veränderung	-
B	T_0	<	T_1	B	2 a)	Modifizieren	Position 3
C	T_0	<	T_1	C	2 a)	Modifizieren	Position 2
D	T_0	<	T_1	D (gelöscht)	5	Löschen	-
E (gelöscht)	T_2	=	T_2	E (gelöscht)	1	Keine Veränderung	-
---	---	-	T_1	F	3	Hinzufügen	Position 4

Tabelle 4-2: Phase 2 der Synchronisation des Seitenregisters

Nachdem die Phase 2 ebenfalls abgeschlossen ist, ergibt sich das in Abbildung 4-16 gezeigte Bild der Seitenregister im Coordinator- und Consumer-Place. Auffällig ist, dass zwar, wie beabsichtigt, das eigentliche Seitenregister erneut synchron ist, wobei die Zeitstempel gleiche Werte aufweisen, gelöschte Seiten aber nur im Coordinator-Place vermerkt werden. Dies lässt sich mit der gewählten Rollenverteilung erklären, bei der die Synchronisation immer nur zwischen dem Coordinator-Place und den Consumer-Places stattfindet, niemals aber zwischen zwei Consumer-Places. Daher ist es ausreichend, dass, nachdem evtl. vorhandene Löschungen von einem Consumer-Place an den Coordinator-Place übermittelt wurden, diese ausschließlich beim Coordinator-Place vorgehalten werden, um sie an weitere Consumer-Places übermitteln zu können. Für den Consumer-Place, von dem die Löschungen ausgingen, ist es nicht notwendig diese länger vorzuhalten, da sie in dem Synchronisationszyklus bereits in das nun erneut synchrone Seitenregister eingeflossen sind.

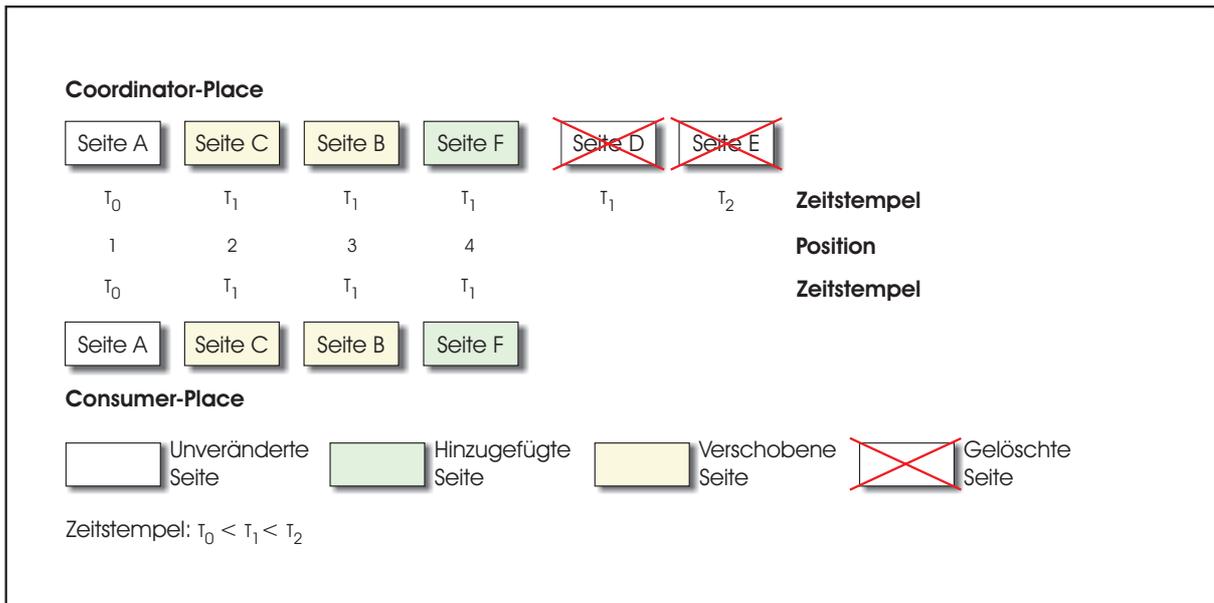


Abbildung 4-16: Seitenregister nach Abschluss der zweiphasigen Synchronisation

Der Umgang mit gelöschten Seiten für den Consumer-Place ist nicht verbindlich vorgeschrieben. Die Berücksichtigung der Empfehlung vermeidet aber die unnötige Übertragung der Daten gelöschter Seiten in folgenden Synchronisationszyklen und kann damit zu einer Steigerung der Performance beitragen.

Das im Folgenden dargestellte Modell zur *Synchronisation der Struktur einer Seite* ist in das Modell zur Synchronisation des Seitenregisters eingebettet. Wird in den Fällen 2, 3 und 4 die Asynchronität der Struktur einer Seite festgestellt, ist ein Abgleich vorzunehmen. Entsprechend der Einbettung in die Synchronisation des Seitenregisters erfordert die Synchronisation der Struktur der Seite ebenfalls zwei Phasen. Jede dieser Phasen lässt sich weiter in zwei Bestandteile untergliedern, welche die beiden Elemente der Seitenstruktur berücksichtigen. Neben der Synchronisation des Layouts aus Zeilen- und Spalten-Containern hat diese auch für die Auswahl und Anordnung der Portlets zu erfolgen. Die Fälle 3 und 4 sind Spezialfälle des allgemeinen Falls 2, bei denen die vollständige Struktur einer Seite in eine neue Seite kopiert werden muss und es sich ausschließlich um eine einseitige Synchronisation handelt.

Als Repräsentation für das Layout der Seite wird die in Abschnitt 4.2.3 identifizierte flexible Zusammensetzung aus Zeilen- und Spalten-Containern zugrunde gelegt. Diese an Tabellen in der Hypertext Markup Language (HTML) angelehnte hierarchische Struktur erlaubt auf oberster Ebene nur Zeilen-Container. Diese müssen immer mindestens einen Spalten-Container enthalten, wobei diese selbst wieder entweder nur weitere Zeilen-Container oder nur Portlets enthalten können (beides zusammen ist nicht zulässig) oder leer sein dürfen (vgl. auch Abbildung 4-3). Durch eine Verschachtelung entstehen komplexe Layouts. Die Position

jedes Containers und jedes Portlets in der Seite ist ausschließlich über den Container, in dem dieses Element enthalten ist, und seine Position in diesem bestimmt. Spalten-Container können als zusätzliche Eigenschaften eine Breite aufweisen. Die Identifizierung eines Containers oder Portlets findet wie bei den Seiten selbst über eindeutige, unveränderliche Schlüssel statt. Ob eine Veränderung stattgefunden hat, kann ebenfalls über entsprechende Zeitstempel ermittelt werden.

Die Synchronisation des Layouts und der Portlets weist ohne das Vorhandensein von Konflikten dieselben Fälle und Regeln auf, wie sie für die Synchronisation des Seitenregisters identifiziert wurden. Deshalb wird auf eine erneute Darstellung verzichtet. Der Ablauf der Synchronisation wird anhand Abbildung 4-17 bis Abbildung 4-19 sowie Tabelle 4-3 und Tabelle 4-4 dargestellt.

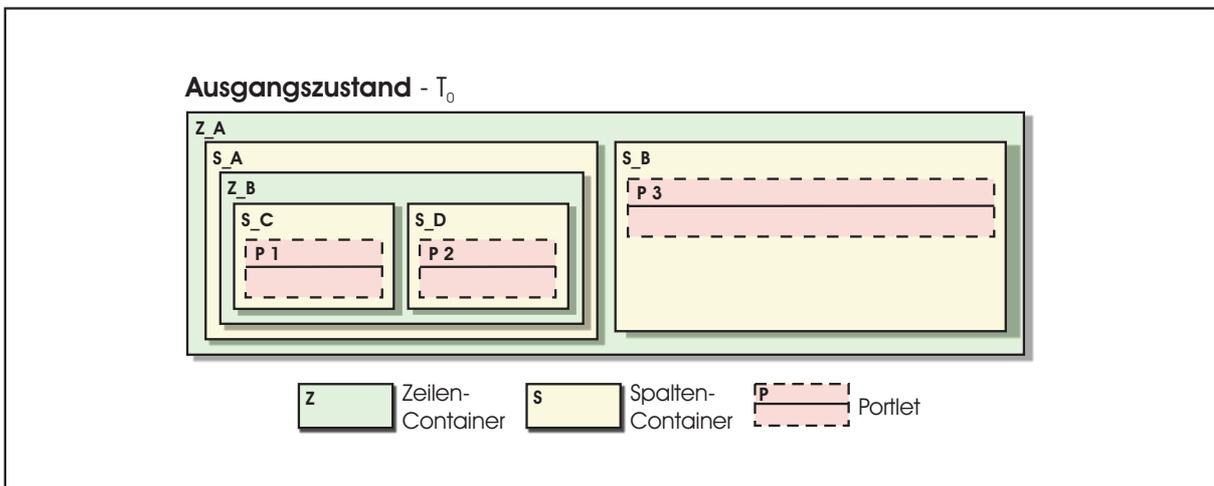


Abbildung 4-17: Ausgangszustand des Layouts einer Seite vor der Synchronisation

Abbildung 4-17 zeigt den gemeinsamen Ausgangszustand der Seite im Coordinator- und Consumer-Place. Die Benennung erfolgt in den Abbildungen für Zeilen-Container mit „Z_“, für Spalten-Container mit „S_“ und jeweils einem fortlaufenden Buchstaben, sowie für Portlets mit „P“ und einer fortlaufenden Ziffer. Dadurch wird jedoch nichts über die konkreten Schlüssel der Elemente ausgesagt. Die Seitenstruktur ist durch die Veränderungen der Container und Portlets sowohl beim Coordinator als auch beim Consumer in einen nicht mehr synchronen Zustand überführt worden (vgl. Abbildung 4-18 obere Hälfte).

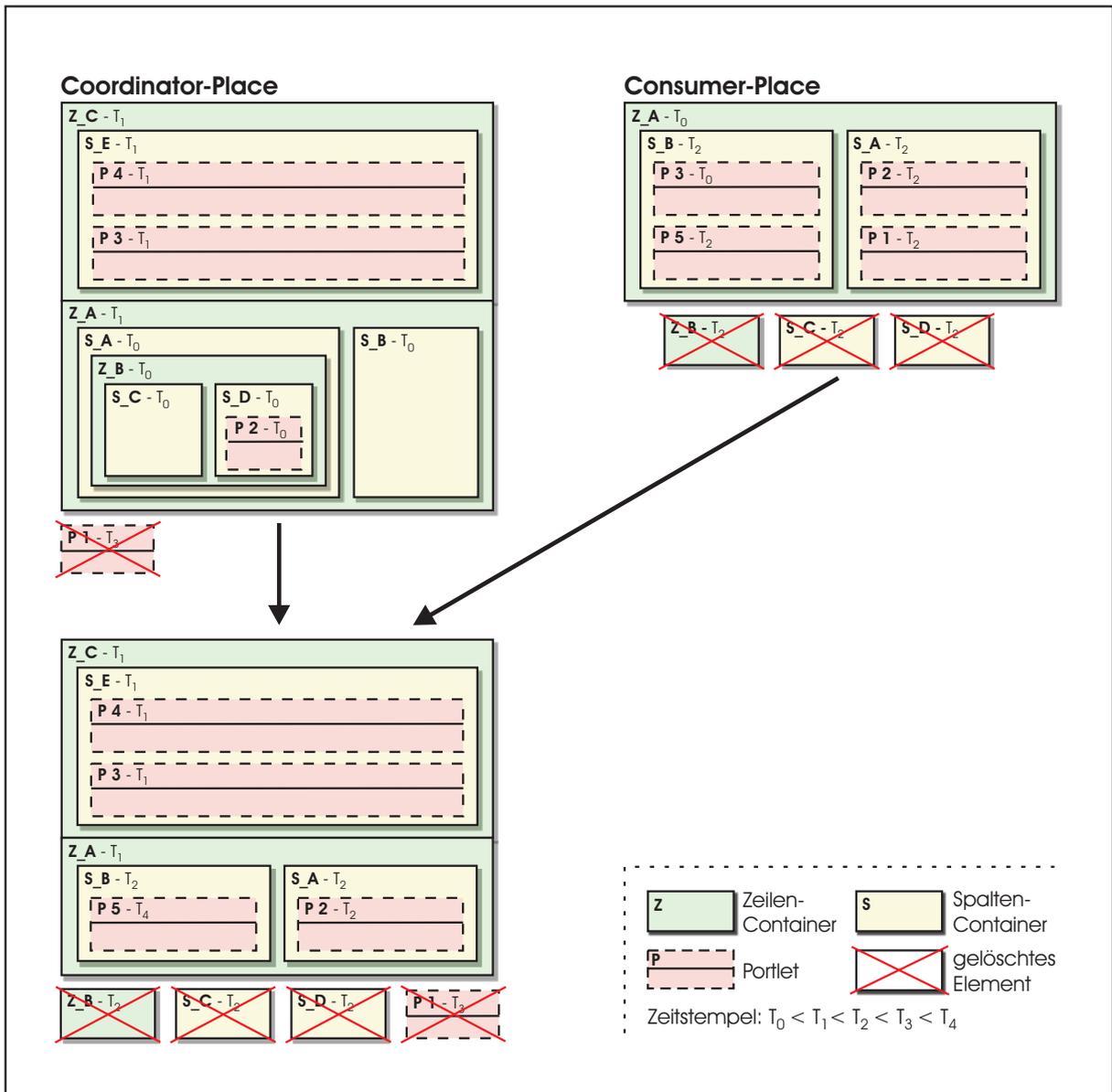


Abbildung 4-18: Visualisierung der Phase 1 der Synchronisation der Seitenstruktur

Durch Anwenden der Änderungen der Seite des Consumer-Place auf die Seite des Coordinator-Place (vgl. Tabelle 4-3) entsteht für die Seite des Coordinator-Place die neue Seitenstruktur (vgl. Abbildung 4-18 untere Hälfte). Dieser Vorgang setzt sich aus den beiden Teilen Synchronisation der Container-Struktur und Synchronisation der Portlets zusammen.

					Phase 1		
Coordinator-Place			Consumer-Place		Fall	Aktion	Aktion
Portalelement	Änderungszeit		Änderungszeit	Portalelement			
Synchronisation der Container							
Z_C	T ₁	-	---	---	3	Keine Veränderung	-
S_E	T ₁	-	---	---	3	Keine Veränderung	-
Z_A	T ₁	>	T ₀	Z_A	2 a)	Keine Veränderung	-
S_A	T ₀	<	T ₂	S_A	2 b)	Modifizieren	Position 2
Z_B	T ₀	<	T ₂	Z_B (gelöscht)	6	Löschen	-
S_C	T ₀	<	T ₂	S_C (gelöscht)	6	Löschen	-
S_D	T ₀	<	T ₂	S_D (gelöscht)	6	Löschen	-
S_B	T ₀	<	T ₂	S_B	2 b)	Modifizieren	Position 1
Synchronisation der Portlets							
P 4	T ₁	-	---	---	3	Keine Veränderung	-
P 3	T ₁	>	T ₀	P 3	2 a)	Keine Veränderung	-
P 2	T ₀	<	T ₂	P 2	2 b)	Modifizieren	S_A - Position 1
P 1 (gelöscht)	T ₃	>	T ₂	P 1	5	Keine Veränderung	-
---	---	-	T ₂	P 5	4	Hinzufügen	S_B - Position 1

Tabelle 4-3: Phase 1 der Synchronisation der Struktur einer Seite

In der zweiten Phase wird der neue verbindliche Zustand der Seite im Coordinator-Place auf die Seite im Consumer-Place angewendet (vgl. Tabelle 4-4).

					Phase 2		
Consumer-Place			Coordinator-Place		Fall	Aktion	Aktion
Portalelement	Änderungszeit		Änderungszeit	Portalelement			
Synchronisation der Container							
Z_A	T ₀	<	T ₁	Z_A	2 b)	Modifizieren	Position 2
S_B	T ₂	=	T ₂	S_B	1	Keine Veränderung	-
S_A	T ₂	=	T ₂	S_A	1	Keine Veränderung	-
---	---	-	T ₁	Z_C	3	Hinzufügen	Position 1
---	---	-	T ₁	S_E	3	Hinzufügen	Z_C - Position 1
Synchronisation der Portlets							
P 3	T ₀	<	T ₁	P 3	2 b)	Modifizieren	S_E - Position 2
P 5	T ₂	<	T ₃	P 5	2 b)	Modifizieren	S_B - Position 1
P 2	T ₂	=	T ₂	P 2	1	Keine Veränderung	-
P 1	T ₂	<	T ₃	P 1 (gelöscht)	5	Löschen	-
---	---	-	T ₁	P 4	3	Hinzufügen	S_E - Position 1

Tabelle 4-4: Phase 2 der Synchronisation der Struktur einer Seite

Abbildung 4-19 zeigt die Struktur der zweiten Phase der Synchronisation und den Zustand nach deren Abschluss, bei dem erneut aus bereits dargestellten Gründen die Löschmarkierungen nur für die Seite des Coordinator-Place vorgehalten werden.

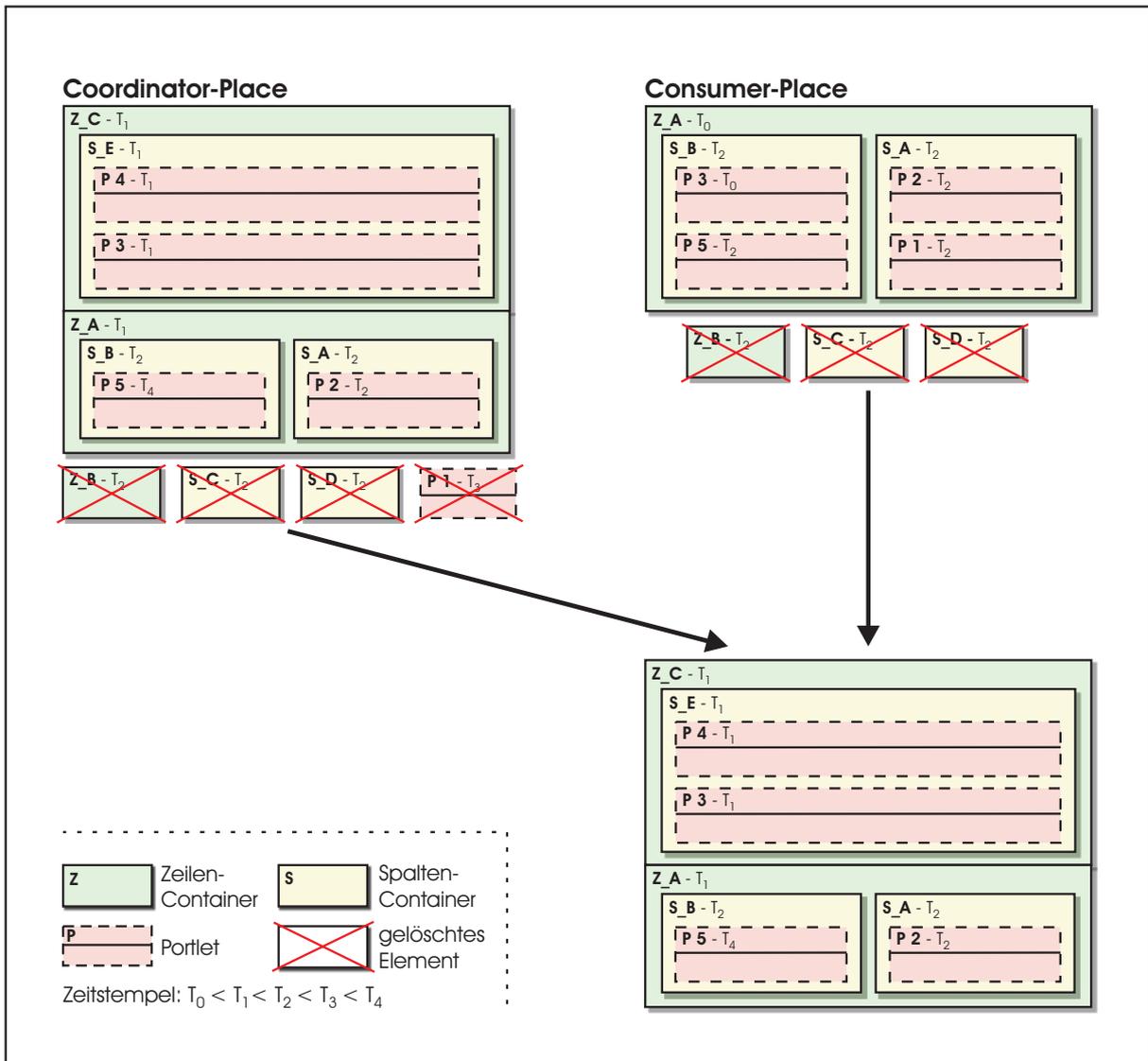


Abbildung 4-19: Struktur der Seiten nach Abschluss der zweiphasigen Synchronisation

4.5.4.3 Konfliktmanagement

Konflikte treten auf, wenn zwischen zwei Synchronisationen sich überschneidende oder in Widerspruch zueinander stehende Modifikationen an den Eigenschaften eines Platzes, dessen Seiten oder Portlets sowohl am Coordinator-Place als auch an einem oder mehreren Consumer-Places vorgenommen werden. Dies sollte grundsätzlich durch Absprachen und eine entsprechende Steuerung der Rechte auf dem gemeinsam genutzten Platz vermieden werden. Oftmals sind Konflikte daher ein Zeichen organisatorischer Defizite. Treten sie trotzdem auf, sind sie weitgehend informationserhaltend, automatisch, d. h. ohne manuellen Eingriff aufzulösen, so dass durch die Synchronisation immer ein konsistenter Zustand erreicht wird.

Als verbindliche Basisregel bei der Synchronisation wird vorgeschlagen, immer die Änderung zu übernehmen, die zuletzt vorgenommen wurde. Diese Regel ist hierarchisch auf alle Grund-

elemente von den Eigenschaften des Platzes bis zur Anordnung der Portlets einzeln anzuwenden. Die Abarbeitung hat sequenziell von links nach rechts und bei zweidimensionalen Strukturen zusätzlich von oben nach unten zu erfolgen. So sind Konflikte, wie z. B. sich widersprechende Namen eines Platzes oder einer Seite, auflösbar. Komplexere Konflikte (s. u.) machen zu deren weitgehend informationserhaltender Auflösung weitere Regeln notwendig. Informationserhaltend bedeutet in diesem Zusammenhang, dass der Verlust von Änderungen so gering wie möglich ist. Im Einzelfall wird dies in den weiteren Ausführungen präzisiert. Es kann jedoch nicht garantiert werden, dass das Ergebnis der automatischen Konfliktauflösung dem von allen Beteiligten intendierten Zustand entspricht, so dass eine weitere manuelle Korrektur durch berechtigte Benutzer notwendig sein kann.

Im Folgenden werden beispielhafte Szenarien vorgestellt, die zu Konflikten auf den verschiedenen Ebenen vom Seitenregister, über die Seiten mit Zeilen- und Spalten-Containern bis zu den Portlets führen. Anhand von Beispielen werden die Regeln zur Auflösung der möglichen Konflikte erläutert.

4.5.4.3.1 Seitenregister

Bei der Synchronisation des Seitenregisters können zwei unterschiedliche Arten von Konflikten auftreten. Beim so genannten *Modifikationskonflikt* werden durch das Modifizieren oder Hinzufügen von Seiten einzelne Positionen mehrfach belegt. Die zweite Konfliktart wird als *Löschkonflikt* bezeichnet. Sie entsteht, wenn vor der Synchronisation eine Seite im Coordinator- oder Consumer-Place gelöscht, aber zeitlich nach dem Löschen in dem jeweils anderen Platz dieselbe Seite verändert wird. Durch das Auftreten von Konflikten ist es möglich, dass die Positionierung der Seiten nicht mit der ersten Position beginnt oder Lücken aufweist. Dies ist unzulässig und muss mithilfe der folgenden Regeln korrigiert werden.

- *Regeln für die Behandlung von Modifikationskonflikten:* Sind Veränderungen derart vorgenommen worden, dass zwei Seiten die gleiche Position innerhalb des Seitenregisters beanspruchen, dann erhält die Seite die entsprechende Position, die zuletzt geändert wurde. Die unterlegene Seite wird hinter alle anderen regulären Seiten verschoben. Das heißt, Seiten, die bei einem Konflikt unterlegen sind, werden immer hinter den Seiten angeordnet, die entweder bei einem Konflikt gewonnen haben oder bei denen kein Konflikt aufgetreten ist. Treten mehrere Konflikte auf, so sind die unterlegenen Seiten in der Reihenfolge des Auftretens der Konflikte hintereinander anzuordnen.

- *Regeln für die Behandlung von Löschkonflikten:* Sind Veränderungen derart vorgenommen worden, dass eine beim Coordinator-Place gelöschte Seite zeitlich nach dieser Löschung, aber vor der nächsten Synchronisation beim Consumer-Place verändert wurde oder umgekehrt, dann ist die Löschung rückgängig zu machen und der Aufbau der veränderten Seite zu übernehmen. Dies betrifft sowohl die Bezeichnung und die Position als auch die Struktur der Seite. Die Auflösung eines Löschkonflikts kann gleichzeitig zu einem Modifikationskonflikt führen, der wie oben beschrieben aufzulösen ist.

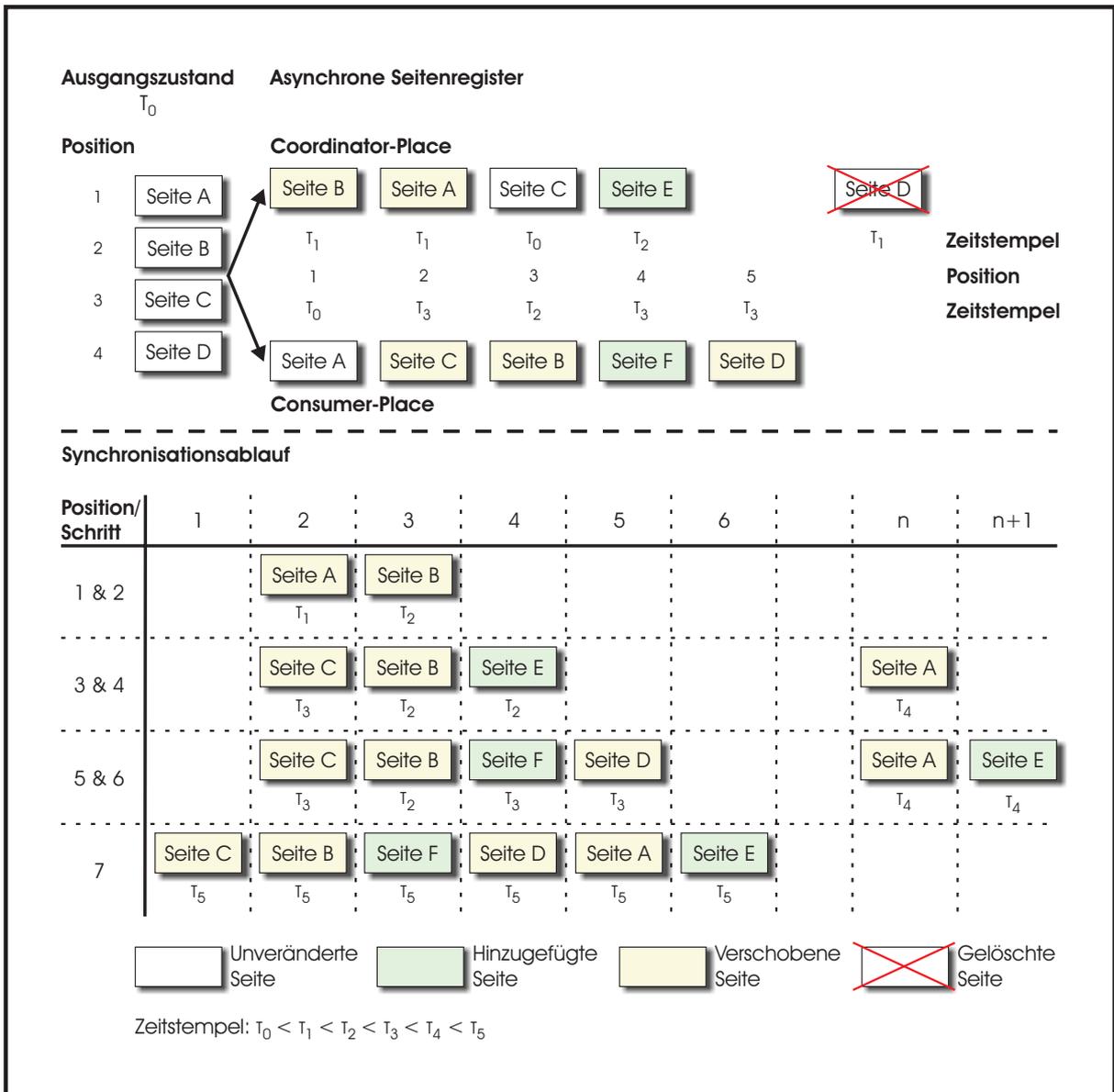


Abbildung 4-20: Synchronisation des Seitenregisters mit Konflikten

Das in Abbildung 4-20 dargestellte Beispiel zeigt sowohl einen Löschkonflikt als auch mehrere Modifikationskonflikte und deren Auflösung. Im oberen Bereich ist das konsistente Seitenregister nach einer Synchronisation und die durch sich überschneidende Veränderungen entstandenen, nicht mehr synchronen Seitenregister des Coordinator- und Consumer-Place dar-

gestellt. Im unteren Bereich ist der Synchronisationsablauf zur Herstellung des korrekten Zustands für das Seitenregister des Coordinator-Place schematisch wiedergegeben.

Im Schritt 1 und 2 werden die Positionsdaten der Seiten übernommen, die zuletzt geändert wurden. Dies erfolgt ebenfalls in allen weiteren Schritten, in Schritt 3 wird jedoch festgestellt, dass ein Modifikationskonflikt vorliegt. Dieser wird derart aufgelöst, dass die zuletzt geänderte Seite C die Position 2 einnimmt und die Seite A an das Ende verschoben wird. Im Schritt 5 wird ein Löschkonflikt erkannt und es tritt erneut ein Modifikationskonflikt auf. Die Löschung wird überschrieben, und durch Verschieben der Seite E hinter die Seite A wird der Modifikationskonflikt aufgelöst. Zum Schluss werden in Schritt 7 die Positionen angepasst, so dass die Seiten wie gefordert mit der ersten Position beginnen und keine Lücken aufweisen.

4.5.4.3.2 Layout von Seiten

Im Fall der Synchronisation der Struktur aus Zeilen- und Spalten-Containern einer Seite können dieselben Arten von Konflikten auftreten, wie dies bei der Synchronisation des Seitenregisters der Fall ist. Sie sind ebenfalls auf die gleichen Ursachen zurückzuführen. Trotz der geschachtelten Struktur des Layouts einer Seite kann die im vorherigen Abschnitt beschriebene Konfliktauflösung grundsätzlich beibehalten werden, ist aber für Löschkonflikte um die folgende Ausnahme zu ergänzen.

Veränderungen eines Containers im Consumer-Place (Coordinator-Place), die nach der Löschung des Containers im Coordinator-Place (Consumer-Place), aber vor einer Synchronisation erfolgen, führen dazu, dass der Container mit den veränderten Eigenschaften wieder hergestellt wird. Dies bleibt die Grundregel, um Löschkonflikte aufzulösen. Durch die verschachtelte Struktur des Layouts ist es möglich, dass ein Container wieder herzustellen ist, dessen ihn enthaltender Container nicht mehr vorhanden ist. In dem beschriebenen Fall muss der Container gelöscht bleiben. Diese Vorgehensweise resultiert aus der hierarchischen Anwendung der Grundregel auf alle Elemente ausgehend vom obersten Zeilen-Container, die in diesem Fall keine Wiederherstellung des übergeordneten Containers zulässt. Eine ähnliche Situation tritt ein, wenn ein Container in einen Container einzufügen ist, der nach der Synchronisation nicht mehr existent ist. Hier muss das Hinzufügen unterlassen werden. Die Auflösung eines Löschkonflikts kann auch im Fall der Synchronisation der Struktur der Seite gleichzeitig zu einem Modifikationskonflikt führen, der wie zuvor beschrieben aufzulösen ist.

Abbildung 4-21 zeigt ein Beispiel zur Veranschaulichung möglicher Konflikte sowohl beim Layout der Seite als auch unter Einschluss der im folgenden Abschnitt dargestellten Konflikte bei der Berücksichtigung von Portlets. Der obere Teil der Abbildung stellt den Ausgangszustand der synchronen Seite im Coordinator- und Consumer-Place dar. Durch sich widersprechende Modifikationen entstehen die zwei in Konflikt zueinander stehenden Repräsentationen der Seite.

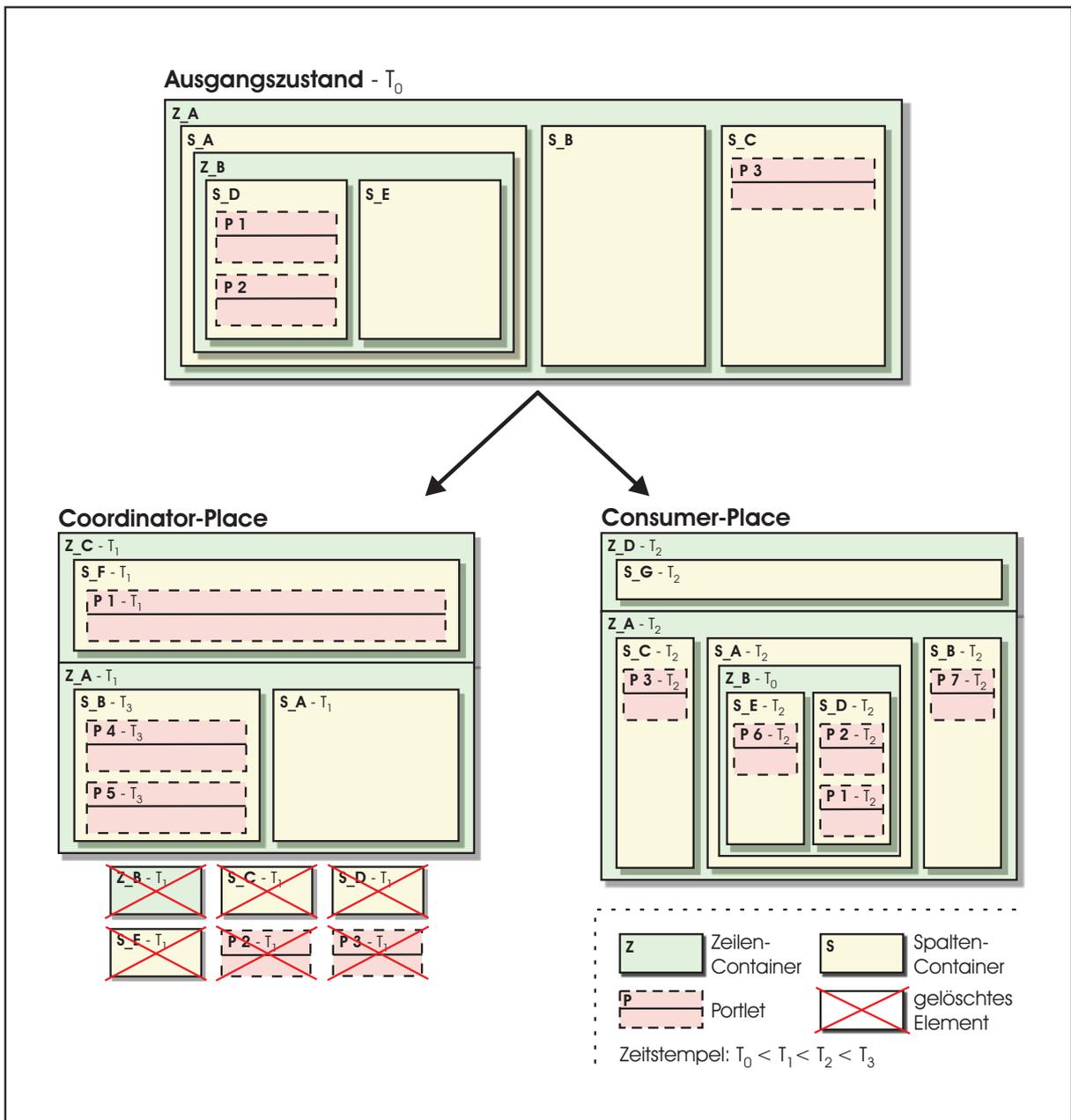


Abbildung 4-21: Ausgangssituation bei Konflikten in der Struktur einer Seite

Durch Auflösung der auftretenden Layoutkonflikte entsteht das neue verbindliche Layout der Abbildung 4-22. Der Modifikationskonflikt, der durch das Hinzufügen zweier neuer Zeilen-Container an erster Stelle entsteht, wird derart gelöst, dass der zuletzt eingefügte Container

Z_D die erste Position erhält und der unterlegene Container Z_C ans Ende verschoben wird. Dies entspricht ebenso der Standardstrategie wie die Wiederherstellung von Container S_C und der Auflösung des dadurch entstehenden Modifikationskonflikts um die erste Position im Zeilen-Container Z_A. Die erweiterte Konfliktbehandlung wird für den Löschkonflikt der Container S_E und S_D notwendig. Diese wurden gegenüber der Löschung in der Coordinator-Seite zu einem späteren Zeitpunkt in der Seite des Consumers verändert, was eine Wiederherstellung erfordern würde. Der sie enthaltende Zeilen-Container Z_B ist und bleibt gelöscht, so dass die beiden Spalten-Container nicht wieder hergestellt werden können. Die damit verbundenen Auswirkungen auf die enthaltenen Portlets sind u. a. Gegenstand der Betrachtung im folgenden Unterkapitel.

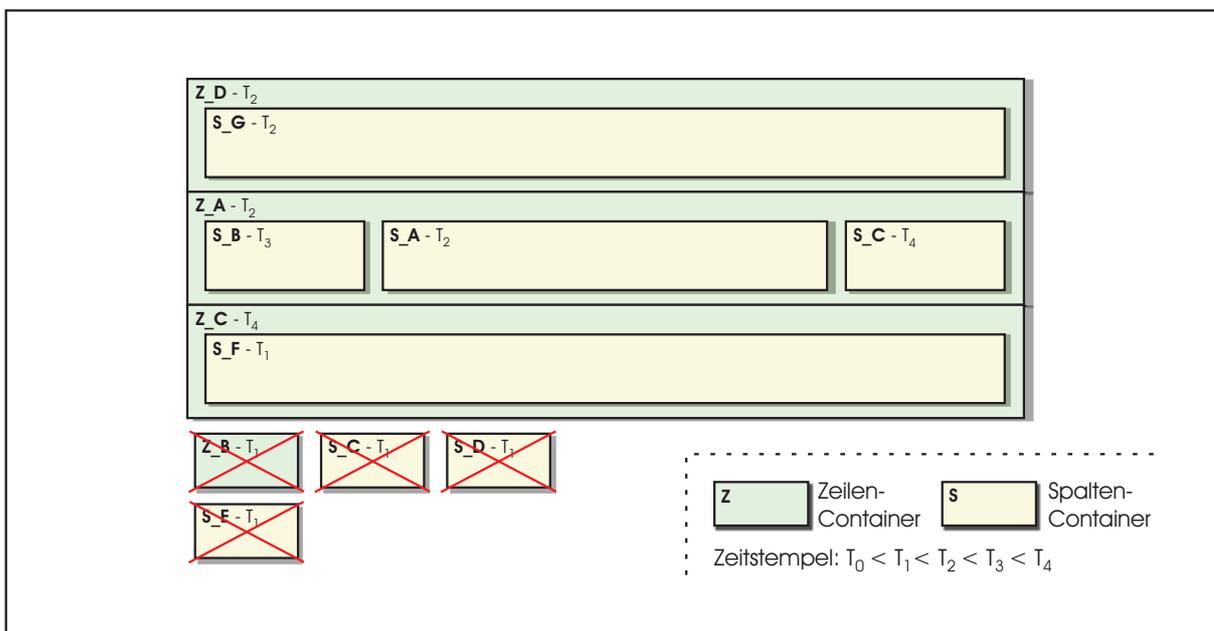


Abbildung 4-22: Layout der Seite nach Auflösung der Konflikte beim Coordinator-Place

4.5.4.3.3 Portlets

Die Synchronisation der Portlets einer Seite kann erst erfolgen, nachdem das neue verbindliche Layout berechnet wurde. Auch für Portlets sind infolge derselben Ursachen Modifikations- und Löschkonflikte möglich, die über die entsprechenden Standardstrategien aufgelöst werden können. Da Portlets aber immer einem sie enthaltenden Spalten-Container zugeordnet sind, muss die Konfliktauflösung für die Synchronisation von Portlets in Abhängigkeit von dem synchronisierten Layout erfolgen.

Sowohl bei Modifikations- als auch bei Löschkonflikten kann die Situation auftreten, dass der Spalten-Container, in dem sich das Portlet nach Auflösung des Konflikts befinden würde, durch die Synchronisation des Layouts der Seite nicht mehr vorhanden ist. Führt die ähnliche

Situation im Fall der Synchronisation des Layouts dazu, dass Container nicht wieder hergestellt oder eingefügt wurden, ist dies für die Synchronisation von Portlets unzureichend. Bei Portlets – als den eigentlichen Trägern der Informationen (vgl. Abschnitt 4.2.2), deren Auswahl und Konfiguration einen wesentlichen Aufwand, aber auch Wert bedeuten – ist es nicht hinnehmbar, dass diese durch unbeabsichtigte Wechselwirkungen mit der Synchronisation des Layouts der Seite verloren gehen.

Um dies zu vermeiden, sind Portlets an die letzte Position innerhalb des letzten Spalten-Containers des verbindlichen Layouts der Seite einzufügen, wenn die sie nach der Synchronisation enthaltenden Spalten-Container nicht mehr vorhanden sind. Tritt diese Situation für mehrere Portlets ein, so sind sie in der Reihenfolge des Auftretens jeweils als letztes Portlet in den letzten Spalten-Container einzufügen. Diese Strategie zur Konfliktauflösung erfordert in den meisten Fällen die manuelle Korrektur durch einen berechtigten Benutzer, sorgt aber für eine maximale Informationserhaltung, weil keine Portlets verloren gehen.

Abbildung 4-23 visualisiert das Ergebnis der Synchronisation einer Seite für die Phase 1 zwischen Coordinator- und Consumer-Place unter Einschluss der Betrachtung von Portlets. Der Modifikationskonflikt, der durch das Einfügen der Portlets P4, P5 und P7 in den Spalten-Container S_B entstanden ist, und der Löschkonflikt im Zusammenhang mit Portlet P3, werden nach der Standardstrategie aufgelöst. Der Modifikationskonflikt des Portlets P1, der Löschkonflikt des Portlets P2 und das Einfügen des Portlets P6 erfordern den Einsatz der erweiterten Strategie. Obwohl die jeweiligen Spalten-Container, in denen die Portlets enthalten sein sollten, nicht mehr vorhanden sind, gehen diese jedoch nicht verloren, sondern werden in der Reihenfolge des Auftretens der Konflikte in den letzten Spalten-Container der Seite (S_F) eingefügt.

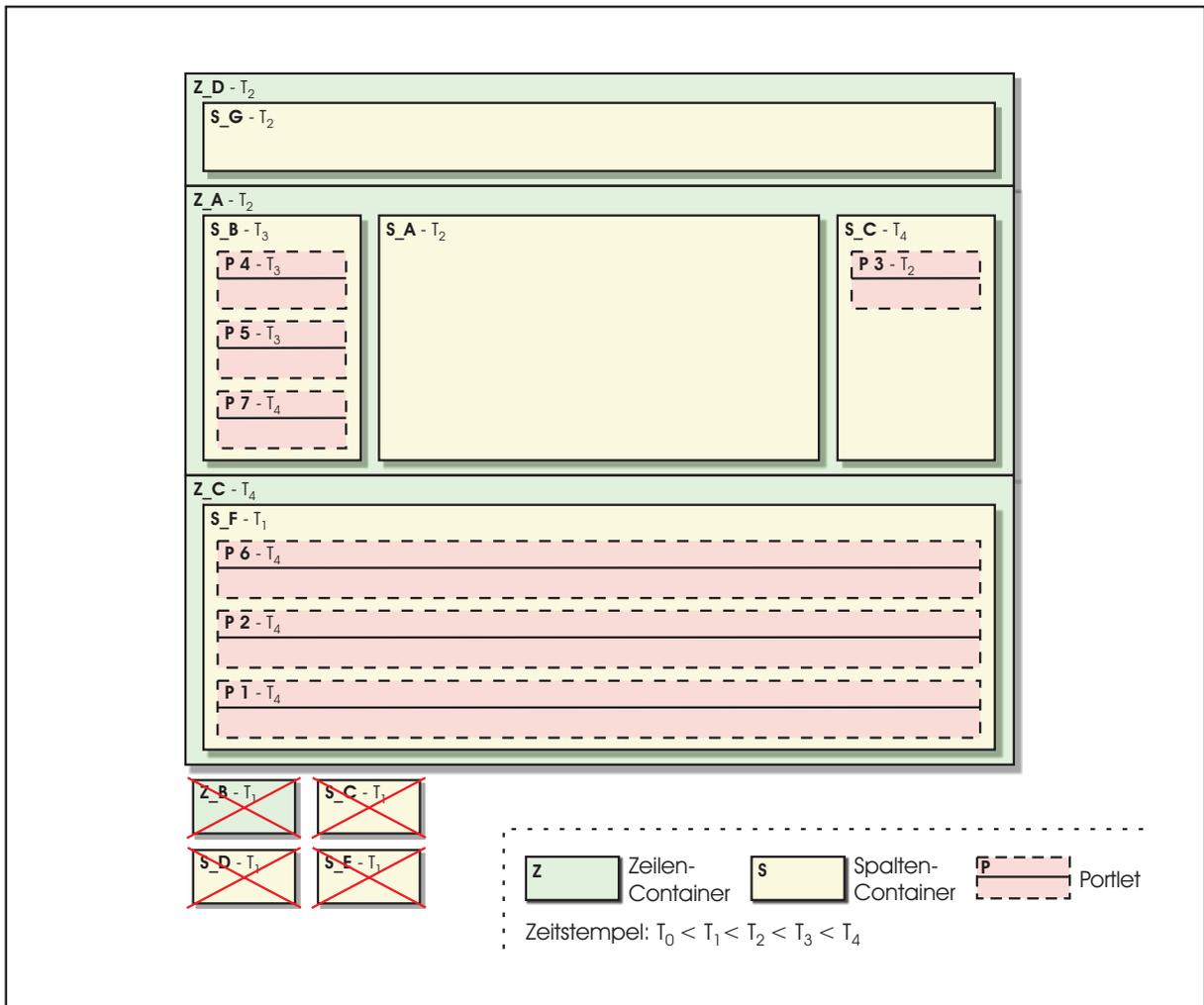


Abbildung 4-23: Repräsentation der Seite nach Auflösung der Konflikte beim Coordinator-Place

4.5.4.4 Zeitstempel zur Minimierung der Synchronisationsdaten

Die von dem vorgeschlagenen Synchronisationsmodell zur Identifikation der letzten Änderung des Portalelements verwendeten Zeitstempel bieten die Möglichkeit, das Datenaufkommen einer Synchronisation zu minimieren. Dies ermöglicht die Reduktion benötigter Bandbreiten und beschleunigt allgemein den Vorgang der Synchronisation.

Das Synchronisationsmodell schreibt, um die Entwurfsautonomie der Portalimplementierungen nicht weiter zu beschneiden, nicht vor, dass die Zeitstempel zu diesem Zweck eingesetzt und vorgehalten werden müssen. Allerdings legt es eine entsprechende Verfahrensweise fest, welche die verbindliche Interpretation der ausgetauschten Zeitstempel definiert. So ist es ohne spezielle Abstimmungs- und Konfigurationszyklen möglich, dass Portale gekoppelt werden können, bei denen eines die im Folgenden beschriebene Verwendung von Zeitstempeln unterstützt, das andere aber nicht.

Die Minimierung der Synchronisationsdaten wird auf zwei Ebenen mit jeweils zwei Ausprägungen empfohlen. Es steht den Implementierungen offen, evtl. auch nur einzelne oder ausgewählte Ebenen und Ausprägungen zu implementieren.

- Die erste Ebene umfasst den Platz selbst. Beim Consumer-Place kommt einerseits der Zeitstempel der letzten Änderung des Platzes zur Anwendung (schließt jede Form der Änderung eines in ihm enthaltenen Portalelements ein), andererseits der Zeitpunkt der letzten Veränderung des Coordinator-Place, der von der letzten Synchronisation mit diesem übermittelt wurde. Durch den Abgleich kann festgestellt werden, ob beim Consumer-Place eine zu übertragende Veränderung stattgefunden hat. Ist dies nicht der Fall, so wird ausschließlich dies dem Coordinator-Place mitgeteilt und es müssen keine weiteren Daten über die Eigenschaften des Platzes, dessen Seiten, Container und Portlets gesendet werden. Eine Variante ist, dass nur die Eigenschaften des Platzes übertragen werden, wenn das Seitenregister und seine Bestandteile nicht verändert wurden.

In der zweiten Phase, der Antwort des Coordinator-Place, sind die Regeln analog anwendbar. Dies hat jedoch unter Berücksichtigung des Ergebnisses der ersten Phase der Synchronisation zu erfolgen. Wurden in dieser Phase Veränderungen bis auf die Ebene eines Portalelements vorgenommen, dann sind diese verpflichtend mindestens bis zur gleichen Ebene zurückzuübermitteln, damit ein konsistentes Modell des Platzes erreicht werden kann.

- Auf Ebene des Seitenregisters und der Seiten ist gleichfalls ein Abgleich des Zeitpunktes der letzten lokalen Veränderung jeder Seite beim Consumer-Place mit dem bei der letzten Synchronisation übermittelten Zeitpunkt der Änderung derselben Seite beim Coordinator-Place möglich. Abhängig davon, ob nur die Eigenschaften der Seite oder auch deren Layout verändert wurde, kann auf eine Übertragung der möglicherweise sehr komplexen Struktur des Seitenlayouts verzichtet werden.

Für die Übermittlung des Modells durch den Coordinator-Place in der zweiten Phase der Synchronisation gilt dies unter Berücksichtigung der Ergebnisse der ersten Phase gleichermaßen.

- Eine gesonderte Betrachtung von Portlets erscheint aufgrund der engen Verzahnung mit dem Layout der Seite nicht sinnvoll und ist deshalb nicht Bestandteil des Synchronisationsmodells.

Neben der Reduktion der zu übertragenden Datenmenge entfallen ggf. sowohl beim Coordinator- als auch beim Consumer-Place durchzuführende aufwändige Verarbeitungsschritte, um

die beiden Repräsentationen des Modells des Platzes miteinander abzugleichen und zusammenzuführen. Dies wirkt sich positiv auf die Belastung und Antwortzeit der Systeme aus. Der bei der automatischen, zeitgesteuerten Synchronisation am häufigsten zu erwartende Fall ist, dass keine Veränderung stattgefunden hat. Durch den Einsatz der Zeitstempel kann in diesem Fall die maximal zu erzielende Reduktion des Datenaufkommens und der Systemlast erreicht werden. Dies bedeutet außerdem, dass unter Verwendung von Zeitstempeln pro Zeiteinheit eine größere Anzahl von (asynchronen) Synchronisationen durchgeführt werden kann, ohne dass – gegenüber dem Zustand ohne Verwendung der Zeitstempel – das Datenvolumen oder die Systemlast erhöht wird. Positiv zu bewerten ist die durch die Verkürzung des Synchronisationsintervalls erreichte schnellere Synchronität des Platzes, die gleichfalls zur Vermeidung von Konflikten beiträgt.

4.5.4.5 Transparenz zwischen lokalen und Remote-Portlet-Instanzen

Das in Abschnitt 4.5.4.2.3 entwickelte Synchronisationsmodell für gemeinsam genutzte Plätze schließt zwar grundsätzlich die Portlet-Ebene mit ein, berücksichtigt bisher aber keine spezifischen Regeln, die für den Umgang mit Portlet-Instanzen anzuwenden sind. Abweichend von den anderen Elementen – Platz, Seite und Zeilen- sowie Spalten-Container –, die jeweils vollständig lokal erzeugt, gelöscht, wieder hergestellt und vorgehalten werden können, kann die eigentliche Portlet-Instanz, wie in Abschnitt 4.4.2.1 dargestellt, immer nur von einem Portal, dem Portlet-Provider, zur Verfügung gestellt werden.

Bei der Synchronisation der Meta-Daten der Portlet-Instanzen sind drei Fälle zu unterscheiden:

- Das Portlet steht lokal zur Verfügung.
- Es handelt sich um ein Portlet, das von einem Portlet-Provider angeboten wird und auf das direkt oder indirekt zugegriffen werden kann.
- Es handelt sich um ein Portlet, das von einem Portlet-Provider angeboten wird; auf das Portlet kann aber weder direkt noch indirekt zugegriffen werden.

Das folgende Unterkapitel behandelt den ersten und zweiten Fall. Anschließend wird eine Erweiterung, die so genannte Portlet-Ersetzung, eingeführt. Die Behandlung des dritten Falls der nicht verfügbaren Portlets schließt die Ausführungen zu diesem Thema ab.

4.5.4.5.1 Standardverarbeitung

Jedes Portal, das an einer Portalkopplung teilnimmt, die auf dem in dieser Arbeit entwickelten Konzept basiert, ist über einen eindeutigen Schlüssel identifizierbar (vgl. Abschnitt 4.5.2.2). Dieser Schlüssel ist selbst Teil eines Schlüssels, der eine Portlet-Definition eindeutig kennzeichnet und bei der Synchronisation mit übertragen wird. Durch Vergleich der eigenen Portal-ID mit der im Schlüssel der Portlet-Definition enthaltenen kann eindeutig festgestellt werden, ob das Portal selbst als Portlet-Provider für dieses Portlet fungiert oder nicht.

Handelt es sich um ein lokal zur Verfügung gestelltes Portlet, dann sind alle Operationen unverändert lokal durchzuführen. Portlets, die von einem kooperierenden Portal zur Verfügung gestellt werden, müssen lokal durch so genannte Remote-Portlets ersetzt werden. Die lokale Instanz des Remote-Portlets dient als Verweis auf eine im Portlet-Provider zur Verfügung gestellte Portlet-Instanz. Operationen, die auf der lokalen Portlet-Instanz des Remote-Portlets durchgeführt werden, sind direkt oder indirekt zum Portlet-Provider weiterzuleiten, der diese auf der für ihn lokalen Portlet-Instanz ausführt und das Ergebnis an den Aufrufer zurückübermittelt.

Für die Synchronisation relevante Operationen sind das Hinzufügen und damit Erzeugen einer neuen Portlet-Instanz sowie das Löschen und das Wiederherstellen einer vormals gelöschten Portlet-Instanz. Das Verändern der Position von Portlets innerhalb einer Seite hat an dieser Stelle keine Bedeutung, da diese Operation ausschließlich lokal auf der Portlet-Instanz, die als Verweis dient, durchgeführt wird. Im Folgenden sind die genannten drei Operationen unter Berücksichtigung der im Abschnitt 4.5.3 getroffenen Entscheidung für eine spezifische Rollenverteilung daraufhin zu untersuchen, welche Implikationen sich daraus ergeben und welche Regeln entsprechend definiert werden müssen.

Zur besseren Nachvollziehbarkeit der Argumentation wird diese anhand eines Beispiels veranschaulicht, dem die Abbildung 4-24 zugrunde liegt. Es wird angenommen, dass die drei Portale CO (Coordinator-Portal), C1 und C2 (Consumer-Portale) einen Platz gemeinsam nutzen. Jedes Portal stellt jeweils eine lokale Portlet-Definition zur Verfügung, die den anderen Portalen über vorhandene Kommunikationsbeziehungen bekannt ist. Hinzu kommt ein weiteres Portal PR (Provider-Portal), das eine weitere Portlet-Definition zur Verfügung stellt, aber nicht an der gemeinsamen Nutzung des Platzes beteiligt ist.

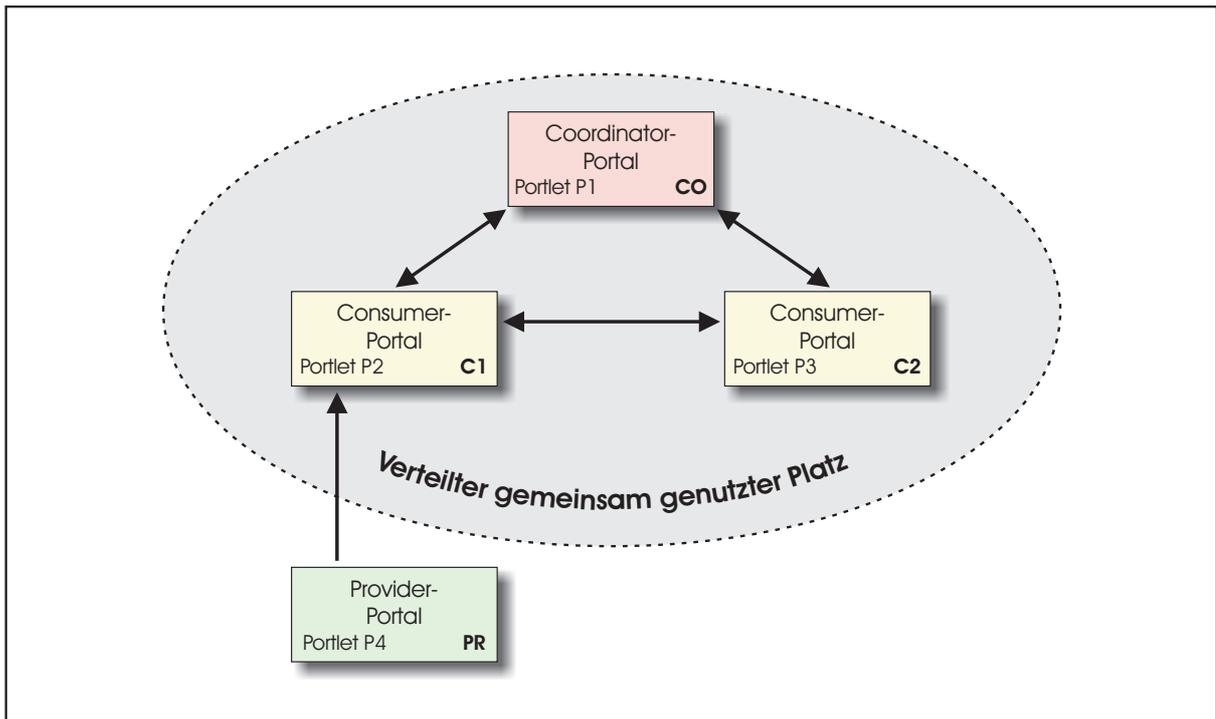


Abbildung 4-24: Beispielszenario für Remote-Portlets in verteilten gemeinsam genutzten Plätzen

Hinzufügen einer Portlet-Instanz

Ein Portlet kann vor einer Synchronisation entweder im Coordinator- oder in einem Consumer-Place hinzugefügt worden sein. Es ist jeweils möglich, dass ein lokal verfügbares (z. B. P1 für CO) oder von einem anderen Portal angebotenes Portlet (z. B. P2 für CO) hinzugefügt wird. Beim Hinzufügen eines nicht lokal verfügbaren Portlets in einen Consumer-Place sind drei Fälle zu unterscheiden:

Es handelt sich um ein Portlet, das

- von dem Portal angeboten wird, das die Rolle des Koordinators für den Platz hat (z. B. P1 für C1);
- von einem Portal angeboten wird, das sich ebenfalls in der Rolle des Consumer-Place befindet (z. B. P2 für C1);
- von einem nicht an der gemeinsamen Nutzung des Platzes teilnehmenden externen Portlet-Provider (z. B. P4 für C1) angeboten wird.

Bestandteil des Modells ist die Festlegung desjenigen Portals, das ein Portlet erstmals in einen gemeinsam genutzten Platz eingebracht hat und welches Portal die Kontrolle über die Referenz-Instanz des Portlets besitzt. Der erste Aspekt ist beim Austritt oder beim Ausschluss eines Portals relevant. Beim Ausschluss werden alle Portlets, die das ausgeschlossene Portal in den gemeinsam genutzten Platz eingebracht hat, aus diesem entfernt. Beim Austritt kann

das austretende Portal selbst entscheiden, ob die von ihm eingebrachten Portlets gelöscht oder weiterhin enthalten bleiben sollen. Nach dem Hinzufügen eines Portlets ist die Eigenschaft, die das das Portlet erstmalig einbringende Portal identifiziert, unveränderlich. Der zweite Aspekt ist für die endgültige Löschung des Referenz-Portlets von Bedeutung. Diese ist nur durch das Portal zulässig, das die Kontrolle über das Referenz-Portlet ausübt. Entsprechend den Erläuterungen in Abschnitt 4.5.2.3 ist dies bis zur Durchführung einer Synchronisation die Aufgabe des Portals, das das Portlet eingebracht hat. Nach der Synchronisation muss die Aufgabe an das Portal übertragen worden sein, das als Koordinator für den Platz fungiert. Die Eigenschaft ist daher maximal einmal änderbar. Die Änderung kann abhängig von der Situation entweder gar nicht auftreten (das Portlet wurde vom Coordinator-Portal eingebracht), sich auf das lokale Portal beschränken (das Portlet wurde als lokales Portlet von einem Consumer-Portal eingebracht) oder aber ein entferntes Portal einbeziehen (das Portlet wurde als entferntes Portlet von einem Consumer-Portal eingebracht).

Ein als hinzugefügt erkanntes Portlet ist in die eigene Repräsentation des Modells des Platzes aufzunehmen, in dem das Referenz-Portlet geklont wird. Fungiert das Portal selbst als Portlet-Provider, handelt es sich um eine lokale Operation, andernfalls um eine entfernte. Im Fall der entfernten Operation ist zusätzlich lokal ein Remote-Portlet als Platzhalter und Referenz auf das entfernte Referenz-Portlet anzulegen.

Löschen einer Portlet-Instanz

Das explizite Löschen einer Portlet-Instanz kann entweder im Coordinator- oder in einem Consumer-Place erfolgen. Unterscheiden lässt sich zum einen, ob das Portal, das das Portlet löscht, die Kontrolle über das Referenz-Portlet ausübt oder ob diese bei einem anderen Portal liegt. Zum anderen wird unterschieden, ob es sich um ein – bezogen auf die Löschung – lokales oder entferntes Portlet handelt. Die Löschung des Referenz-Portlets ist nur durch das Portal möglich, das die Kontrolle über dieses ausübt. Andernfalls wird ausschließlich die eigene Referenz auf das Referenz-Portlet gelöscht. Handelt es sich um ein lokales Portlet, ist dieses Vorgehen ausreichend. Bei einem als Platzhalter dienenden Remote-Portlet ist zusätzlich der entfernte Klon des Referenz-Portlets zu löschen.

Die Kontrolle kann, wie im vorherigen Abschnitt dargestellt, bei einem Consumer-Place liegen, der das Portlet in den Platz eingebracht hat, bei dem aber noch keine Synchronisation mit dem Coordinator-Place stattgefunden hat. In diesem Fall kann sowohl die eigene Referenz auf das Referenz-Portlet ggf. mit den erzeugten Klonen als auch das Referenz-Portlet selbst gelöscht werden. Andernfalls übt die Kontrolle der Coordinator-Place aus, der die Löschung

des Referenz-Portlets erst nach Ablauf einer Zeitspanne vornimmt, in der die eigene Referenz auf das Referenz-Portlet als Löschmarkierung vorgehalten wurde. Dieses Vorgehen macht ein Wiederherstellen des Portlets bei einem Konflikt möglich.

Wiederherstellen einer Portlet-Instanz

Die Wiederherstellung einer bereits gelöschten Portlet-Instanz ist nur durch den Coordinator-Place möglich und notwendig. Sie kann erfolgen, solange der Coordinator-Place das Referenz-Portlet, über das er die Kontrolle ausübt, nicht gelöscht hat, sondern seine Referenz – als lokales oder Remote-Portlet – auf das Referenz-Portlet weiterhin als Löschmarkierung vorhält.

Die deklarativ beschriebenen Operationen und mit ihnen verbundenen Semantiken sind verbindlich umzusetzen, machen jedoch keine Aussagen über konkrete Realisierungen kompatibler Portal-Frameworks. Die Entwurfsautonomie bleibt dadurch so weit wie möglich erhalten.

4.5.4.5.2 Portlet-Ersetzung mittels Portlet-Typen

Die vollständig transparente Behandlung von Portlets – unabhängig davon, ob es sich um lokal zur Verfügung stehende oder von anderen Portlet-Providern angebotene Portlets handelt – schafft die Basis für die gemeinsame Nutzung der durch die Portlets zugreifbaren Informations- und Anwendungssysteme. In Ausnahmefällen kann es aber auch wünschenswert sein, nicht ein konkretes für alle beteiligten Partner gleiches Portlet, sondern nur einen Typ von Portlet in einen Platz einzubringen, der von jedem Partner durch ein eigenes lokales Portlet ersetzt werden kann.

Zwei Beispiele verdeutlichen dieses: Soll ein Portlet die persönliche E-Mail jedes Benutzers zeigen, so ist es nicht zweckmäßig, dass ein Partner ein Portlet einbindet, das dies für seine lokalen Benutzer leistet. Das Portlet würde zwar auch bei den Benutzern der Partner angezeigt werden, jedoch leer bleiben bzw. einen Fehler anzeigen, da das anbietende Portal keinen Zugriff auf die E-Mails externer Benutzer hat. Soll ein Portlet Informationen zu Verfahrensanweisungen anzeigen, die von Partner zu Partner unterschiedlich sind, so sorgt das einfache Hinzufügen eines entsprechenden lokalen Portlets durch einen Partner nur dafür, dass alle Partner die gleichen Anweisungen angezeigt bekommen.

Allgemein lässt sich aus den Beispielen ableiten, dass es sich um Fälle handelt, in denen Informationen oder Anwendungen eingebracht werden sollen, die in dem gemeinsamen Kontext von allen benötigt werden, aber jeweils in der konkreten Ausprägung nur lokal für die

einzelnen Partner zur Verfügung stehen. Um diesem Sachverhalt Rechnung zu tragen, wird das Konzept der *Portlet Type Identifier* in das Synchronisationsmodell eingeführt. Dieses ermöglicht eine Ersetzung von Remote-Portlets durch lokale Portlets vom gleichen Portlet-Typ. Durch die Abstimmung der Partner untereinander wird ein eindeutiger Portlet Type Identifier definiert, der einen spezifischen Typ von Portlet beschreibt. Den Identifier kann jeder Partner selbst einem bei ihm lokal vorhandenen Portlet zuordnen, das diesen Typ repräsentiert.

Das Synchronisationsmodell wird dahingehend erweitert, dass auf der einen Seite der Portlet Type Identifier eines sich in einem gemeinsam genutzten Platz befindlichen Portlets immer mit synchronisiert wird. Auf der anderen Seite wird festgelegt, dass beim Vorhandensein eines Portlet Type Identifiers anstelle von Remote-Portlets lokale, auf der Portlet-Definition mit dem gleichen Portlet Type Identifier basierende Portlets erzeugt und verwendet werden müssen. Der Versuch der Erzeugung eines Remote-Portlets trotz gesetztem Portlet Type Identifier muss einen Fehler und ein Scheitern der Operation zur Folge haben.

Das Konzept der Portlet Type Identifier erweitert die Flexibilität der Portalkopplung. Beim Einsatz des Mechanismus ist aber darauf zu achten, dass der gemeinsame Kontext nicht zu weit eingeschränkt oder vollständig zerstört wird. Der Einsatz der Portlet-Ersetzung sollte daher auf Ausnahmesituationen beschränkt und nicht die Regel sein.

4.5.4.5.3 Nicht verfügbare Portlets

Das Konzept zur Sicherstellung der Transparenz zwischen lokalen und Remote-Portlets setzt voraus, dass alle beteiligten Portale Zugriff auf die jeweiligen Remote-Portlets haben. Gleichmaßen kann eine Ersetzung von Portlets auf der Basis von Portlet-Typen nur durchgeführt werden, wenn die jeweiligen Portale die Portlet-Typen entsprechenden lokalen Portlets zugewiesen haben. Diese Annahmen scheinen zweckdienlich zu sein, da eine sinnvolle Zusammenarbeit nur dann möglich erscheint, wenn die Partner die in einem Platz befindlichen Portlets tatsächlich gemeinsam nutzen können. Für den sicheren Betrieb einer Portalkopplung sind die Annahmen jedoch unzweckmäßig.

Gründe dafür, dass ein Portlet nicht zugreifbar oder kein Portlet mit dem entsprechenden Portlet Type Identifier konfiguriert ist, können Konfigurationsprobleme und intentionale Zugriffsbeschränkungen sein. In keinem Fall ist es hinnehmbar, dass dies zu einer Situation führt, welche die Funktionsfähigkeit der Portalkopplung gefährdet und zu Inkonsistenzen des Synchronisationsmodells führt. Das Modell macht keine Vorschriften darüber, wie Portale

diese Situation intern abbilden, es definiert aber die zur Konsistenzerhaltung notwendigen Regeln.

In einem Portal unbekannte Portlets sind bei allen Operationen so zu berücksichtigen, als wären sie bekannt. Sie werden während der Synchronisation nach den gleichen Regeln behandelt und bleiben vollständig im Modell erhalten. Die interaktive Löschung eines unbekanntes Portlets aus einem gemeinsam genutzten Platz ist zu verhindern. Sie kann ggf. nur durch eine weitere Synchronisation erfolgen, bei der das Portlet in einem Portal gelöscht wurde, dem das Portlet bekannt ist. Es steht den Portalimplementierungen frei, die unbekanntes Portlets den Benutzern oder evtl. ausschließlich den Administratoren anzuzeigen oder diese vollständig zu verbergen, solange diese Bestandteil des Modells bleiben.

Unbekannte Portlets können zu bekannten Portlets werden, indem entweder die Portlet-Definition über eine direkte oder indirekte Kommunikationsverbindung zugreifbar gemacht wird oder indem einer lokalen Portlet-Definition der Portlet Type Identifier zugeordnet wird, der für das unbekanntes Portlet benötigt wird. Gleichmaßen kann ein Portlet zu einem unbekanntes Portlet werden, wenn die Zuordnung des Portlet Type Identifier zu einer Portlet-Definition aufgehoben wird oder im Verlauf kein direkter oder indirekter Zugriff mehr auf die Portlet-Definition besteht.

4.6 Daten- und Funktionsmodell gekoppelter Portale

Neben den Regeln der Kommunikations- und Synchronisationsarchitektur wird ein Daten- und ein Funktionsmodell benötigt. Diese entsprechen der Portal Element Definition Language (PEDL) und der Portal Element Manipulation Language (PEML), die bereits in Abschnitt 4.4.2.2 eingeführt wurden. Sie stellen die Grundlage für die eigentliche Kopplung, die Durchführung der notwendigen Operationen und den Austausch spezifischer Daten zwischen jeweils zwei Portalen dar. Die ausgetauschten Daten selbst sind entsprechend den Regeln der Kommunikations- und Synchronisationsarchitektur (vgl. Abschnitt 4.5) zu interpretieren und zu verarbeiten.

Im Rahmen der vorliegenden Arbeit kann auf diese Aspekte nicht so detailliert eingegangen werden, wie es für eine direkt davon abgeleitete Implementierung notwendig wäre. Stattdessen findet eine Beschränkung auf die weitgehend abstrakte Darstellung der sich auf drei Schichten verteilenden vier funktionalen Bestandteile der Portalkopplung (Makroebene) statt (vgl. Abbildung 4-25). Eine Auflistung der konkreten Datenstrukturen und Funktionen ist im

Anhang (Kapitel 9) wiedergegeben. Dazu wird als Beschreibungssprache die Web Service Description Language (WSDL) verwendet, die in Abschnitt 5.1.2.2 vorgestellt wird.

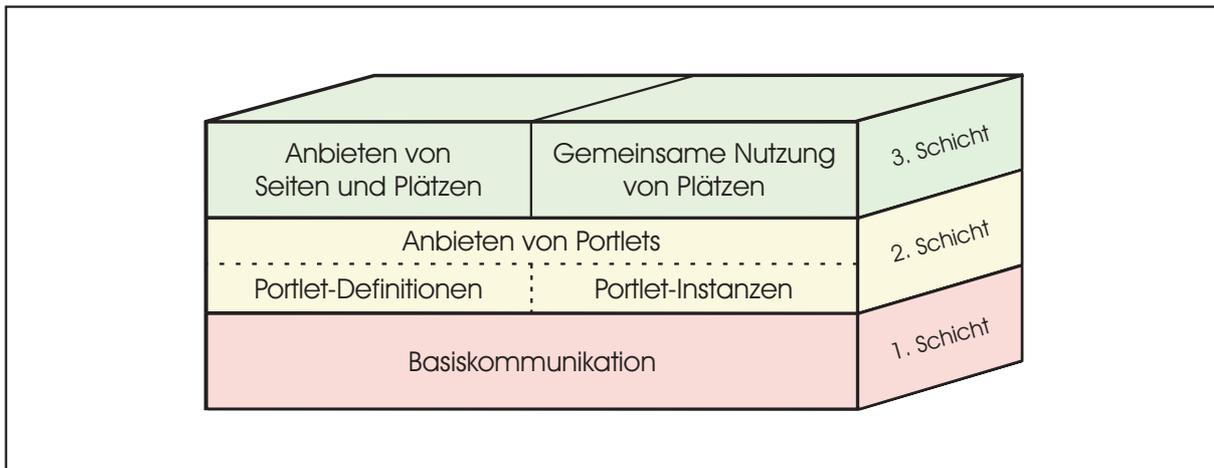


Abbildung 4-25: Schichten des Daten- und Funktionsmodells zur Kopplung von Portalen

4.6.1 Kommunikationsbeziehung zwischen Portalen

Voraussetzung für jede Form der Kommunikation ist die grundsätzliche Etablierung einer Kommunikationsschnittstelle und -beziehung zwischen zwei Portalen. Die hierzu notwendigen Datenstrukturen und Funktionen werden in der untersten Schicht zusammengefasst. Eine Kommunikationsbeziehung beinhaltet die eindeutige Identifikation der Portale und die Festlegung der Endpunkte der Kommunikation. Die gegebene Sicherheitsrelevanz des Aufrufs von Operationen und des Austauschs von Daten macht die Sicherung der Kommunikation und die Authentifizierung der beteiligten Portale notwendig.

Als Servicefunktion kapselt diese Schicht die mit dem Aufbau der Kommunikationsbeziehung und der Authentifizierung verbundenen Aufgaben und stellt diese den höheren Schichten transparent zur Verfügung. Eine schematische Darstellung findet sich in Form eines State-Charts in Abbildung 4-26. Die strikte Trennung der Schichten sorgt für deren Entkopplung. Im konkreten Fall können so unterschiedliche Transportprotokolle und Mechanismen zur Authentifizierung zur Anwendung kommen, ohne dass sich dies auf Funktionen höherer Schichten auswirkt.

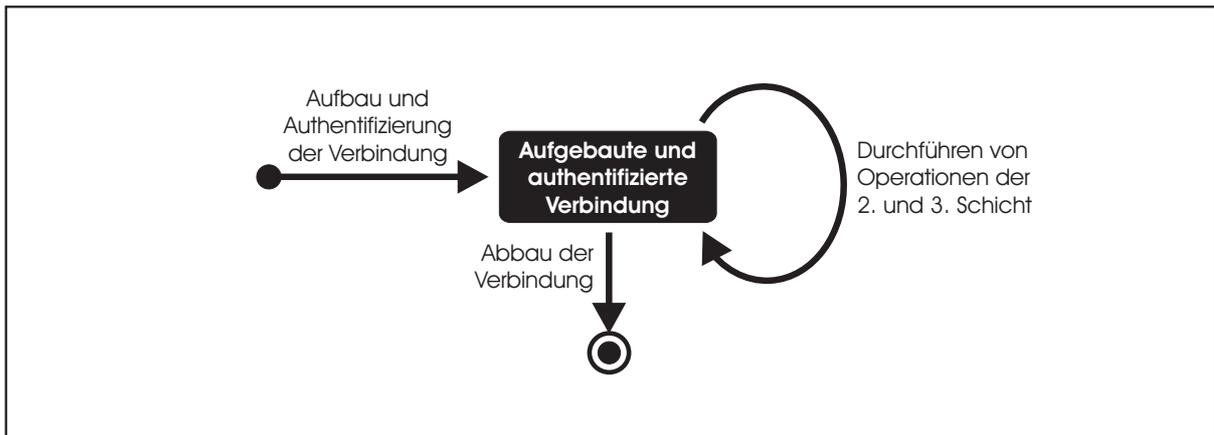


Abbildung 4-26: Operationen zum Management der Kommunikationsbeziehung

4.6.2 Anbieten von Portlets

Die zweite Schicht stellt Datenstrukturen und Funktionen zum Anbieten und für die entfernte Nutzung von Portlets zur Verfügung. Diese ist, wie bereits in Abschnitt 4.4.2.4.1 erläutert, in zwei Bereiche zu unterteilen. Der eine Bereich behandelt die Portlet-Definition, der andere die konkreten Portlet-Instanzen. Die Konzeption der Nutzung von Remote-Portlets ist im weitesten Sinne an die Spezifikation für „Web Service for Remote Portlets“ (WSRP) angelehnt, die zu Beginn des Dissertationsprojekts erst in einer sehr frühen Alpha-Version vorlag. Obwohl die Spezifikation seit Herbst 2003 in der Version 1.0 verfügbar ist (vgl. [Thompson/Leue/Kropp 2003]), wurden von Seiten des Autors der vorliegenden Arbeit wegen des Fehlens einiger benötigter Konzepte vorerst keine Bemühungen zur Angleichung der bereits bestehenden eigenen Modellierung an den WSRP-Standard unternommen wurden.

Im Bereich der Portlet-Definitionen sind Funktionen und Datenstrukturen definiert, die dem Portlet-Consumer bekannt machen, welche Portlets ihm mit welchen Eigenschaften und Funktionen zur Verfügung gestellt werden. Die Datenstrukturen enthalten unter anderem Festlegungen bzgl. der eindeutigen Identifikation einer Portlet-Definition, auch über Portalgrenzen hinweg. Ebenso besteht die Möglichkeit zur Zuordnung der Portlet-Definition zu ggf. einem Intermediär-Portlet-Provider und dem eigentlichen Portlet-Provider. Diese Informationen sind Voraussetzung für die Nutzung der Funktionen im Bereich der Portlet-Instanzen.

Die Funktionen im Bereich der Portlet-Instanzen ermöglichen unter anderem die Erzeugung eines entfernten Portlets, entweder als vollständig neues Portlet oder auf Basis eines bereits bestehenden Portlets als Klon, der optional mit diesem verknüpft bleiben kann. Weitere Funktionen sind zuständig für die eigentliche Nutzung des Portlets, wie die Generierung der Ausgabe oder die Ausführung einer Aktion. Die Verfügbarkeit einer Portlet-Instanz endet mit

deren Löschung, für die ebenfalls eine entsprechende Funktion definiert ist. Im Wesentlichen existiert für jeden Zustandswechsel im Lebenszyklus einer Portlet-Instanz eine entsprechende Funktion mit den zugehörigen Datenstrukturen. Gleichartige Operationen werden zur Vereinfachung der Schnittstelle zusammengefasst. Eine schematische Darstellung findet sich in Form eines State-Charts in Abbildung 4-27.

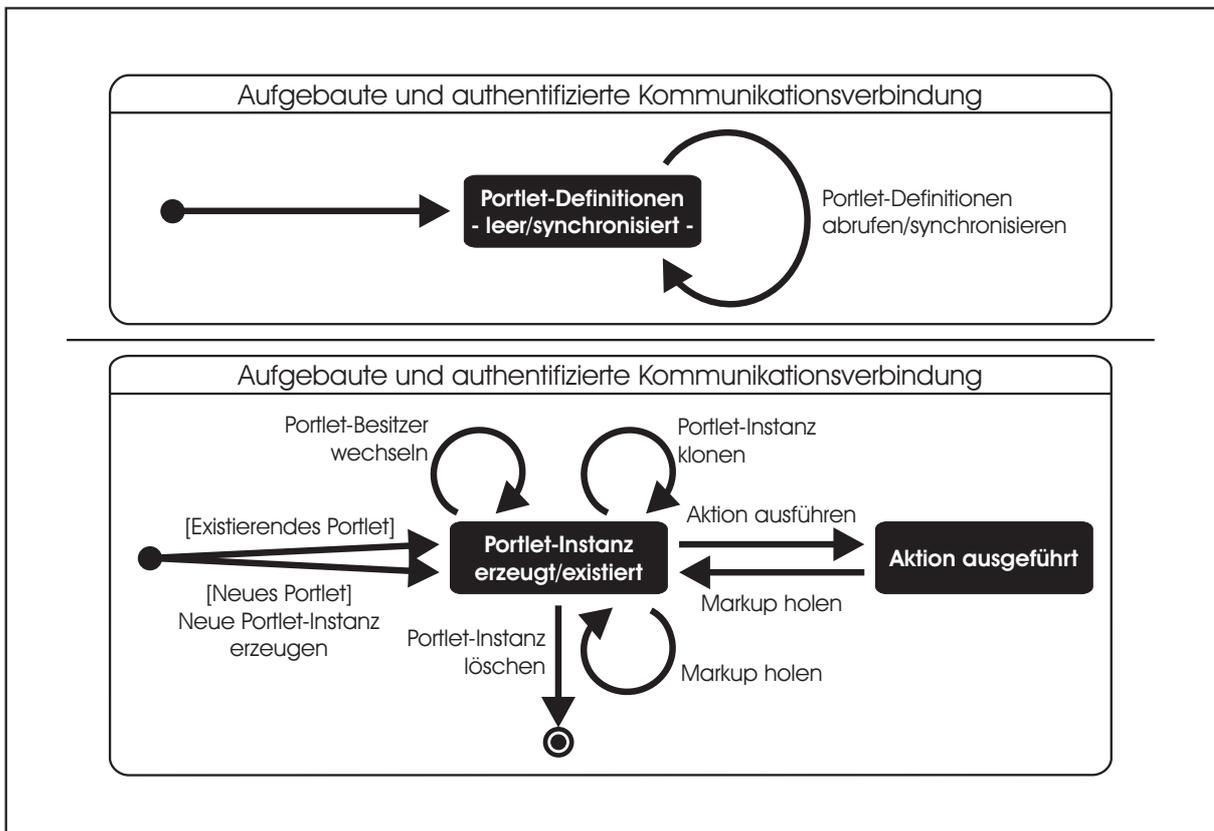


Abbildung 4-27: Operationen zum Abruf von Portlet-Definitionen und zur Nutzung von Portlet-Instanzen

Die Schicht zum Anbieten von Portlets wird auch von der Schicht zum Anbieten von Seiten und Plätzen sowie zur gemeinsamen Nutzung von Plätzen für die Erbringung ihrer erweiterten Aufgaben herangezogen, da in beiden die Einbettung von Portlets stattfindet, die über die entsprechenden Funktionen teilweise als Remote-Portlets genutzt werden müssen.

4.6.3 Anbieten von Seiten

Zum Anbieten von Seiten ist nur eine weitere Funktion notwendig, über die es dem Consumer-Portal möglich ist, alle für ihn zugreifbaren Seiten abzurufen. Die von der Funktion verwendeten Datenstrukturen berücksichtigen die Meta-Daten über den Aufbau der jeweiligen Seite aus Zeilen- und Spalten-Containern sowie die in den Spalten-Containern enthaltenen Portlets. Die Aktion geht immer vom Consumer-Portal aus, das lokal und implementations-

abhängig die Struktur der Seite ablegt und auf die Funktionen der Portlet-Schicht zurückgreift, um entsprechend die Portlets für die Benutzer der Seite zur Verfügung zu stellen. Eine schematische Darstellung findet sich in Form eines State-Charts in Abbildung 4-28.

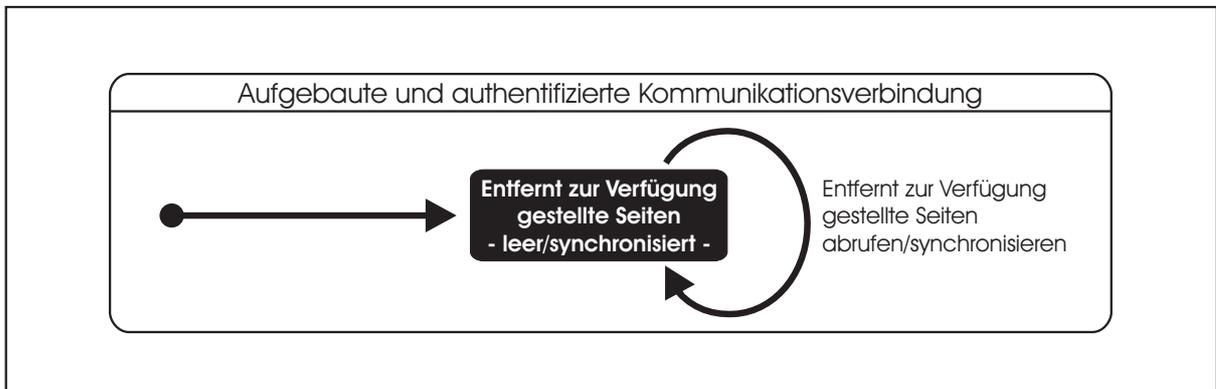


Abbildung 4-28: Operation zum Abruf angebotener Seiten

4.6.4 Gemeinsame Nutzung von Plätzen

Für die gemeinsame Nutzung von Plätzen sind aufgrund der zweiseitigen Beziehung und der dedizierten Rollenverteilung mehrere Funktionen notwendig. Die Datenstrukturen sind gegenüber denen beim Anbieten von Seiten um die Ebene des Platzes selbst zu erweitern, während die Definition einer Seite und der in ihr enthaltenen Elemente mit denen beim Anbieten von Seiten identisch sind und beibehalten werden können. Das Hinzufügen eines weiteren Portals zu den Consumer-Portalen, das durch den Coordinator-Place erfolgt, führt zu einer direkten Kommunikation mit dem neuen Consumer-Portal. Es wird darüber informiert, dass bei ihm ein neuer Platz hinzuzufügen ist. Im Gegensatz zum Anbieten der Seite geht hier initial die Aktion nicht vom Consumer-Portal, sondern vom Coordinator-Portal aus. Entsprechend dem Lebenszyklus bei der gemeinsamen Nutzung eines Platzes (siehe Abbildung 4-12) sind darüber hinaus Funktionen zur eigentlichen Synchronisation und zur Beendigung der gemeinsamen Nutzung des Platzes notwendig. Zur Nutzung der Portlets innerhalb der Seiten wird ebenfalls auf die Funktionen der Portlet-Schicht zurückgegriffen. Eine schematische Darstellung findet sich in Form eines State-Charts in Abbildung 4-29.

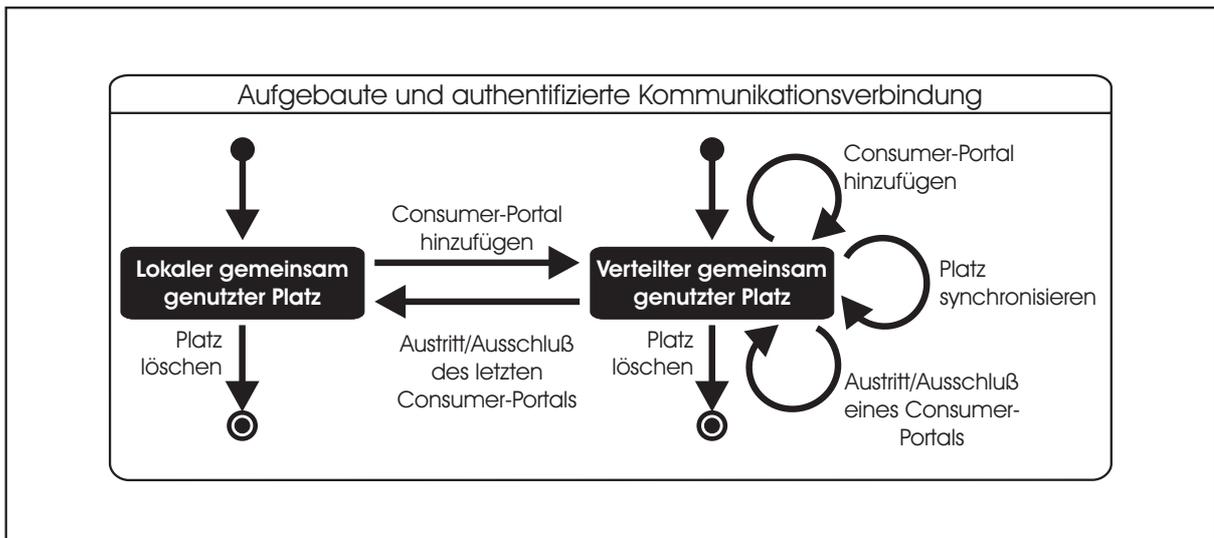


Abbildung 4-29: Operationen zur verteilten gemeinsamen Nutzung von Plätzen

4.7 Zusammenfassung

Mit der Operationalisierung und Erstellung eines Konzepts zur Kopplung von Portalen wurde der erste und wesentlichste Baustein der Zieldefinition dieser Arbeit abgeschlossen (vgl. Abschnitt 2.3.1). Die Operationalisierung hat durch die Einführung des Kontinuums der Portalkopplung und orthogonal dazu durch die Heranziehung der Klassifikationsdimensionen Verteilung, Heterogenität und Autonomie den Raum aufgespannt, in dem das Konzept zu entwickeln ist. Die Definition der betriebswirtschaftlichen und technischen Anforderungen diente der weiteren Orientierung bei der darauf aufbauenden Erstellung einer Kommunikations- und Synchronisationsarchitektur sowie der Modellierung des Daten- und Funktionsmodells.

Die Architektur und das Modell unterstützen alle Stufen des Kontinuums der Portalkopplung und berücksichtigen alle Klassifikationsdimensionen. Sie erreichen so eine weitgehende intra- und interorganisationale Kopplung von Portalen. Aufgrund der hohen Komplexität muss jedoch eine Beschränkung auf die Basiselemente eines Portals vorgenommen werden. In dem Anspruch, Ausgangspunkt für herstellerübergreifende Standardisierungsbemühungen zu sein, müssen teilweise fortgeschrittene Funktionen einzelner Portalimplementierungen zugunsten einer möglichst universellen Ein- und Umsetzbarkeit ausgeblendet werden. Ausführungen zu möglichen Erweiterungen in einer späteren Version finden sich im Ausblick in Abschnitt 7.1.

5 Umsetzung des Konzepts gekoppelter Portale am Beispiel des Workplace Portals G8

Das im vorangegangenen Kapitel 4 dargestellte abstrakte Konzept ist im Folgenden durch eine informationstechnische Realisierung, die dem application engineering (vgl. Abschnitt 4.1.2) zuzuordnen ist, in seiner praktischen Umsetzbarkeit zu validieren.

5.1 Basistechnologien

Grundlage der Umsetzung ist die Erweiterung des im nächsten Abschnitt überblicksartig vorgestellten Workplace Portals G8. Neben dieser Basistechnologie für die zu koppelnden Portale an sich wird eine Infrastruktur zur Kommunikation zwischen diesen benötigt. Eine hierfür besonders geeignete Technologie stellen Web Services dar. Auf diesen zweiten technologischen Pfeiler der Umsetzung wird nach der Vorstellung des Workplace Portals G8 näher eingegangen.

5.1.1 Workplace Portal G8

Das Workplace Portal G8, das im weiteren Verlauf der Ausführungen verkürzend als G8-Portal bezeichnet wird, entstand am Groupware Competence Center (GCC) der Universität Paderborn als Forschungsprojekt eines modularen Workplace Portals. Dessen Kernaufgabe ist die effiziente Versorgung der „Knowledge Worker“ mit für ihre Arbeit notwendigen Informationen, Anwendungen und Werkzeugen. Durch integrierte Funktionalitäten zur Unterstützung von Teamstrukturen zeichnet sich das G8-Portal als Lösung im Groupware-Bereich aus. Sein modularer Aufbau ermöglicht die personalisierte Aggregation von Informationen auf der Basis der unterschiedlichsten Datenquellen und Anwendungen. Die Ausgabe des Portals erfolgt über einen Internet-Browser. Der Zugriff über WAP-Endgeräte oder Personal Digital Assistants (PDA) ermöglicht zusätzlich eine mobile Nutzung und hohe Flexibilität in den Arbeitsabläufen.

5.1.1.1 Architektur

Zur Gewährleistung hoher Flexibilität, ist das Java-basierte Portal-Framework des G8-Portals in drei Schichten (vgl. Abbildung 5-1) umgesetzt (vgl. [Hahl 2000], S. 23 f.). Den Kern der Architektur bildet die *Portal Core Engine*. Sie stellt verschiedene Portaldienste wie die Benutzerverwaltung, die Zugriffssteuerung und die benutzerabhängige Aggregation zur Ver-

fügung. Darüber hinaus übernimmt sie die Steuerung der Benutzerinteraktion und koordiniert das Zusammenwirken der weiteren Komponenten. Die im Portal zu integrierenden Systemtypen werden als *Content-Pools* bezeichnet. Die Anbindung an die Portal Core Engine erfolgt durch die Zwischenschicht der *Content-Adaptoren*. Diese sind für die Erzeugung der Darstellung der auf ihnen basierenden Portlets verantwortlich. Hierzu interagieren die informations- und anwendungssystemspezifischen Content-Adaptoren mit den Content-Pools über deren native Protokolle. Gegenüber der Portal Core Engine weisen sie jedoch eine einheitliche Schnittstelle auf, so dass eine Abstraktion und eine Entkopplung der eigentlichen Back-End-Systeme erfolgt. Die Administration der im Portal verfügbaren Portlets erfolgt über eine Lotus-Notes-Datenbank, die als *Portlet Definition Repository* bezeichnet wird. Für jedes Portlet definiert ein Konfigurationsdokument die Eigenschaften und Zugriffsrechte und legt gleichzeitig die Parameter zur Anbindung an einen spezifischen Content-Pool fest. Neben den Meta-Daten über den Aufbau und die Struktur des Portals werden zusätzlich Personalisierungsinformationen der Portalbenutzer in einer relationalen Datenbank (*Portal-DB*) abgelegt.¹³

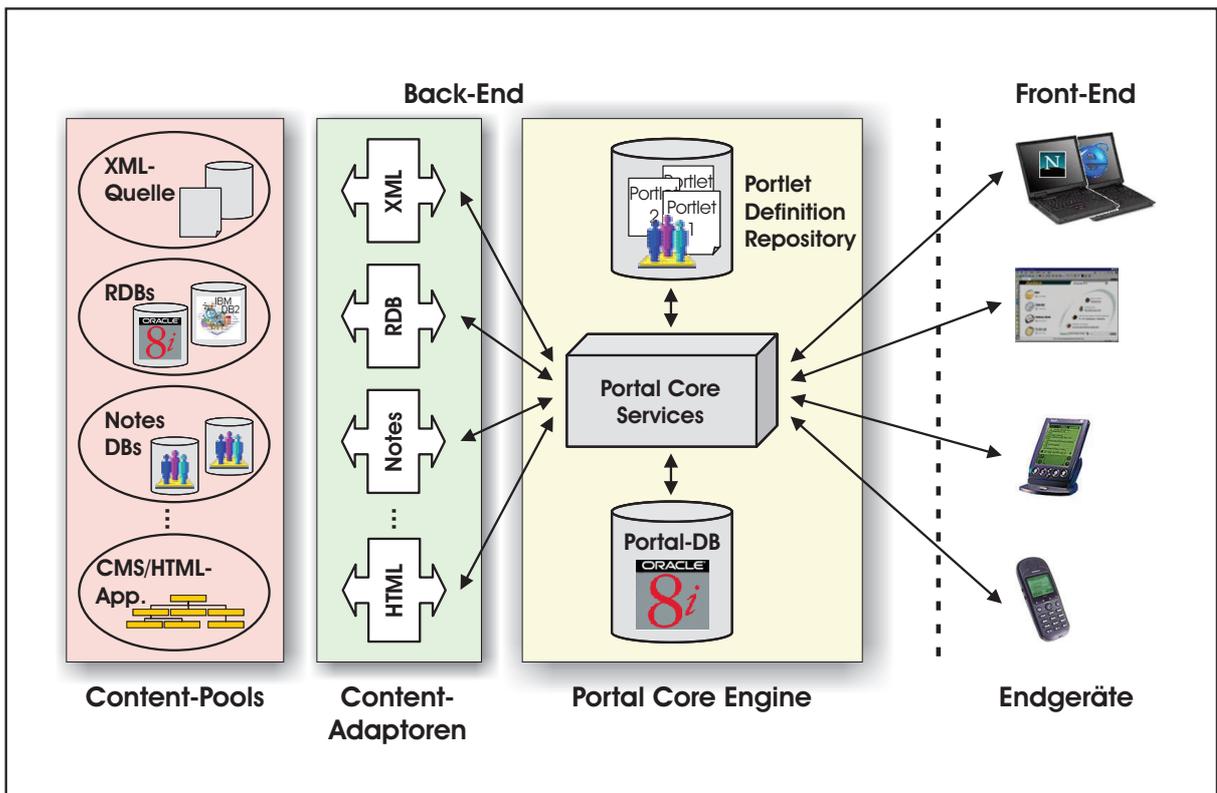


Abbildung 5-1: Konzeptionelle Portalarchitektur des G8-Portals

¹³ Für weitergehende Betrachtungen der Architektur des G8-Portals wird auf Hahl [Hahl 2000] verwiesen.

5.1.1.2 Basiselemente

Das G8-Portal unterstützt die wesentlichen, im Abschnitt 4.2 vorgestellten Basiskomponenten und -dienste eines Portals.

Den Benutzern wird ein so genannter Home-Place zur Verfügung gestellt, der dem persönlichen Platz aus Abschnitt 4.2.4 entspricht. Darüber hinaus haben berechtigte Benutzer die Möglichkeit, weitere gemeinsam nutzbare Plätze anzulegen. Diese weisen eine Zugriffskontrollliste (ZKL) auf, über die gesteuert werden kann, welche Benutzer oder Gruppen welche Rechte auf dem Platz besitzen. Ein Platz kann für alle Benutzer zugänglich oder auf einen geschlossenen Benutzerkreis beschränkt sein. Unabhängig davon, um welche Art des Platzes es sich handelt, kann dieser beliebig viele Seiten in linearer Ordnung enthalten. Eine spezifische Rechtevergabe auf Ebene der Seiten ist nicht vorgesehen. Sie sind für alle Benutzer mit Zugriff auf den Platz sichtbar. Änderungen können indes ausschließlich von Benutzern vorgenommen werden, denen in der ZKL das entsprechende platzweite Recht eingeräumt wurde. Die Seiten selbst basieren auf dem flexiblen Ansatz des Aufbaus aus Zeilen und Spalten, der weit reichende Gestaltungsmöglichkeiten für das Layout der Seiten bietet. Jede Seite kann in Anzahl und Typen beliebige Portlets aufnehmen und anzeigen. Deren Sichtbarkeit ist ausschließlich abhängig von den in der Portlet-Definition festgelegten Rechten der Benutzer und steht in keinem Zusammenhang mit der ZKL des Platzes. Die Konfigurierbarkeit der Portlets ist hingegen – wie auf der Ebene der Seiten – an das platzweite Recht gebunden.

Eine spezielle Form der Seite wird im G8-Portal als Vorlage bezeichnet. Vorlagen werden von den Administratoren des Portals definiert und weisen grundsätzlich die gleichen Eigenschaften wie normale Seiten auf. Sie haben ein flexibles Layout und können beliebige Portlets enthalten. Abhängig von der Art der Vorlage können Benutzer, für welche die Vorlage freigegeben wurde, diese in jeden Typ von Platz einbringen oder sie wird ihnen für ihren Home-Place automatisch hinzugefügt. Das Konzept der Vorlagen verfolgt das Ziel, thematisch zusammengehörige Portlets zentral zu Seiten zusammenzustellen und den Benutzern entweder als Vorschlag anzubieten oder zentral vorzugeben. Dies verringert den Aufwand, den jeder einzelne Benutzer bei der Zusammenstellung seines Portals hat, und erlaubt die gezielte zentrale Steuerung der Informationsverteilung.

Eine portalweite Suche ist über die Portlet-basierte Integration externer Suchdienste realisiert. Es stehen sowohl Portlets für index- als auch brokerbasierte Suchdienste zur Verfügung. Echtzeitkollaborative Eigenschaften werden gleichermaßen durch die Integration eines externen Produkts abgedeckt. Aufgrund des proprietären APIs des externen Produkts ist ein Austausch

nicht ohne weiteres möglich. Als wesentliches Personalisierungswerkzeug kommen die im vorherigen Absatz beschriebenen Vorlagen zum Einsatz. Die Personalisierung findet durch die deklarative Zuordnung von Vorlagen zu Benutzern oder Benutzergruppen statt. Weitergehende Möglichkeiten, die auf Basis der automatischen Analyse des Nutzungsverhaltens und weiterer Benutzerinformationen eine fortgeschrittenere Personalisierung erlauben, sind nicht verfügbar. Auf Ebene der in den Portlets angezeigten Informationen ist es Aufgabe der Portlets selbst, Profilinformationen der Benutzer abzufragen, um für diese eine personalisierte Anzeige zu generieren. Portalweite Dienste stehen aktuell nicht zur Verfügung.

5.1.2 Web Services

Web Services unterstützen ganz allgemein den Nachrichtenaustausch zwischen Software-Applikationen. Eine durchgängig akzeptierte Definition des Begriffs Web Service hat sich bisher nicht durchgesetzt. Stellvertretend soll die Definition des World Wide Web Consortiums (W3C) herangezogen werden:

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.” ([W3C 2004a])

Während das W3C allgemein einen Web Service als Software-System umschreibt, definiert Kreger ([Kreger 2001], S. 6) einen Web Service konkreter als aus einer Menge von Methoden bestehend, die mithilfe von XML-Nachrichten über ein Netzwerk aufgerufen werden können. Dieses Konzept ist in der Literatur seit längerem als „Distributed Computing“ bekannt (vgl. [Graham et al. 2002], S. 11). Gegenüber bisherigen Ansätzen basiert die Kommunikation bei Web Services jedoch nicht auf proprietären Protokollen, sondern auf der standardisierten Extensible Markup Language (XML). Deren Verwendung ermöglicht die als zentrale Eigenschaften von Web Services zu identifizierende Unabhängigkeit von Plattformen, Programmiersprachen und Implementierungen (vgl. [W3C 2004b]).

Die Web-Service-Technologie ist im Hinblick auf ein serviceorientiertes Architekturkonzept (SOA) (vgl. [W3C 2004b] und [Graham et al. 2002], S. 23 f.) konzipiert worden. Dieses definiert drei grundlegende Rollen und zugehörige Aufgaben (vgl. Abbildung 5-2):

- Die *Service-Anbieter* stellen ihre Software-Module als Services anderen Systemen (Service-Nachfragern) über ein Netzwerk zur Verfügung. Hierzu veröffentlichen sie die Adresse, unter der die Service-Beschreibung abrufbar ist, in einer Service-Registrierungsstelle.

- Die *Service-Registrierungsstellen* fungieren als Vermittler zwischen dem Service-Nachfrager und dem Service-Anbieter, indem sie eine Suchfunktion über die registrierten Web Services anbieten und dem Service-Nachfrager die Adresse, unter der die Service-Beschreibung abrufbar ist, zur Verfügung stellen.
- Die *Service-Nachfrager* können mithilfe der von der Service-Registrierungsstelle erhaltenen Adresse der Service-Beschreibung diese beim Service-Anbieter abrufen. Die Etablierung der Kommunikation mit dem Service-Anbieter und die Nutzung von dessen Services erfordert keine weitere Einbeziehung der Service-Registrierungsstelle.

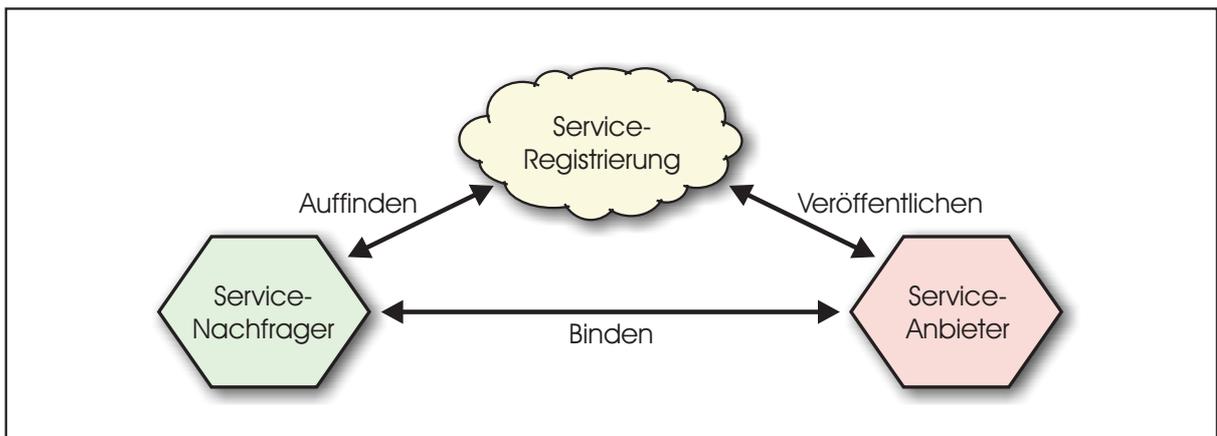


Abbildung 5-2: Rollen und Aufgaben in einer serviceorientierten Architektur
(in Anlehnung an [Graham et al. 2002], S. 23)

Verschiedene Autoren konzipieren einen so genannten Web Services Stack (vgl. Abbildung 5-3), der den verschiedenen Aufgaben einer serviceorientierten Architektur jeweils diejenigen Web-Service-Technologien gegenüberstellt, die sich als Standards oder Quasistandards etabliert haben. Abhängig vom Fokus des Autors wird dieser Stack um verschiedene Aufgaben erweitert bzw. anders strukturiert, so dass keine einheitliche Darstellung möglich ist.

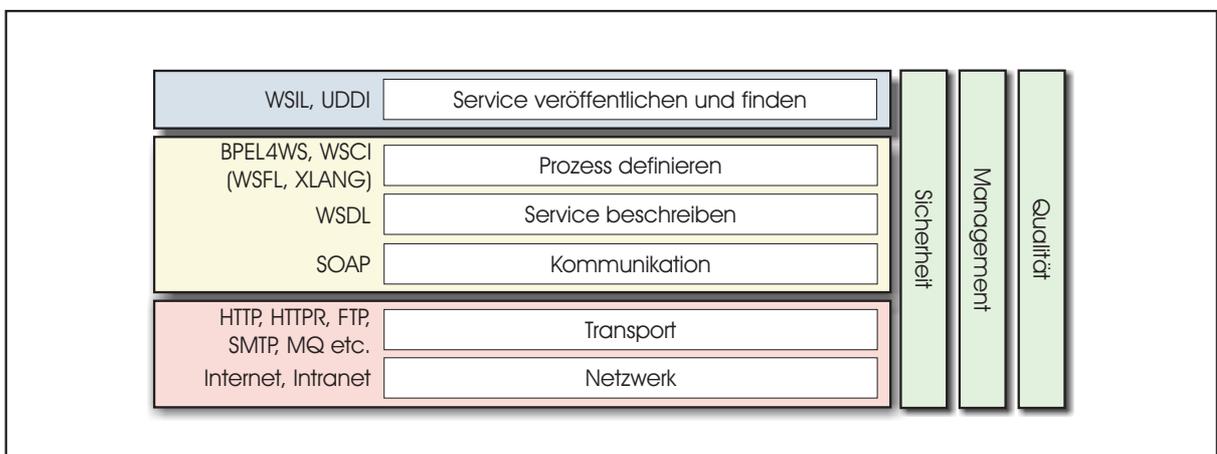


Abbildung 5-3: Web Service Stack
(in Anlehnung an [Kreger 2001], S. 10)

Im Folgenden wird nur auf die drei Kerntechnologien Simple Object Access Protocol (SOAP), Web Services Definition Language (WSDL) und Universal Description, Discovery and Integration (UDDI) eingegangen, die zusammen zum Synonym für Web Services geworden sind (vgl. [Graham et al. 2002], S. 114).¹⁴

5.1.2.1 Simple Object Access Protocol

Beim Simple Object Access Protocol handelt es sich um ein XML-basiertes, plattform- und programmiersprachenunabhängiges Kommunikationsprotokoll (vgl. [Cauldwell et al. 2001], S. 20), das auf einem vorhandenen Transportprotokoll aufsetzt und zum Austausch von durch XML strukturierten Nachrichten in verteilten Umgebungen zum Einsatz kommt. Die SOAP-Spezifikation ([W3C 2003a]) definiert eine verbindliche Struktur für SOAP-Nachrichten (vgl. Abbildung 5-4).

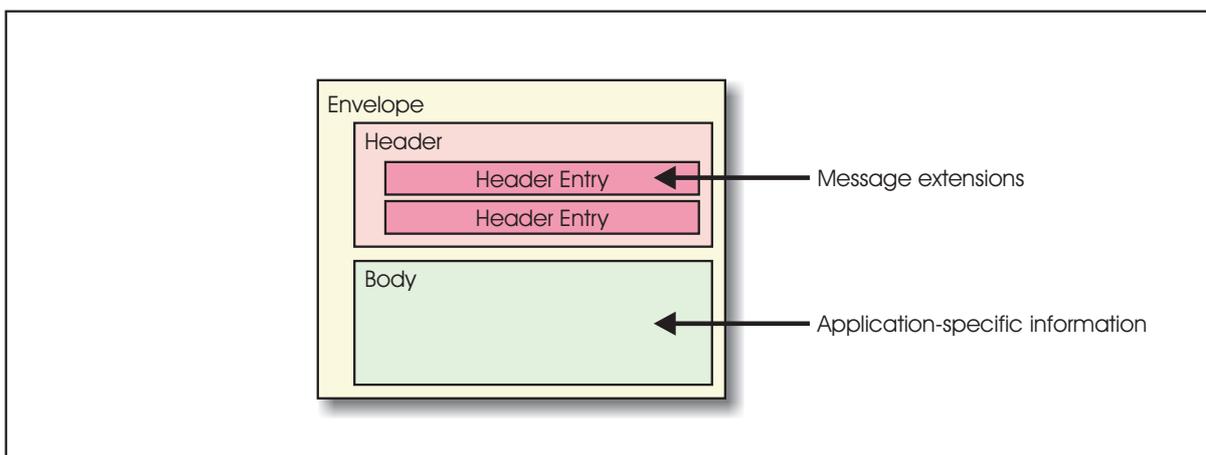


Abbildung 5-4: Struktur einer SOAP-Nachricht
(in Anlehnung an [Cauldwell et al. 2001], S. 82)

Eine SOAP-Nachricht besteht aus genau einem *Envelope*, der als Container für die zwei Komponenten *Header* und *Body* dient. Der optionale SOAP-Header kann aus mehreren Header-Einträgen bestehen und nimmt Meta-Informationen über die Nachricht auf. SOAP-Header stellen ein einfaches Konstrukt dar, um in verteilten Systemen beispielsweise Transaktionsmanagement- oder Authentifizierungsmechanismen unabhängig von der zu übertragenden Information zu realisieren (vgl. [Graham et al. 2002], S. 135). Der SOAP-Body muss zwingend vorhanden sein und enthält die eigentliche auszutauschende Nachricht. Diese kann

¹⁴ Weitergehende, auch die anderen Bereiche einbeziehende Ausführungen finden sich z. B. in Cauldwell et al. [Cauldwell et al. 2001] und Graham et al. [Graham et al. 2002].

beliebige XML-Konstrukte enthalten, so dass sowohl dokumentenorientierte als auch auf Remote Procedure Calls (RPC) basierende Interaktionsmodelle abgebildet werden können.¹⁵

5.1.2.2 Web Service Description Language

Web Services sind als sich selbst beschreibende Software-Komponenten konzipiert. Die Beschreibung setzt sich zusammen aus einer abstrakten Definition von Operationen und Nachrichten sowie den konkreten Angaben, wie und wo der Service zur Verfügung steht (vgl. Abbildung 5-5). Zur Erfüllung dieser Aufgabe wurde die ebenfalls XML-basierte Web Service Description Language (WSDL) geschaffen (vgl. [W3C 2001]). Ein WSDL-Dokument stellt dem Service-Nachfrager in maschinenlesbarer und -interpretierbarer Form im Bereich der *Definition der Service-Schnittstelle* alle notwendigen Informationen über die für die Kommunikation notwendigen Strukturen und angebotenen Funktionen zur Verfügung. Die *Definition der Service-Implementierung* ordnet diesen eine Zieladresse zu, unter der ein konkreter Service erreichbar ist. Steht ein Service über unterschiedliche Transportprotokolle zur Verfügung, so ist es möglich, für jedes Transportprotokoll eine eigene Zieladresse festzulegen.

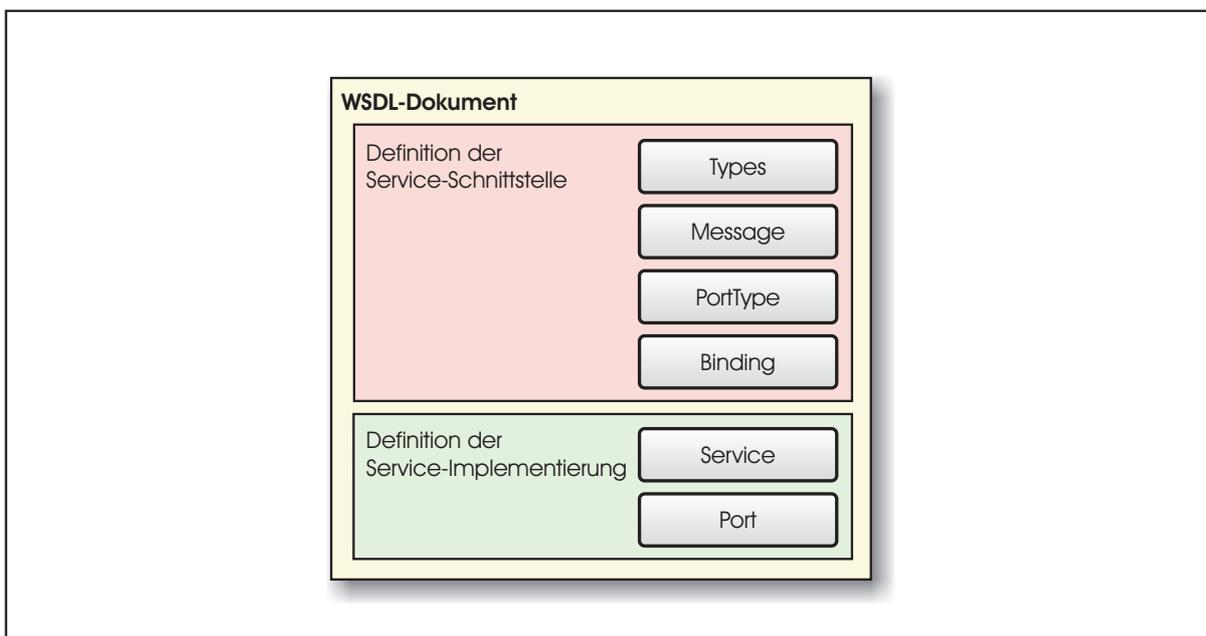


Abbildung 5-5: Struktur eines WSDL-Dokuments
(in Anlehnung an [Kreger 2001], S. 16)

Durch die verbindlich definierte Struktur und automatische Interpretierbarkeit ist es möglich, aus einer als WSDL-Dokument vorliegenden Service-Beschreibung automatisch so genannten

¹⁵ Eine ausführliche Einführung in SOAP findet sich z. B. in W3C [W3C 2003b] und Snell, Tidwell und Kulchenko [Snell/Tidwell/Kulchenko 2002].

Stub-Code zu erzeugen. Dieser stellt ein Programmgerüst zur Interaktion mit dem Web Service dar und kann als Ausgangspunkt für eine Integration in ein Anwendungssystem dienen. Umgekehrt ist es ebenfalls möglich, ein WSDL-Dokument automatisch aus einem existierenden Web Service zu generieren.

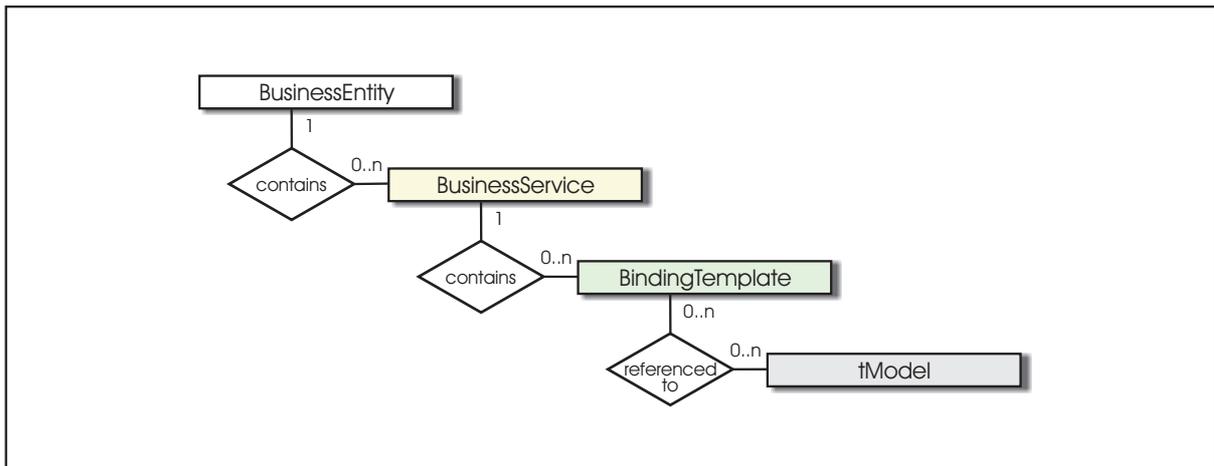
5.1.2.3 Universal Description, Discovery and Integration

Die serviceorientierte Architektur definiert eine Service-Registrierungsstelle, in der sich Service-Anbieter registrieren und ihre Services veröffentlichen können. Für die Web-Services-Architektur hat sich zu diesem Zweck die Spezifikation der Universal Description, Discovery and Integration (UDDI) als Quasistandard etabliert (vgl. z. B. [Webber/Dutton 2000]).

Die UDDI-Spezifikation wurde mit dem Ziel konzipiert, ein universelles, allgemein als Universal Business Registry (UBR) bzw. speziell im Zusammenhang mit UDDI auch als UDDI-Registry bezeichnetes Verzeichnis zu schaffen, in dem Unternehmen ihre angebotenen, internetbasierten Dienstleistungen veröffentlichen können. Zu diesem Zweck sieht eine UBR Informationen in drei Kategorien vor (vgl. [Graham et al. 2002], S. 395 f.):

- *Weißer Seiten*: Informationen über den Anbieter auf informeller Ebene; z. B. über seinen Firmennamen, Kontaktdaten und Ansprechpartner sowie textuelle Beschreibungen.
- *Gelbe Seiten*: Klassifikation der Anbieter gemäß Geschäftskategorien, unterteilt in Produkte, angebotene Dienste und geografische Zuordnung.
- *Grüne Seiten*: technische Beschreibungen und Klassifizierung der angebotenen Dienste.

Die grundlegende Datenstruktur der UDDI-Spezifikation lässt sich in Anlehnung an OASIS ([OASIS 2002]) für die registrierten Dienste eines Anbieters als Entity Relationship (ER) Diagramm darstellen (vgl. Abbildung 5-6).



**Abbildung 5-6: ER-Diagramm des UDDI-Datenmodells
(in Anlehnung an [OASIS 2002])**

Der Datentyp *BusinessEntity*, der den weißen Seiten zuzuordnen ist, bildet den Ausgangspunkt der Hierarchie und enthält die allgemeinen Informationen über den Anbieter. Elemente vom Typ *BusinessService* erfüllen durch die Bereitstellung beschreibender Informationen über eine bestimmte Kategorie von Diensten eines Anbieters die Funktion der gelben Seiten. Die *BindingTemplates* nehmen Informationen über die technischen Eigenschaften der Schnittstellen der angebotenen Dienste auf und lassen sich daher den grünen Seiten zuordnen.

Dem Datentyp *tModel* (Technology Model) kommt eine besondere Funktion zu. Das tModel selbst erlaubt ausschließlich die abstrakte Definition und Identifikation einzelner Spezifikationen durch die Verwendung eindeutiger Bezeichnungen (vgl. [Graham et al. 2002], S. 397). Durch die Zuordnung eines tModels zu einem BindingTemplate ist es möglich, eben dieses BindingTemplate als die vom tModel repräsentierte Spezifikation implementierend zu kennzeichnen. Damit können tModels ebenfalls zur Klassifikation und Identifikation von Diensten herangezogen werden (vgl. [Cauldwell et al. 2001], S. 193). Die beispielsweise durch Standardisierungsgremien vorgenommene Definition von tModels ermöglicht es, in einem UDDI-Verzeichnis eine gezielte Suche nach allen Services durchzuführen, die den entsprechenden Standard unterstützen.¹⁶

5.1.2.4 Web Services als Middleware für die Kopplung von Portalen

Die Verwendung etablierter Standards, die hohe Akzeptanz und damit weite Verbreitung bei Herstellern und Anwendern, die freie Verfügbarkeit und die Plattform-, Programmierspra-

¹⁶ Eine vollständige Darstellung der UDDI-Spezifikation und weitere Informationen finden sich auf den Internetseiten der sie entwickelnden Organisation UDDI.ORG (<http://www.uddi.org>).

chen- sowie Implementierungsunabhängigkeit prädestinieren Web Services für den Einsatz bei der Konzeption einer hersteller- und produktübergreifenden Kopplung von Portalen. Vor diesem Hintergrund erfolgt die Umsetzung des Daten- und Funktionsmodells auf Basis von Web-Service-Technologien als Infrastruktur- und Middleware-Komponente im Bereich der standardisierten Kommunikation und dem Auffinden von Partnern.

Aufgrund der verhältnismäßig jungen Technologie fehlen in Bereichen, die über die Basisfunktionen hinausgehen, teilweise noch wichtige Bausteine, die für einen robusten Einsatz in verschiedenen Anwendungsbereichen notwendig sind. Als Beispiele für fehlende Funktionen lassen sich Aspekte der Sicherheit, der garantierten Kommunikation und verteilte Transaktionen anführen. Obwohl diese Faktoren bei der Entscheidung für die Verwendung von Web Services als Basistechnologie zur Kopplung von Portalen zu berücksichtigen sind, ist davon auszugehen, dass diese Fragestellungen kurz- bis mittelfristig gelöst werden. Es ist zu beobachten, dass die extrem hohe Dynamik der vergangenen Jahre weiterhin ungebrochen anhält und zu einer massiven Verbesserung und Erweiterung der schon jetzt gegebenen Möglichkeiten beiträgt. Durch die Eingliederung in die bestehende grundlegende Struktur des Web Service Stack (vgl. Abschnitt 5.1.2) sind diese bei deren Verfügbarkeit in das entwickelte Konzept zur Kopplung von Portalen als Erweiterung und nicht als vollständige Neudefinition integrierbar.

5.1.2.5 Web-Services-Frameworks und UDDI-Implementierungen

Bei der Implementierung des in Kapitel 4 vorgestellten Konzepts zur Kopplung von Portalen kann auf am Markt verfügbare Web-Service-Frameworks und UDDI-Implementierungen zurückgegriffen werden. Diese brauchen nicht selbst erstellt zu werden, so dass die benötigte Web-Service-Infrastruktur grundsätzlich als gegeben angesehen werden kann.

Die Wahl eines konkret zu verwendenden Web-Service-Frameworks wird im Folgenden auf Java-basierte Systeme eingeschränkt, da das für die Implementierung herangezogene G8-Portal-Framework auf dieser Programmiersprache basiert. Darüber hinaus stehen aber auch für praktisch alle in aktivem Gebrauch befindlichen Programmiersprachen entsprechende Web-Service-Frameworks zur Verfügung.

Das ursprünglich im Rahmen der IBM alphaWorks-Initiative entwickelte und anschließend als Open-Source-Projekt unter dem Dach der Apache Software Foundation (ASF) freigegebene Apache Soap Toolkit stellt eine der ersten Java-basierten Web-Service-Implementierungen dar (vgl. [Cauldwell et al. 2001], S. 34). Da es sich um die erste Generation von Web-Service-

Frameworks handelt, weist das Apache Soap Toolkit wesentliche Limitationen auf (vgl. [Basha/Irani 2002], S. 19 f.):

- Fehlende WSDL-Unterstützung.
- Unzureichende Unterstützung von SOAP-Intermediaries.
- Keine Zugriffsmöglichkeiten auf SOAP-Header bei RPC-Interaktionen.
- Eingeschränkte Unterstützung von alternativen Transportprotokollen.
- Langsame XML-Verarbeitung durch die Verwendung des Document Object Model (DOM) anstelle der ereignisgesteuerten Auswertung über das Simple API for XML (SAX).

Da die Limitationen im Wesentlichen architekturimmanent sind, wurde die Weiterentwicklung eingestellt. Stattdessen entstand, basierend auf den gesammelten Erfahrungen, ein vollständig neues, auf Performance, Flexibilität, Offenheit und Erweiterbarkeit angelegtes Framework, das unter dem Namen Apache Extensible Interaction System (AXIS) firmiert (vgl. [Graham et al. 2002], S. 125 f.). Die wesentliche Architekturentscheidung beim Design des neuen Frameworks bestand darin, die gesamte Verarbeitung modular auszulegen. Jeder Verarbeitungsschritt – beim Client zwischen dem Versand der Nachricht bis zum Empfang der Antwort und beim Server vom Empfang einer Nachricht bis zum Versand der Antwort – durchläuft mehrere so genannte Message-Paths. Diese setzen sich aus speziellen, als Handler bezeichneten Elementen zusammen. Bei der Verarbeitung der SOAP-Nachricht passiert diese die für die Message-Paths konfigurierten Handler. Die Handler haben weit reichende Möglichkeiten zur Verarbeitung einzelner Teile oder der ganzen SOAP-Nachricht und leiten diese nach Beendigung ihrer Aufgabe an den nächsten Handler im Message-Path weiter. Neben bereits im AXIS-Framework vorhandenen Handlern, welche die Standardverarbeitung von Web Services sicherstellen, bietet dieser Mechanismus eine flexible Schnittstelle zur Erweiterung und Anpassung an zusätzliche Standards und anwendungsspezifische Anforderungen.¹⁷

Beiden Frameworks fehlt die Möglichkeit zum Zugriff auf eine UDDI-basierte Service-Registrierungsstelle. Geschlossen wird diese Lücke durch das UDDI4J-Projekt, das ebenfalls von IBM entwickelt und später als Open-Source unter dem Dach der ASF freigegeben wurde. Es stellt ein Java-basiertes Application Programming Interface (API) zur Publikation und Suche in UDDI-Verzeichnissen zur Verfügung. UDDI4J kann diese ergänzend sowohl mit dem Apache Soap Toolkit als auch mit Apache AXIS eingesetzt werden.

¹⁷ Eine ausführlichere Beschreibung der AXIS-Architektur findet sich z. B. in Basha und Irani [Basha/Irani 2002] und Graham et al. [Graham et al. 2002].

Im Bereich der UDDI-Verzeichnisse bieten mehrere große Hersteller wie IBM, Microsoft und HP öffentliche, miteinander vernetzte und synchronisierte Service-Registrierungsstellen an. Zu Entwicklungszwecken und zur Realisierung geschlossener, nicht öffentlich zugänglicher UDDI-Verzeichnisse kann z. B. auf jUDDI zurückgegriffen werden, eine ebenfalls unter dem Dach der ASF angesiedelte, in Java realisierte Open-Source-Implementierung der UDDI-Spezifikation in der Version 2.0. Die Daten der Registrierungen können in einer Vielzahl von relationalen Datenbanken vorgehalten werden. In der verwendeten Version wird jedoch keine Administrationsoberfläche zum Management bestehender, zum Anlegen neuer und zur Suche in vorhandenen Registrierungen angeboten. Diese Aufgaben sind ausschließlich programmatisch über das standardisierte UDDI-API möglich.

Für die Server-Komponente und den Teil der Client-Komponente der Portalkopplung (vgl. Abschnitt 5.2), der direkt von den Portal-Diensten verwendet wird, ist das Apache AXIS-Framework dem Apache Soap Toolkit vorzuziehen. Dieses bietet mit seiner guten Performance, Offenheit und Erweiterbarkeit die notwendigen Voraussetzungen für die erfolgreiche Umsetzung der serverbasierten Portalkopplung. Diese Eigenschaften gehen gleichzeitig mit erhöhten Anforderungen an die Systemumgebung und einer komplexeren Konfiguration einher. Für den Teil der Client-Komponente, der direkt von den Administratoren zur Konfiguration der Umgebung verwendet wird und der sich, wie im Verlauf der weiteren Ausführungen dargestellt wird, innerhalb der vom G8-Portal verwendeten Lotus-Notes-Datenbank des Portlet Definition Repositories befindet, wird aus diesen Gründen weiterhin auf das Apache Soap Toolkit zurückgegriffen. Dieses wird mit dem UDDI4J-API kombiniert zum Einsatz gebracht, um sowohl auf das durch jUDDI zur Verfügung gestellte, als auch auf andere standardkonforme UDDI-Verzeichnisse zugreifen zu können.

5.2 Kopplung von G8-Portalen

Die Anwendung des allgemeinen Konzepts zur Kopplung von Portalen auf die Kopplung von G8-Portalen hat das in Abschnitt 4.6 dargestellte Daten- und Funktionsmodell als Schnittstelle zu den externen Portalen zu implementieren und die im Abschnitt 4.5 festgeschriebene Kommunikations- und Synchronisationsarchitektur intern umzusetzen. Neben der Darstellung der Basisdienste zur für die vorhandene Implementierung transparenten Verbindung mehrerer G8-Portale werden ausgewählte Aspekte der Realisierung vorgestellt. Sie berücksichtigen alle Stufen des Kontinuums der Portalkopplung (vgl. Abschnitt 4.3.2). Die Ausführungen werden mit konkreten Beispielen einer Testumgebung des Systems unterlegt.

5.2.1 Basisdienste für die Portalkopplung

Um das bestehende G8-Portal-Framework für den entfernten Zugriff zu öffnen, wird dieses um drei Web-Service-Schnittstellen erweitert:

- *Portal Soap Gateway*
- *Portal Soap Gateway Proxy*
- *Portal Router Service*

Das *Portal Soap Gateway* ist die zentrale Komponente, welche die für die Kopplung notwendigen Operationen über die Realisierung des festgelegten Daten- und Funktionsmodells den kooperierenden Portalen zur Verfügung stellt. Aus Sicht eines Client-/Server-Ansatzes handelt es sich beim Portal Soap Gateway um den Server, der eingehende SOAP-Nachrichten nach einer Authentifizierung und Autorisierung der mithilfe eines SOAP-Headers übertragenen Benutzerdaten entsprechend verarbeitet und selbst durch eine SOAP-Nachricht beantwortet. Der Aufbau und die Struktur der eingehenden und ausgehenden Nachrichten entsprechen den festgelegten Standards und basieren auf dem Interaktionsmodell eines Remote Procedure Calls (RPC). Die Verarbeitung hat die entsprechend definierten Abläufe der Kommunikations- und Synchronisationsarchitektur umzusetzen. Die Implementierung ist jedoch spezifisch für das Framework des G8-Portals. Der Austausch der Nachrichten erfolgt nach dem Anfrage/Antwort-Schema über das Hypertext Transport Protocol (HTTP) als Transportsystem. Dies erlaubt die notwendige synchrone Echtzeitkommunikation.

Der *Portal Soap Gateway Proxy* stellt den zum Portal Soap Gateway passenden Client zur Verfügung. Er fungiert als lokales Abbild des Portal Soap Gateways und kapselt die Web-Service-basierte Kommunikation zwischen den gekoppelten Portalen. Die Umsetzung des in der Software-Entwicklung weit verbreiteten Proxy-Patterns ermöglicht es, so weit wie möglich zu verbergen, dass mit einem entfernten Portal kommuniziert wird. Dadurch können die diesbezüglichen Veränderungen an der bestehenden Architektur begrenzt werden.

Indem jedem G8-Portal sowohl ein Portal Soap Gateway als auch ein Portal Soap Gateway Proxy zur Verfügung steht, kann es abhängig von der Situation sowohl als Server als auch als Client der Kommunikation agieren. Ähnlich wie in klassischen Client/Server-Systemen erfolgt die Initiierung der Kommunikation ausschließlich durch den Client über den Portal Soap Gateway Proxy. Dieser baut die Kommunikationsverbindung mit dem Portal Soap Gateway des Kommunikationspartners auf. Abbildung 5-7 zeigt schematisch die Interaktion der beiden Komponenten für direkt miteinander gekoppelte G8-Portale.

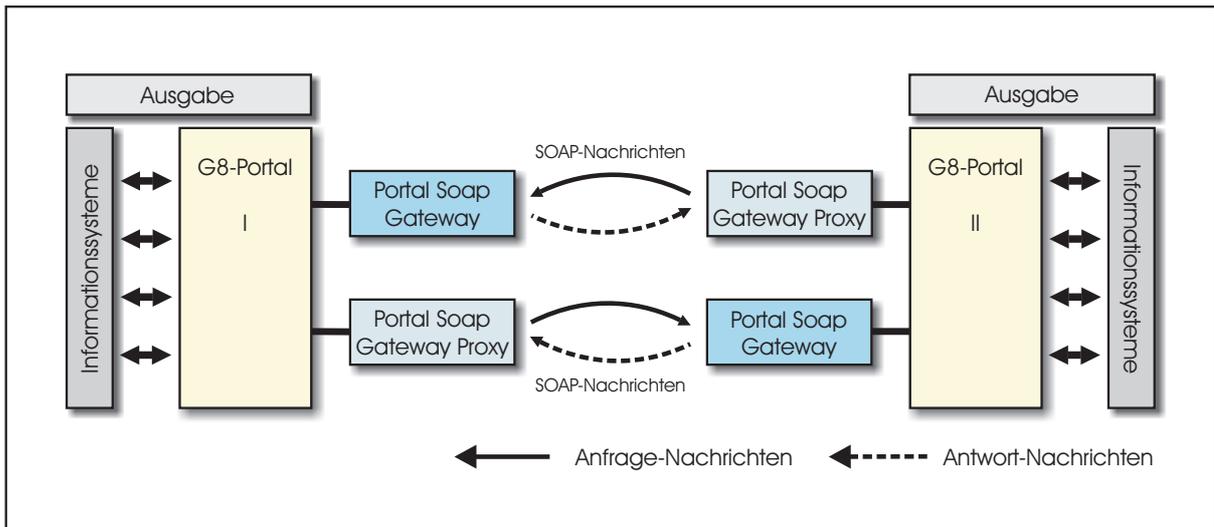


Abbildung 5-7: Direkte Kommunikationsbeziehung zwischen dem Portal Soap Gateway und dem Portal Soap Gateway Proxy
(in Anlehnung an [Ploch 2003])

Der *Portal Router Service* wird für die Unterstützung des Portlet-Routings benötigt (vgl. Abschnitt 4.5.3). Somit wird die Notwendigkeit zur Bildung von Maschen-Strukturen vermieden, da der Zugriff auf Portlets auch dann möglich ist, wenn keine direkte Kommunikationsbeziehung zwischen zwei Portalen existiert und diese nur über weitere Intermediär-Portlet-Provider zugreifbar sind (vgl. Abschnitt 4.5.1). Abbildung 5-8 visualisiert die Funktionsweise des Portal Router Service. Besteht keine direkte Kommunikationsbeziehung, so richtet der Portal Soap Gateway Proxy seine Anfrage nicht direkt an das Portal Soap Gateway des eigentlichen Portlet-Providers, sondern an den Portal Router Service des Portals, das ihm das Portlet als Intermediär-Portlet-Provider zur Verfügung stellt. Der Portal Router Service ermittelt – nach erfolgreicher Prüfung der Zugriffsrechte – mithilfe eines SOAP-Headers und der intern vorliegenden Konfiguration, ob er das Portlet direkt vom Portlet-Provider beziehen kann oder abermals auf einen Intermediär-Portlet-Provider zugreifen muss. Der SOAP-Header enthält Informationen über das Portlet, für das die Aktion durchgeführt wird. Kann das Portlet nicht direkt vom Portlet-Provider abgerufen werden, leitet der Portal Router Service die Anfrage an den nächsten Portal Router Service weiter, andernfalls findet die Weiterleitung an das Portal Soap Gateway des Portlet-Providers statt. In beiden Fällen ist der SOAP-Header, der für die Authentifizierung der Portale untereinander herangezogen wird, für die nächste Verbindung durch einen neuen zu ersetzen. Wird der Name des Benutzers, der die Aktion tatsächlich ausführt, übermittelt und ist Subject Switching erlaubt und konfiguriert, ist es die Aufgabe des Portal Router Service, dieses entsprechend durchzuführen (vgl. Abschnitt 4.4.2.4.1). Die Antwort des Portlet-Providers nimmt den umgekehrten Weg und erreicht den

ursprünglichen Absender der Anfrage ebenfalls unter Vermittlung aller zwischengeschalteten Portal Router Services.

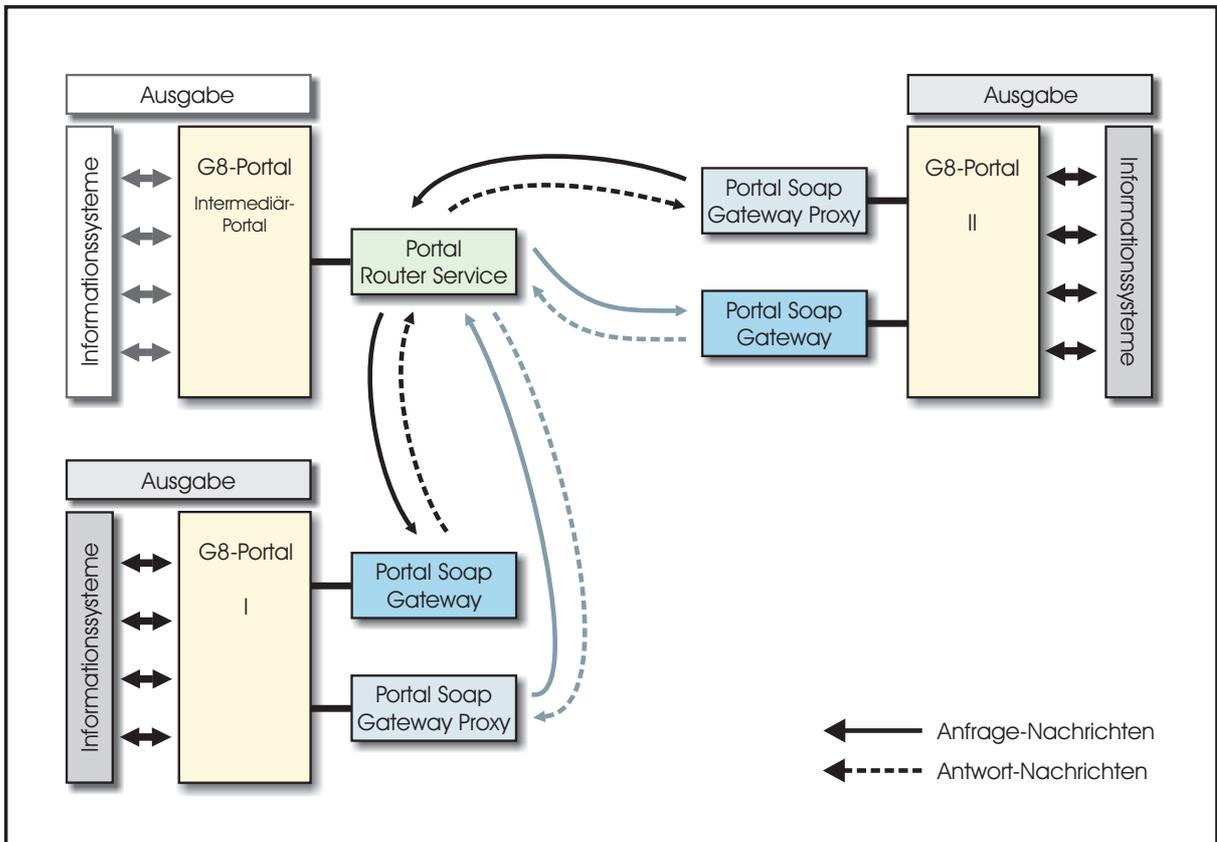


Abbildung 5-8: Indirekte Kommunikationsbeziehung über einen Intermediär-Portlet-Provider mittels des Portal Router Service
(in Anlehnung an [Ploch 2003])

Die Bidirektionalität des Portal Router Service und die transparente Vermittlung der Anfragen an den eigentlichen Portlet-Provider erlauben eine indirekte, aber gleichwohl transparente Nutzung der Portlets. Die Architektur lässt prinzipbedingt beliebig lange Ketten aus Portal Router Services zu. Mit wachsender Länge besteht durch die sich summierenden Latenzen die Gefahr, dass die Transparenz der Nutzung eines entfernten Portlets für den Endbenutzer zu sehr eingeschränkt wird oder völlig verloren geht. Abhängig vom Einzelfall ist dann zu prüfen, ob entweder eine direkte Verbindung zum Portlet-Provider etabliert werden kann oder aber zumindest eine Verkürzung der Länge des Kommunikationswegs durch die Etablierung einer Verbindung zu einem anderen Intermediär-Portlet-Provider möglich ist.

Technologisch sind zahlreiche Funktionen wie die Authentifizierung, das Logging und Auditing sowie das Portlet-Routing als Handler der AXIS-Architektur realisiert (vgl. Abschnitt 5.1.2.5). Dies ermöglicht die weitgehende Entkopplung dieser Service-Funktionen von der

Kernarchitektur des G8-Portals. Die Wartung und Weiterentwicklung der Gesamtarchitektur wird dadurch ebenfalls erleichtert.

5.2.2 Kommunikationsbeziehung zwischen Portalen

Die Grundlage für jede Form der Kopplung von Portalen ist die Etablierung einer direkten oder indirekten Kommunikationsbeziehung zwischen diesen. Dieser Aspekt ist Gegenstand der folgenden Betrachtungen und stellt die Umsetzung zur Konfiguration des in Abschnitt 4.6.1 vorgestellten Teils des entwickelten Daten- und Funktionsmodells dar. Die zu beschreibenden Erweiterungen wurden in die bereits vorhandene Lotus-Notes-basierte Administrationsumgebung des Portlet Definition Repositories integriert.

5.2.2.1 Portalregistrierung

Zur Vereinfachung der Administrierbarkeit und Umsetzung einer SOA ist vorgesehen, dass an einer Kopplung beteiligte bzw. für diese in Frage kommende Partner ihr Portal in einer Business Partner UDDI registrieren. Über ein entsprechendes Dokument konfiguriert der Administrator auf der einen Seite das zu verwendende UDDI-Verzeichnis mit allen zum Zugriff notwendigen Parametern, auf der anderen Seite steuert er die Veröffentlichung und Aktualisierung der Informationen über die eigene Organisation und das zu registrierende Portal. Der Name der Organisation und allgemeine Kontaktinformationen werden im Bereich der weißen Seiten des UDDI-Verzeichnisses abgelegt. Aktuell ist keine Klassifizierung im Sinne der gelben Seiten vorgesehen. Die Definition der Ziel-Adressen zum Zugriff auf das Portal Soap Gateway und den Portal Router Service decken zusammen mit der Registrierung einer weltweit eindeutigen Portal-ID (vgl. Abschnitt 4.5.2.2), die sich in diesem Fall aus der Replika-ID des als Lotus-Notes-Datenbank realisierten Portlet Definition Repositories ergibt, den Bereich der grünen Seiten ab.

Die Registrierung eines Portals und seiner Schnittstellen erfordert zunächst die UDDI-spezifische Definition einiger allgemeiner Bezeichnungen. Diese ergeben sich unmittelbar aus der Spezifikation zur Kopplung von Portalen und sind unabhängig von einer konkreten Implementierung. Die Bezeichnungen werden in Form der in Abschnitt 5.1.2.3 beschriebenen tModels für ein öffentliches UDDI-Verzeichnis durch das Standardisierungsgremium veröffentlicht. Im Fall der Verwendung eines privaten UDDI-Verzeichnisses ist es die Aufgabe eines Teilnehmers, die einmalige Veröffentlichung vorzunehmen. Eine in das Portlet

Definition Repository eingebettete Funktion kann diese Aufgabe übernehmen. Die folgenden Bezeichnungen sind für die verschiedenen Anwendungszwecke vorgesehen:

- *Portal*: klassifiziert einen UDDI-Eintrag als Portal, das für Kopplungen nach dem in dieser Arbeit entwickelten Standardisierungsvorschlag zur Verfügung steht.
- *Portal-ID*: definiert den Datentyp „Portal-ID“, der ein Portal weltweit eindeutig identifizierbar macht.
- *Portal Soap Gateway*: kennzeichnet einen UDDI-Eintrag als Zieladresse des Portalservice des Portal Soap Gateways.
- *Portal Router Service*: kennzeichnet einen UDDI-Eintrag als Zieladresse des Portalservice des Portal Router Service.
- *Portlet Type Identifier*: definiert den Datentyp „Portlet-Typ“ zur Klassifizierung von Portlets (vgl. Abschnitt 4.5.4.5.2).

Über die Definition und jeweils automatische Zuordnung der tModels zu den Daten der Registrierung eines Portals ist es möglich, im konfigurierten UDDI-Verzeichnis gezielt nach Portalen zu suchen, die für eine zur vorliegenden Spezifikation konforme Kopplung zur Verfügung stehen. Da die verwendete UDDI-Implementierung keine Administrationsoberfläche anbietet, aber auch zur Vereinfachung der Lotus-Notes-basierten Administration selbst, werden alle mit den entsprechenden tModels gekennzeichneten Portale zyklisch aus dem UDDI-Verzeichnis ausgelesen, im Portlet Definition Repositories gespiegelt und entsprechende Änderungen abgeglichen. Dadurch, dass jedes Portal auf eine lokale Kopie des Informationsstands aller registrierten Portale zugreifen kann, besteht keine direkte Abhängigkeit von der Service-Registrierung, was einen hohen Grad an lokaler Autonomie (vgl. Abschnitt 4.4.2.2) ermöglicht. Eventuell entstehende Inkonsistenzen zwischen den Werten des lokalen Spiegelbilds der Portalregistrierung und der Business Partner UDDI werden in regelmäßigen Intervallen oder durch eine manuell angestoßene Synchronisation aufgelöst.

5.2.2.2 Portalverbindung

Die eigentliche Konfiguration einer Verbindung zwischen zwei Portalen erfolgt mithilfe von *Portal-Connection-Dokumenten*. Für eine Punkt-zu-Punkt-Verbindung ist bei jedem der Partner für den jeweils anderen im Portlet Definition Repository ein Portal-Connection-Dokument anzulegen.

Das Dokument setzt sich im Wesentlichen aus drei Bereichen zusammen (vgl. Abbildung 5-9). Der erste erfasst allgemeine Eigenschaften des Portals des Partners. Hierzu gehören die ausschließlich zur Anzeige im eigenen Portal verwendete Bezeichnung, die das Portal weltweit eindeutig identifizierende Portal-ID und der Benutzername, der dem entfernten Portal lokal zugeordnet wird (vgl. Abschnitt 4.4.2.4.1). Der zweite Bereich umfasst die Festlegung der Schnittstellenparameter zum Zugriff auf das entfernte Portal. Diese beinhalten die Service-Adressen der Portaldienste und die zur Authentifikation notwendigen Informationen. Zuletzt können für die Verbindung global geltende Einstellungen bzgl. der Synchronisation der Portlet-Definitionsdocumente (vgl. Abschnitt 5.2.3.2.1) und der Behandlung des Portlet-Routings (vgl. Abschnitt 5.2.3.2.2) vorgenommen werden.

Portal Connection	
Bitte beachten: Im Remote-Portal muss auch eine Portal Connection für dieses Portal angelegt werden!	
Allgemein Sprachen	
Bezeichnung des Portals	🔍
PortalID	🔍
Lokaler Portal-Benutzer	🔍
Sichere Verbindung verwenden	<input checked="" type="radio"/> Ja <input type="radio"/> Nein
Portal-Soap-Gateway-URL	🔍
Router-Service-URL	🔍
Service-URN	urn:g8-remote-portal-service
Service-Adressen mit UDDI synchronisieren	<input checked="" type="radio"/> Aktiv <input type="radio"/> Ausgesetzt
Login beim Remote-Portal	🔍
Passwort	🔍
Portlets automatisch synchronisieren	<input checked="" type="radio"/> Aktiv <input type="radio"/> Ausgesetzt
Portlet-Routing für Partner gestatten	<input checked="" type="radio"/> Ja <input type="radio"/> Nein
Geroutete Portlets automatisch transferieren	<input checked="" type="radio"/> Ja <input type="radio"/> Nein
Remote-Portlets Namenszusatz	🔍
User-Switching für Partner gestattet	<input checked="" type="radio"/> Ja <input type="radio"/> Nein

Abbildung 5-9: Portal-Connection-Konfigurationsdokument

Ein Portal-Connection-Dokument kann vollständig manuell angelegt werden. Durch die Auswahl eines im UDDI-Verzeichnis registrierten Portals ist es jedoch auch möglich, die allgemeinen Einstellungen und Schnittstellenparameter direkt aus der Portalregistrierung zu übernehmen. Zusätzlich besteht die Option, im UDDI-Verzeichnis veränderte Werte automatisch in das Konfigurationsdokument übernehmen zu lassen. Beides reduziert die Fehleranfälligkeit und erleichtert die Administrierung der Verbindungen. Die Reduktion des Administrationsaufwands ist umso bedeutsamer, als sie es wirtschaftlich sinnvoller macht, Strukturen zu

schaffen, die einer Masche (vgl. Abschnitt 4.5.1) nahe kommen und den höchsten Grad an Robustheit und Performance für die Portalkopplung bieten.

5.2.3 Anbieten von Portlets

Die erste Stufe des Kontinuums der Portalkopplung stellt das Anbieten von Portlets dar, das zugleich auch die Grundlage für die weiteren Stufen ist (vgl. Abschnitt 4.3.2). Die notwendigen Erweiterungen des Portal-Frameworks lassen sich im Wesentlichen in zwei Bereiche untergliedern. Beim Portlet-Provider ist die Unterstützung zum eigentlichen Anbieten von Portlets und beim Portlet-Consumer zum Einbinden der Remote-Portlets zu ergänzen.

5.2.3.1 Erweiterungen zum Anbieten von Portlets

5.2.3.1.1 Konfiguration

Lokal verfügbare Portlets werden im G8-Portal über jeweils ein Konfigurationsdokument im Portlet Definition Repository abgebildet. In diesem wird neben zahlreichen weiteren Parametern der Zugriff für die lokalen Benutzer definiert. Da es sich beim Anbieten von Portlets um eine Ausweitung der Zugriffsrechte auf entfernte Portale handelt, ist es folgerichtig, deren Definition ebenfalls direkt in die Konfigurationsdokumente aufzunehmen (vgl. Abbildung 5-10).

Remote Benutzer / -gruppen	<input type="text" value="Portal/Partner1/DE;Portal/Partner2/DE"/>	<input type="button" value="Portal auswählen"/>
Erfordert tatsächlichen Benutzer	<input checked="" type="radio"/> Ja <input type="radio"/> Nein	
Benutzerzugriffsbeschränkungen		
Zugriff	<input type="text" value="*/Sales/Partner1/DE
*/Partner2/DE"/>	
Kein Zugriff	<input type="text" value="*/Partner3/DE"/>	
User-Switching erlauben	<input type="radio"/> Ja <input checked="" type="radio"/> Nein	
Portlet-Routing erlauben	<input checked="" type="radio"/> Ja <input type="radio"/> Nein	

Abbildung 5-10: Bereich zur Zugriffssteuerung eines gekoppelten Portals auf ein Portlet in einem Portlet-Konfigurationsdokument

Grundsätzlich ist festzulegen, welche gekoppelten Portale Zugriff auf ein lokales Portlet erhalten sollen. Dies geschieht über die Auswahl des generischen Benutzernamens des nachfragenden Portals. Sind für dieses Portlet weitergehende Zugriffsbeschränkungen durchzusetzen, so kann erzwungen werden, dass der Benutzername des tatsächlich den Zugriff durchführenden Benutzers übertragen wird. Auf dieser Basis kann der Zugriff anschließend explizit

zugelassen oder ausgeschlossen werden. Wie im Abschnitt 4.4.2.4.1 empfohlen, ist durch die Definition von Platzhaltern keine genaue Angabe von einzelnen Benutzern notwendig. Bei erlaubtem Routing dieses Portlets (vgl. Abschnitt 5.2.3.2.2) kann dem Intermediär-Portlet-Provider das Subject Switching weiterhin gestattet oder dieses ausgeschlossen werden.

Als zweite Erweiterung besteht die Möglichkeit, einem lokalen Portlet einen Portlet Type Identifier zuzuweisen, der in den beiden höheren Stufen des Kontinuums der Portalkopplung zum Einsatz kommen kann (vgl. Abschnitt 4.5.4.5.2). Ein Dialog erlaubt die Auswahl eines im konfigurierten UDDI-Verzeichnis registrierten Portlet Type Identifiers. Zur Verwaltung der Portlet Type Identifier im UDDI-Verzeichnis sind in das Portlet Definition Repository entsprechende Funktionen eingebettet. Analog zur Definition und Synchronisation der Portaldefinition mit dem UDDI-Verzeichnis (vgl. Abschnitt 5.2.2.1) besteht auch hier die Möglichkeit, neue Portlet Type Identifier anzulegen, als Erzeuger bestehende zu verändern oder zu löschen und jeweils ein lokales Abbild aller definierten Portlet Type Identifier im Portlet Definition Repository vorzuhalten.

5.2.3.1.2 Erweiterung des lokalen Objektmodells

Wird von einem Portlet-Consumer ein Remote-Portlet erzeugt, so ist dieses sowohl vom Portlet-Consumer als auch vom Portlet-Provider entsprechend bei sich lokal abzubilden. Der Portlet-Consumer verwaltet eine lokale Instanz des Remote-Portlets, die ausschließlich als Verweis auf die im Portlet-Provider zur Verfügung gestellte Instanz des Remote-Portlets dient (vgl. Abschnitt 4.5.4.5.1). Hierzu ist es notwendig, das bestehende interne G8-Portal-Objektmodell anzupassen. Das Objektmodell einer Portlet-Instanz wird um die notwendigen Schlüssel zur Identifikation einer Portlet-Instanz für ein Remote-Portlet erweitert. Darüber hinaus muss die Möglichkeit hinzugefügt werden, dass bei der Löschung von Portlet-Instanzen entsprechende Löschmarkierungen beibehalten und diese erst nach Ablauf einer festzulegenden Zeitspanne endgültig entfernt werden (vgl. Abschnitt 4.5.2.2). Zur weitgehend transparenten Integration in die bestehende Architektur werden die beim Portlet-Provider erzeugten Remote-Portlet-Instanzen einem für das nachfragende Portal angelegten Home-Place hinzugefügt und in diesem verwaltet. Obwohl dieser Home-Place ausschließlich als Container und nicht zur direkten Anzeige dient, ermöglicht dieses Vorgehen die Beibehaltung und Nutzung des bestehenden Objektmodells.

Durch die Implementierung abgeschlossener Service-Klassen, welche die Methoden und Datenstrukturen des Portal Soap Gateway auf entsprechende lokale Varianten abbilden, kann

eine weitgehende Modularisierung und Entkopplung erreicht werden. Die zusätzlich realisierten Dienste unterscheiden sich entsprechend den Ausführungen in Abschnitt 4.6.2 nach Methoden, die Portlet-Consumern die notwendigen Informationen aus den Portlet-Konfigurationsdokumenten zur Verfügung stellen, und nach Methoden, die das Management und die Interaktion mit konkreten Portlet-Instanzen für Remote-Portlets erlauben.

5.2.3.1.3 Integration bestehender Content-Adaptoren

Die Schnittstelle zu den bestehenden, die eigentlichen Content-Pools anbindenden Content-Adaptoren konnte praktisch unverändert bleiben. Für die Content-Adaptoren ist es transparent, ob ein Aufruf zur Durchführung einer Aktion für ein lokal im Portal angezeigtes oder ein Remote-Portlet erfolgt. Durch die Anpassung der Implementierungen der Service-Klassen und -Methoden werden die Spezifika der entfernten Aufrufe für die Content-Adaptoren verborgen. Beispielsweise erzeugt die Service-Methode zur Generierung eines Links zur Durchführung einer Aktion auf dem Portlet – abhängig davon, ob es sich um ein lokales oder entferntes Portlet handelt – automatisch die richtigen Referenzen. Auch die Umsetzung der Namen der Benutzer, für die eine Aktion durchgeführt wird, wird transparent vom vorgelagerten Objektmodell behandelt. Dies erlaubt in den meisten Fällen die unveränderte Weiterverwendung bestehender Content-Adaptoren für die Nutzung beim Anbieten von Portlets und erfordert gleichermaßen keinen zusätzlichen Aufwand bei der Entwicklung neuer Content-Adaptoren.

5.2.3.2 Erweiterung zur Integration von Remote-Portlets

5.2.3.2.1 Einbettung in das Portlet Definition Repository und Konfiguration

Zur Sicherstellung einer einheitlichen Architektur und Verwaltung von Portlets wird die Konfiguration von Remote-Portlets, in Anlehnung an lokal zur Verfügung gestellte Portlets, in das Portlet Definition Repository integriert. Für alle definierten Portal Connections (vgl. Abschnitt 5.2.2.2) werden manuell oder zyklisch wiederkehrend alle entfernt zugreifbaren Portlets abgerufen und für diese entsprechende Konfigurationsdokumente gepflegt. Es sind drei grundlegende Situationen zu unterscheiden:

- Besteht für das Portlet noch kein Konfigurationsdokument, so wird dieses angelegt und die unten dargestellten Informationen darin abgelegt.

- Besteht für das Portlet bereits ein Konfigurationsdokument, so werden die in ihm enthaltenen Werte wenn notwendig aktualisiert.
- Besteht auf ein Portlet, für das ein Konfigurationsdokument vorhanden ist, kein Zugriff mehr, so sind zwei Situationen zu unterscheiden: Wird das Portlet im Portal nicht referenziert, ist das Konfigurationsdokument zu löschen. Wird das Portlet hingegen verwendet, so ist das Konfigurationsdokument als Platzhalter für das unbekannte Portlet beizubehalten, aber entsprechend zu kennzeichnen (vgl. Abschnitt 4.5.4.5.3).

Das Definitionsdokument beim Portlet-Consumer stellt kein direktes Abbild des Konfigurationsdokuments beim Portlet-Provider dar. Beispielsweise enthält dieses keine Informationen über die Anbindung an das zu integrierende Informationssystem oder vom Benutzer zu konfigurierende Portlet-Eigenschaften, da diese für den Portlet-Consumer irrelevant sind.

Die übertragenen und abgelegten Informationen umfassen drei wesentliche Bereiche. Bei dem ersten Bereich handelt es sich um Informationen, die für den Zugriff auf das Remote-Portlet notwendig sind, und um entsprechende Vorgaben bzgl. der sicherheitstechnischen Behandlung des Portlets. Eine Veränderung beider Aspekte kann ausschließlich über die Synchronisation erfolgen. Dadurch wird die Einhaltung der Sicherheitsvorschriften des Portlet-Providers durch den Portlet-Consumer gewährleistet. Der zweite Bereich umfasst sowohl initiale Werte für den Namen und die Kategorie des Portlets als auch die unterstützten Portlet-Modi und -Status (vgl. Abschnitt 4.2.2). Zuletzt besteht die Möglichkeit, lokal für das Remote-Portlet geltende Einstellungen der Zugriffsrechte und des zu verwendenden Cachings zu machen.

Zugriffs- und Sicherheitsinformationen

Zur Identifikation der Portlet-Konfiguration wird deren eindeutiger Portlet Definition Key herangezogen. Dieser wird für das lokale Konfigurationsdokument übernommen, so dass er sowohl beim Portlet-Provider als auch beim Portlet-Consumer übereinstimmt und als Schlüssel für die Erzeugung der Portlet-Instanzen eingesetzt werden kann. Die Speicherung der Portal-ID des Portals, von dem das Remote-Portlet bezogen werden kann, ermöglicht die Bestimmung der Adressen und Kommunikationsparameter der Portalschnittstellen mittels der zugehörigen Portal-Connection-Dokumente.

Die Sicherheitsinformationen (vgl. Abbildung 5-11) spiegeln wider, ob diese Portlet-Definition über den Portlet-Routing-Mechanismus (vgl. Abschnitt 5.2.3.2.3) an weitere Portale verteilt werden darf. Ist dies nicht der Fall, werden die Felder zur Definition der spezifischen Routing-Optionen ausgeblendet. Andernfalls können abhängig davon, ob Subject Switching

erlaubt ist, verschiedene Filter definiert werden, welche die Regeln für ein durchzuführendes Subject Switching festlegen. Soll die Portlet-Definition weiteren Portlet-Consumern zugänglich gemacht werden, so ist analog zur Veröffentlichung im Fall eines lokalen Portlets die Auswahl der generischen Benutzernamen der nachfragenden Portale vorzunehmen. Abschließend kann für die Consumer-Portale der Portlet-Definition geregelt werden, ob diese selbst erneut das Portlet mithilfe des Portlet-Routing-Mechanismus weiterverteilen dürfen.

The screenshot shows a configuration form for a portlet. It includes the following elements:

- Portlet-Routing erlaubt:** Radio buttons for 'Ja' (selected) and 'Nein', with '(nur lesend)' to the right.
- User-Switching erlaubt:** Radio buttons for 'Ja' (selected) and 'Nein', with '(nur lesend)' to the right.
- Lokale Regeln für User-Switching:** A text input field containing the rule '*Sales/Partner2/DE -> *Sales/Partner1/DE'. To the right of the field are three buttons: 'Hinzufügen', 'Editieren', and 'Löschen'.
- Remote Benutzer / -gruppen:** A dropdown menu showing 'Portal/Partner2/DE' and a 'Portal auswählen' button.
- Portlet-Routing erlauben:** Radio buttons for 'Ja' (selected) and 'Nein'.

Abbildung 5-11: Bereich zur Zugriffssteuerung auf ein Remote-Portlet in einem Portlet-Konfigurationsdokument

Name, Kategorie sowie Modi und Status

Im Standardfall werden der Name und die Kategorie der Definition des Remote-Portlets mit denen des Portlet-Providers abgeglichen. Der Abgleich kann jedoch für diese beiden Eigenschaften deaktiviert werden, um dem Portlet-Consumer die Möglichkeit zu geben, individuelle Einstellungen vornehmen zu können. Im Rahmen der aktivierten Portlet-Modi und -Status steht es dem Portlet-Consumer ebenfalls frei, diese für die lokale Nutzung gezielt zu deaktivieren; eine Aktivierung der beim Portlet-Provider deaktivierten Optionen ist ausgeschlossen.

Lokale Zugriffsrechte und Caching

Analog zu den Möglichkeiten eines lokalen Portlets, das für lokale Benutzer oder Gruppen zugreifbar gemacht werden kann, bestehen diese auch für Remote-Portlets. Durch die Zuordnung von Rechten für lokale Benutzer auf Remote-Portlets wird die Forderung nach Delegation der Zugriffsrechte auf die Consumer-Portale umgesetzt (vgl. Abschnitt 4.4.2.4.1). Zur Steigerung der Geschwindigkeit der Anzeige eines Remote-Portlets ist es dem Administrator möglich, eine Zeitspanne festzulegen, in der je Instanz des Portlets kein Abruf des Markups zur Anzeige im Portlet stattfindet, sondern in der dieser aus einem lokalen Cache entnommen wird. Die Konfiguration des Cachings ist im Wesentlichen abhängig von der Änderungshäufigkeit des Portlet-Inhalts und dem Grad der Notwendigkeit zur Anzeige aktueller Informationen.

der vom Remote Soap Adapter bei allen weiteren Operationen als Referenz auf das lokale Portlet an den Portlet-Provider übermittelt werden muss. Zur Durchführung lokal ausgelöster Aktionen findet die Kommunikation mit dem Portlet-Provider statt. Der Portlet-Provider sorgt für den korrekten und angepassten Aufbau des erzeugten Markups, wobei der Remote Soap Adapter die dazu benötigten Informationen beim Aufruf der entsprechenden Funktionen zur Verfügung stellt. Abbildung 5-13 zeigt beispielhaft einen Ausschnitt einer SOAP-Nachricht des Remote Soap Adapters zum Abruf des Markups für ein Remote-Portlet.

```
<ns5:getPortletMarkup soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns5="urn:g8-remote-portal-service">
  <requestData xsi:type="ns6:PortletRequestData" xmlns:ns6="http://g8.uni-paderborn.de/PortalSoapGateway">
    <consumerPortalID xsi:type="xsd:string">C1256DC90049829E</consumerPortalID>
    <remotePortletKey xsi:type="xsd:string">[3e3cf402ffa4cf77aca35fd59aed1fd31dc7d3fb]</remotePortletKey>
    <remoteUserToken xsi:type="xsd:string">CN=OLAF HAHNL/O=G8Engine/C=DE</remoteUserToken>
    <defaultLanguage xsi:type="xsd:string">DE</defaultLanguage>
    <device xsi:type="xsd:int">1</device>
    <language xsi:type="xsd:string">DE</language>
    <themePath xsi:type="xsd:string">themes/default</themePath>
    <urlPortletKey xsi:type="xsd:string">[66686676e253aa440d6d07e8bc3a9d8adaa36c64]</urlPortletKey>
    <currentPortletMode xsi:type="xsd:int">1</currentPortletMode>
    <windowMode xsi:type="xsd:int">1</windowMode>
    <previousWindowMode xsi:type="xsd:int">1</previousWindowMode>
    ...
  </requestData>
</ns5:getPortletMarkup>
```

Abbildung 5-13: Ausschnitt einer SOAP-Nachricht zum Abrufen des Markups eines Remote-Portlets

Der Remote Soap Adapter sorgt zusätzlich transparent dafür, dass abhängig von der Einstellung im Portlet-Definitionsdocument die Übermittlung des Namens des tatsächlichen die Aktion durchführenden Benutzers stattfindet.

5.2.3.2.3 Portlet-Routing

Als Erweiterung des Remote Soap Adapters unterstützt dieser nicht nur die direkte Kommunikation mit dem Portlet-Provider, sondern ist auch in der Lage, diese über zwischengeschaltete

Intermediär-Portlet-Provider zu realisieren. Hierzu findet eine einfache Fallunterscheidung statt. Entspricht die im Konfigurationsdokument des Remote-Portlets hinterlegte Portal-ID des Portals, das das Portlet dem Portlet-Consumer zur Verfügung stellt, der Portal-ID, die Bestandteil des Portlet Definition Keys ist und den tatsächlichen Portlet-Provider bestimmt, dann wird über den Portal Soap Gateway Proxy direkt das Portal Soap Gateway des Portlet-Providers angesprochen. Ist dies nicht der Fall, wird über den Portal Soap Gateway Proxy in die SOAP-Nachricht ein spezieller SOAP-Header (vgl. Abbildung 5-14) eingefügt, der Informationen über das anzusprechende Portlet beinhaltet. Die Nachricht selbst wird an den Portal Router Service des Portals gerichtet, das das Remote-Portlet dem Portlet-Consumer zur Verfügung stellt. Der weitere Ablauf ist für den Remote Soap Adapter transparent, folgt aber der in Abschnitt 5.2.1 dargestellten Struktur.

```
<ns4:g8Router soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
  xmlns:ns3="http://g8.uni-paderborn.de/PortalSoapGateway"
  xmlns:ns4="urn:g8-remote-portal-service"
  xsi:type="ns3:RouterData" soapenv:mustUnderstand="1">
  <ns4:forPortletDefKey xsi:type="xsd:string">
    C1256DF20036BDF9_D314AE38C4D43AC8C1256DF20047C15B
  </ns4:forPortletDefKey>
</ns4:g8Router>
```

Abbildung 5-14: Beispiel des SOAP-Headers zum Routing von Remote-Portlets

5.2.4 Anbieten von Seiten

Die zweite Stufe des Kontinuums der Portalkopplung umfasst das Anbieten von Seiten oder Plätzen zum ausschließlich lesenden Zugriff. Das Anbieten von Plätzen wird im G8-Portal, wie in der Konzeption vorgeschlagen (vgl. Abschnitt 4.5.4.2.2), als Spezialisierung der gemeinsamen Nutzung von Plätzen betrachtet, bei dem die Consumer-Portale durch die Rechtevergabe des Coordinator-Portals keine Veränderungsmöglichkeiten besitzen. Das Anbieten von Plätzen ist daher implizit Bestandteil der Betrachtung in Abschnitt 5.2.5. Die folgenden beiden Abschnitte berücksichtigen dementsprechend ausschließlich das Anbieten von Seiten.

Zur Realisierung der geforderten Funktionalität ist es möglich, hierfür spezielle Dialoge und eine vollständige Erweiterung des internen Objektmodells vorzunehmen. Dies führt zu einer deutlichen Erhöhung der Komplexität der Administrierung des G8-Portals und macht weitreichende und fehlerträchtige Änderungen der Gesamtarchitektur notwendig. Das im Framework des G8-Portals bereits vorhandene Konzept der Vorlagen (vgl. Abschnitt 5.1.1.2) stellt bereits eine ähnliche Funktionalität für den ausschließlich lokalen Einsatz zur Verfügung. Obwohl im allgemeinen Konzept zur Definition von anzubietenden Seiten Vorlagen im Sinne des G8-Portals kein Bestandteil sind, bieten diese alle notwendigen Voraussetzungen, um die zweite Stufe des Kontinuums zur Portalkopplung umsetzen zu können.

5.2.4.1 Vorlagen als Seiten beim Anbieter

Für die Administration auf Seiten des Anbieters müssen die bestehenden Dialoge zum Anlegen neuer und Editieren bestehender Vorlagen ausschließlich um die Möglichkeit zur Festlegung der generischen Benutzernamen der nachfragenden Portale, für die diese Vorlage zugreifbar sein soll, erweitert werden (vgl. Abbildung 5-15).

The image shows a screenshot of a web-based configuration interface titled "Eigenschaften der Seiten-Vorlage". The interface is organized into several sections:

- Position:** A text input field containing "1" and an "ID" field with a dotted pattern and "(nur lesend)" next to it.
- Sprache:** A dropdown menu set to "Deutsch" and a checked checkbox labeled "Sprache verwenden".
- Name:** A text input field containing "Premium-Artikel".
- Kategorie:** A text input field containing "Artikel".
- Zugriff für:** Three radio buttons: "Alle Benutzer" (selected), "Nur Notes-Nutzer", and "Nur Datenbank-Nutzer".
- Funktion als:** A dropdown menu set to "Für Benutzer oder Gruppe(n)".
- Seite:** A dropdown menu set to "*".
- Standard-Seite:** A dropdown menu.
- Nicht für Benutzer oder Gruppe(n):** A section with a "Seite:" dropdown menu and a "Standard-Seite:" dropdown menu.
- Remote-Seite:** A dropdown menu set to "Für Remote-Zugriff", which is highlighted with a red rectangle.
- Für anonymen Zugriff:** Two checkboxes: "Home" and "Nur als Standard-Seite benutzen (Feld 'Seite' ignorieren)".
- Optionen:** Four checkboxes: "Vorlage als direkten Link verwenden" (unchecked), "Seitenveränderungen erlauben" (checked), "Veränderung der Eigenschaften erlauben" (checked), and "Löschen der Seite erlauben" (checked). There is also an unchecked checkbox for "Feste Position".
- Vorlagen-Verwalter:** A dropdown menu.

Abbildung 5-15: Erweiterung von Vorlagen zum Anbieten von Seiten

Das interne Objektmodell der Vorlagen des G8-Portals wird um dieses Attribut erweitert. Es beinhaltet darüber hinaus neue Methoden, welche die Vorlagen zurückgeben, bei denen dieses Attribut gesetzt ist und die damit für den entfernten Zugriff verfügbar sind. In das Portal Soap Gateway wird eine Funktion aufgenommen, die ausschließlich die Beschreibungen der Vorlagen zurückliefert, auf die das nachfragende Portal Zugriffsrechte besitzt. Die Beschreibungen bestehen jeweils aus einer eindeutigen Vorlagen-ID, dem Namen der Vorlage, deren Aufbau aus Zeilen- und Spalten-Containern und den in ihnen enthaltenen Portlets.

5.2.4.2 Vorlagen als Remote-Seiten beim Nachfrager

Auf Seiten des Consumer-Portals wird der Portal Soap Gateway Proxy um das Gegenstück zum Abruf der verfügbaren Vorlagen beim Provider-Portal ergänzt. Die Synchronisation der verfügbaren Vorlagen kann entweder durch einen automatischen, zyklischen Prozess erfolgen oder durch den Administrator manuell im Dialog zum Verwalten der Vorlagen angestoßen werden. Die Synchronisation sorgt für die Eingliederung der entfernt verfügbaren Seiten in die normalen lokalen Vorlagen. Zu unterscheiden sind dabei die auch bei der Synchronisation der Portlet-Definitionen anzutreffenden drei Fälle Hinzufügen einer neuen, Aktualisieren und Löschen einer bestehenden Seite. Das interne Objektmodell muss zu diesem Zweck nur geringfügig angepasst werden. Da die Synchronisation dafür sorgt, dass entfernt zur Verfügung gestellte Seiten in gleicher Form wie lokale Vorlagen abgebildet werden, muss die Schnittstelle zwischen den Vorlagen und den Seiten in den Seitenregistern der Benutzer nicht verändert werden. Steht eine Vorlage nicht mehr zur Verfügung, so werden die auf ihr basierenden Seiten der Benutzer entsprechend gelöscht. Einzig die Aktualisierung bestehender Vorlagen mit dem veränderten Aufbau der entfernt zur Verfügung gestellten Seite ist zu ergänzen.

Der Dialog zum Editieren der vorhandenen Vorlagen wird entsprechend angepasst, so dass für eine Vorlage, die auf einer entfernt zur Verfügung gestellten Seite basiert, keine Festlegung des Zugriffs eines entfernten Portals möglich ist. Ein Routing von Seiten ähnlich wie bei Portlet-Definitionen ist nicht vorgesehen (vgl. Abschnitt 4.5.3). Zusätzlich sind Einstellungen, die für lokal definierte Vorlagen möglich sind, z. B. ob die Benutzer die Vorlage editieren können, für diesen Typ von Vorlage nicht zweckmäßig und daher auszublenden. Die Steuerung der Zugriffsrechte auf die Vorlage ist unabhängig davon, um welchen Typ von Vorlage es sich handelt. Durch die Zuordnung der Zugriffsrechte beim Consumer-Portal wird die Delegation der Rechtevergabe an dieses umgesetzt (vgl. Abschnitt 4.4.2.4.2).

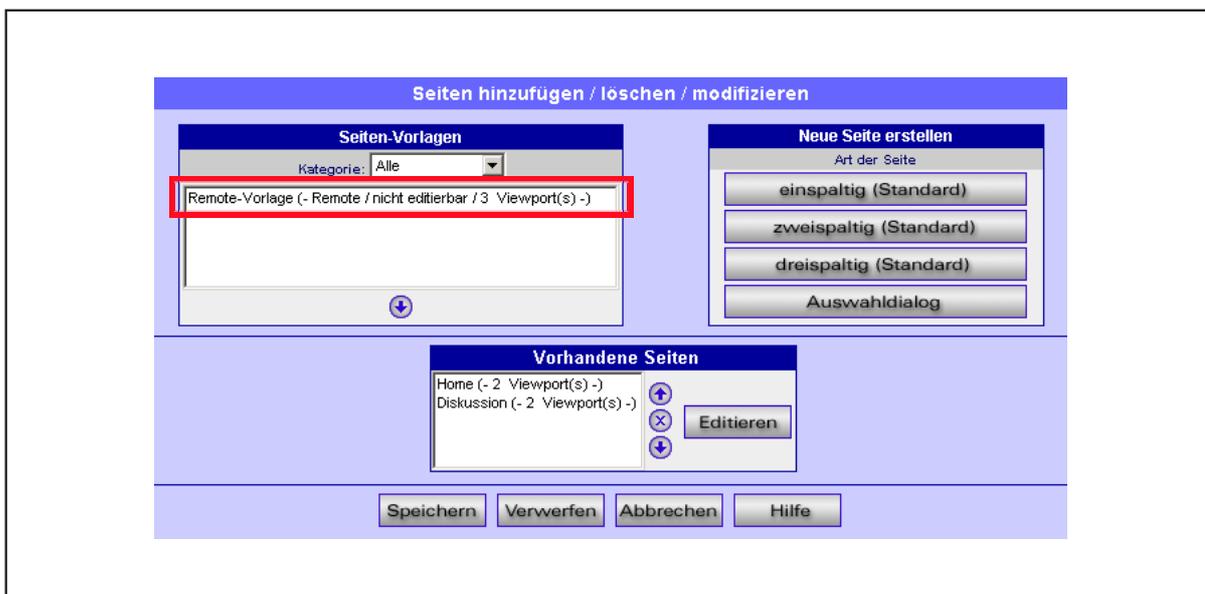


Abbildung 5-16: Integration von extern zur Verfügung gestellten Vorlagen in den Dialog zum Editieren des Seitenregisters

Ist die Vorlage durch berechtigte Benutzer hinzufügbare, so erscheint diese wie andere Vorlagen auch im Dialog zum Editieren des Seitenregisters des Home-Place (vgl. Abbildung 5-16). Da das G8-Portal-Framework für lokal gemeinsam genutzte Plätze aktuell keine Möglichkeit vorsieht, Seiten vollständig für die Bearbeitung zu sperren, sind entfernt zur Verfügung gestellte Seiten in diesen und damit auch in verteilten gemeinsam genutzten Plätzen momentan nicht einsetzbar. Hier sind weitere Überlegungen und Anpassungen am G8-Portal notwendig, um sie zumindest in lokal gemeinsam genutzten Plätzen einsetzen zu können. Abgesehen von einer (optionalen) visuellen Markierung, die offenbart, dass es sich um eine extern zur Verfügung gestellte Vorlage handelt, kann diese in gleicher Weise von den Benutzern in ihren Home-Place eingebracht werden.

Zur Behandlung von in der Seite befindlichen, aber lokal nicht verfügbaren Portlets wird zusätzlich zum Remote Soap Adapter (vgl. Abschnitt 5.2.3.2.2) der *Unknown Portlet Adapter* eingeführt. Wird bei dem Abruf der Definition der entfernt zur Verfügung gestellten Seite ein unbekanntes Portlet (Unknown Portlet) entdeckt, so wird für dieses ein spezielles Konfigurationsdokument im Portlet Definition Repository angelegt, das die wesentlichen Informationen dieses Portlets speichert. Als dem Definitionsdokument zugeordneter Content-Adapter fungiert der Unknown Portlet Adapter, dessen ausschließliche Aufgabe darin besteht, das Portlet konsistent zu den anderen Fällen im internen Objektmodell abzubilden (vgl. Abschnitt 4.5.4.5.3). Eine visuelle Anzeige innerhalb des Portals erfolgt nicht. Abhängig von der Konfiguration werden die Administratoren des Portals darüber informiert, dass ein unbekanntes Portlet entdeckt wurde. Dies gibt ihnen die Möglichkeit, entsprechend zu reagieren und dieses

Portlet verfügbar zu machen. Steht die zu dem Portlet gehörende Portlet-Definition nach einer Synchronisation der Portlet-Definitionsdocumente (vgl. Abschnitt 5.2.3.2.1) zu einem späteren Zeitpunkt zur Verfügung, wurde einer lokalen Portlet-Definition der verwendete Portlet Type Identifier zugeordnet oder wurde das Unknown Portlet in der Seite entfernt und nicht in weiteren Seiten verwendet, so wird das entsprechende Konfigurationsdokument des Unknown Portlets automatisch wieder entfernt.

5.2.5 Gemeinsame Nutzung von Plätzen

Die gemeinsame Nutzung von Plätzen zwischen miteinander gekoppelten Portalen stellt die letzte Stufe des Kontinuums der Portalkopplung dar.

5.2.5.1 Erweiterungen des internen Objektmodells

Die vorhandene Architektur des G8-Portals sieht ein internes Referenzmodell für die Strukturen eines Platzes inklusive seiner Seiten, deren Layout und den in ihnen enthaltenen Portlets vor. Von diesem Referenzmodell wird für die einzelnen Benutzer ein jeweils benutzerspezifisches Objektmodell abgeleitet, das fortlaufend mit dem Referenzmodell synchronisiert wird. Als Folge dieser Struktur wird das Referenzmodell als Ansatzpunkt für die Synchronisation der Meta-Struktur eines Platzes identifiziert. Dies ermöglicht deren vollständige Verteilung und die Umsetzung der in Abschnitt 4.4.2.1 aufgestellten technischen Anforderungen hinsichtlich der Unabhängigkeit von einem zentralen System sowie der Repliziertheit zur Steigerung der Verfügbarkeit und der Zugriffsgeschwindigkeit.

Direkt im Referenzmodell notwendige Änderungen beziehen sich auf die Möglichkeit, Löschemarkierungen und entsprechende Zeitstempel der letzten Änderung eines Elements vorhalten zu können, und auf für die Synchronisation des Layouts der Seiten notwendige Methoden. Darüber hinaus muss die Zuordnung der Koordinatorfunktion bei dem in Abschnitt 4.5.2.3 vorgestellten dezentralen Ansatz mit Koordinatorfunktion berücksichtigt werden. Eine zusätzliche Erweiterung erfordert die Integration der neuen Rechte von gekoppelten Portalen. Im Vergleich zur Gesamtarchitektur des bestehenden G8-Portals kann die Notwendigkeit zur Änderung des Objektmodells weitgehend minimiert werden. Dies gelingt durch die Auslagerung der eigentlichen Funktionalitäten zur gemeinsamen Nutzung eines Platzes in separate Komponenten und die Nutzung bereits vorhandener Methoden und Strukturen, wie etwa die Identifikation der Portalelemente durch eindeutige Schlüssel.

Dies trifft in besonderem Maße auf das interne Management eines Platzes zu. Da die Verteilung so realisiert wurde, dass lokal weiterhin ein vollständiges Referenzmodell vorhanden ist, verändert sich an diesem gegenüber der ausschließlichen Nutzung von Remote-Portlets nichts. Die Aktualisierung des Referenzmodells infolge der Verteiltheit wird ausschließlich von neuen, zusätzlichen Komponenten, transparent für die bestehende Architektur, vorgenommen.

Das Portal Soap Gateway und der Portal Soap Gateway Proxy sind mit entsprechenden Funktionen des Funktionsmodells (vgl. Abschnitt 4.6) zu versehen, welche die gezielte Synchronisation eines gemeinsamen Platzes unterstützen. Die eigentliche Implementierung dieser Funktionen erfolgt getrennt in eigenen Service-Komponenten. Die Umsetzung im G8-Portal berücksichtigt alle Aspekte zur Konfliktauflösung, zur Minimierung der Kommunikation und zur Unterstützung der Transparenz von lokalen und Remote-Portlets des in Abschnitt 4.5.4 festgelegten Synchronisationsmodells.

Werden im Verlauf der Synchronisation unbekannte Portlets entdeckt, so erfolgt die Sicherstellung der Wahrung der Konsistenz des Objektmodells analog zum in Abschnitt 5.2.4.2 beschriebenen Vorgehen unter Verwendung des Unknown Portlet Adapters.

5.2.5.2 Einbettung in die Benutzerschnittstelle

Da die gemeinsame Nutzung eines Platzes als Ausweitung des lokalen Zugriffs auf weitere Portale zu verstehen ist, wird diese folgerichtig in den Dialog zum Verwalten der lokalen Zugriffsrechte auf den Platz integriert (vgl. Abbildung 5-17).

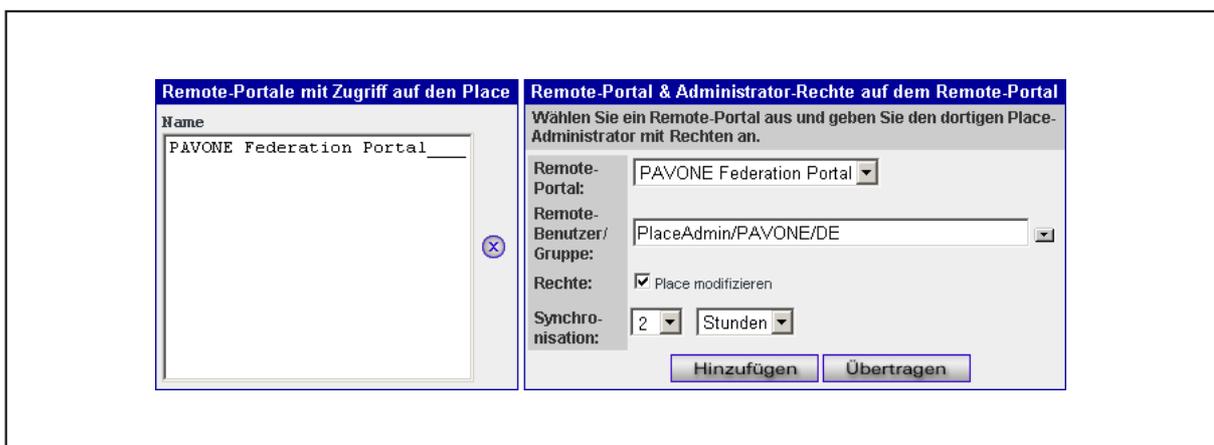


Abbildung 5-17: Ausschnitt aus dem Konfigurationsdialog des Coordinator-Place zur Definition von Consumer-Portalen

Entsprechend den aus Abschnitt 4.4.2.4.2 abgeleiteten Vorgaben besteht für lokale, gemeinsam genutzte Plätze die Möglichkeit, aus der Liste der im Portlet Definition Repository definierten Portale (vgl. Abschnitt 5.2.2.2) eines oder mehrere auszuwählen, die zu

Consumer-Portalen dieses Platzes werden sollen. Zu jedem Consumer-Place ist der vom Coordinator-Portal definierte Platz-Administrator des Consumer-Place festzulegen und zu bestimmen, ob dieser das Recht hat, den Platz zu verändern (vgl. Abschnitt 4.4.2.4.2). Zuletzt kann pro gekoppeltes Portal die Dauer zwischen zwei automatischen Synchronisationen der Meta-Strukturen des Platzes festgelegt werden. Hierüber hat das Coordinator-Portal die Möglichkeit, die Aktualität der Strukturen des Platzes und seine eigene Belastung durch durchzuführende Synchronisationen zu steuern. Die entsprechenden Parameter können vom Coordinator-Portal jederzeit geändert, neue Portale können hinzugefügt und Portalen, die bereits als Consumer-Portale fungieren, kann der Zugriff auf den Platz entzogen werden.

Die Nutzung des verteilten gemeinsam genutzten Platzes unterscheidet sich für die Benutzer des Coordinator-Place in keiner Weise von der eines ausschließlich lokal gemeinsam genutzten Platzes; sie ist vollständig transparent.

Auf Seiten des Consumer-Place besteht keine Möglichkeit, die gemeinsame Nutzung des Platzes auf weitere Portale auszuweiten. Der Forderung nach der Delegation der Vergabe der Zugriffsrechte auf die einzelnen beteiligten Portale nachkommend, ist es auf Seiten des Consumer-Place ausschließlich möglich, analog zu normalen lokalen gemeinsam genutzten Plätzen, für die lokalen Benutzer Zugriffsrechte zu definieren. Die Zugriffsrechte sind derart beschränkt, dass lokale Benutzer maximal dieselben Rechte haben dürfen wie der spezielle vom Coordinator-Place definierte Administrationsbenutzer. Die Implementierung stellt sicher, dass beim Entzug des Rechts zum Editieren des Platzes dieses auch allen anderen Anwendern entzogen wird, die dieses evtl. besitzen. Wird der Platz gelöscht, so wird auf Seiten des Consumer-Place ausschließlich die Teilnahme am gemeinsam genutzten Platz beendet, dieser bleibt auf Seiten des Coordinator-Portals weiterhin bestehen. Sollen Änderungen unmittelbar mit dem Coordinator-Place synchronisiert werden, kann über den Dialog eine sofort stattfindende manuelle Synchronisation angestoßen werden.

Für die Benutzer des Consumer-Place besteht gleichermaßen in der Verwendung des Platzes kein Unterschied gegenüber anderen lokal gemeinsam genutzten Plätzen. Die Verteilung ist auch für sie vollständig transparent.

Die einzig potenzielle Verletzung der Transparenz, die sowohl für das Coordinator- als auch für das Consumer-Portal in Betracht kommt, ist beim Auftreten und der automatischen Auflösung von Konflikten bei der Synchronisation eines verteilten, gemeinsam genutzten Platzes zu erwarten. Hierbei gibt es, wie in Abschnitt 4.5.4.2 dargestellt, keine Garantie dafür, dass das Ergebnis der Konfliktauflösung in allen Fällen den Erwartungen der Benutzer entspricht. In

diesen Fällen ist es die Aufgabe eines mit den entsprechenden Rechten ausgestatteten Benutzers des Coordinator- oder Consumer-Place, das angestrebte konsistente Erscheinungsbild des Platzes wieder herzustellen. Optional, jedoch aktuell nicht implementiert, könnten die Benutzer mit Administrationsrechten für den Platz im Konfliktfall benachrichtigt werden, damit sie schnell reagieren können.

5.3 Zusammenfassung

Mit der Erweiterung des vorhandenen G8-Portal-Frameworks um die Komponenten und Dienste zum Aufbau und zum Betrieb einer Portalkopplung über alle Stufen des Kontinuums der Portalkopplung hinweg wird eine informationstechnische Realisierung des in Kapitel 4 entwickelten abstrakten Konzepts vorgenommen. Durch die flexible Architektur des G8-Portals und die implementierungsneutrale Spezifikation ist eine vollständige Umsetzung aller Anforderungen durch evolutorische Schritte möglich. In Anlehnung an das konzeptionelle Ebenen-Modell mit seinen zwischengeschalteten Prozessoren (vgl. Abschnitt 4.4.2.2) kann die Umsetzung weitgehend als zusätzliche, auf bestehenden Funktionen aufbauende Komponente realisiert werden. Dies ist als Indiz dafür zu werten, dass eine Realisierung gekoppelter Portale keine vollständige Neuentwicklung von Portal-Frameworks notwendig macht, sondern dass bestehende um die notwendigen Komponenten und Dienste erweiterbar sind.

Anhand umfangreicher interner, manueller und automatischer Tests wurde eine weitgehende funktionale Validierung vorgenommen, so dass der zweite Bestandteil des in Abschnitt 2.3.1 formulierten Ziels, die informationstechnische Realisierung, als erreicht angesehen werden kann. Gleichzeitig bildet die funktionale Validierung die Grundlage für die abschließende anwendungsorientierte Validierung in einem praktischen Szenario.

6 Praktische Nutzung gekoppelter Portale – Anwendungsszenario PAVONE AG

Nachdem in Kapitel 4 die Operationalisierung des Begriffs föderierter Portale und die Konzeption einer Architektur für diese vorgenommen wurde, die beide dem Bereich des domain engineering (vgl. Abschnitt 4.1) zuzuordnen sind, und im vorangegangenen Kapitel 5 eine dem application engineering zuzurechnende informationstechnische Umsetzung erfolgte, verbleibt als letzter Aspekt des in Abschnitt 2.3.1 formulierten Forschungsziels die anwendungsorientierte Validierung in Form einer Fallstudie oder eines Anwendungsszenarios.

6.1 Vorbemerkungen

Im Kontext dieser Arbeit ist als Voraussetzung für die Umsetzung einer Fallstudie im Bereich gekoppelter und föderierter Portale das Vorhandensein von Installationen der erweiterten Architektur des G8-Portals bei mindestens zwei Unternehmen oder Unternehmenseinheiten zu nennen. Da es sich beim G8-Portal als universitärem Forschungsprojekt eines Workplace Portals nicht um eine kommerzielle Software handelt, ist eine direkte Verbreitung im unternehmerischen Umfeld nicht gegeben. Darüber hinaus ist die Einführung eines Portals, wie die Ausführungen im Abschnitt 2.2 dargestellt haben, nicht ein punktuelles Standardvorhaben, sondern ein viele Bereiche einer Unternehmung berührendes, an sich bereits hochkomplexes Einführungsprojekt. Zusammengenommen macht es dies weitgehend unmöglich, mehrere bereits miteinander kooperierende Partner zu gewinnen, die bereit sind, ein Portal auf Basis des G8-Portals zu installieren, einzurichten und zu betreiben sowie dieses zusätzlich mit den G8-Portalen der anderen Partner zu koppeln.

Aus diesem Grund war es notwendig, die Ziele bei der anwendungsorientierten Validierung zu reduzieren. Da sich die angestrebte Durchführung einer oder mehrerer Fallstudien zum aktuellen Zeitpunkt als nicht realisierbar darstellte, wurde mit einem Partnerunternehmen exemplarisch ein im Folgenden vorzustellendes praktisches Anwendungsszenario umgesetzt.

6.2 Rahmenbedingungen

Seit mehreren Jahren besteht zwischen dem Groupware Competence Center (GCC) der Universität Paderborn und der in Paderborn ansässigen PAVONE AG, einem mittelständischen Software- und Beratungsunternehmen, eine Entwicklungspartnerschaft im Bereich von Business- und Workplace-Portalen. Grundlage dieser Partnerschaft ist das Framework des

G8-Portals, auf dessen Basis die PAVONE AG das PAVONE-Portal anbietet. Zielgruppe sind kleine bis mittlere Unternehmen, die eine IT-Infrastruktur mit hohem Anteil von Lotus-Domino-Anwendungen aufweisen. Aufgrund des aktuellen Status der Implementierung der Portalkopplung als Entwicklungsprojekt, bei dem Auswirkungen auf die Stabilität des produktiven Einsatzes der Standardfunktionen nicht ausgeschlossen sind, kam eine Fallstudie basierend auf der installierten Basis des PAVONE-Portals ebenfalls nicht in Frage.

Das an der Entwicklungspartnerschaft beteiligte Team umfasst auf Seiten der PAVONE AG einen Kreis von 2-3 Mitarbeitern und auf Seiten des Groupware Competence Centers der Universität Paderborn 1-2 Mitarbeiter, die durch eine wechselnde Zahl von Studierenden unterstützt werden. An beiden Standorten sind jeweils aktuelle Versionen des G8-Portal-Frameworks installiert, die vor allem zu Demonstrations- und Testzwecken und der Entwicklung im Einsatz sind. Die Kommunikation innerhalb des Teams wird bisher zum größten Teil über E-Mail und die teilweise gemeinsame Nutzung einer Lotus-Notes-Datenbank realisiert.

Zu Evaluationszwecken und zur besseren Strukturierung der gemeinsamen Bemühungen wurde ein über die Grenzen eines Partners hinausgehender verteilter gemeinsam genutzter Platz eingeführt. Anlass war die Verfügbarkeit der Erweiterung des G8-Portal-Frameworks um die Option zur Kopplung von Portalen. Ziel bei der Gestaltung des Platzes war die Integration aller für die Kommunikation, Koordination und Kollaboration im Kontext der Partnerschaft benötigten Informationen und Anwendungen. Hierbei konnte auf die bereits bei beiden Partnern vorhandenen, im Wesentlichen Notes-basierten Quellen zurückgegriffen werden. Diese waren durch die Synchronisation des gemeinsam genutzten Platzes beiden Partnern zur Verfügung zu stellen.

Um im ersten Schritt mögliche negative Effekte zu vermeiden, wurde bei beiden Partnern zusätzlich zu den bisher bereits vorhandenen Portalinstallationen eine auf dem erweiterten G8-Portal-Framework basierende Installation vorgenommen. Es ist geplant, den Parallelbetrieb zu beenden und eine Migration zur erweiterten Version vorzunehmen, sobald sich diese – auch im praktischen Einsatz – als ausreichend robust und stabil erwiesen hat. Am GCC wurde zusätzlich eine private UDDI eingerichtet, die zur Verwaltung der Portalregistrierungen etc. (vgl. Abschnitt 5.2.2) verwendet wird. Die bereits vorhandenen Definitionen der Portlets konnten prinzipiell direkt für die zusätzliche Installation übernommen werden und waren ausschließlich um die Zugriffsberechtigung des Partnerportals zu ergänzen (zu Einschränkungen siehe Abschnitt 6.4). Mit dem Anlegen jeweils eines lokalen Benutzers für den Zugriff durch das Partnerportal, der Registrierung des Portals in der privaten UDDI und der

Einrichtung der jeweiligen Portal-Connection-Dokumente war die einmalig notwendige Konfiguration abgeschlossen. Die Teilnehmer selbst waren jeweils ausschließlich in ihren lokalen Systemen als Benutzer registriert und den Systemen des Partners daher unbekannt.

6.3 Aufbau und Inhalt des gekoppelten Platzes

Der verteilte gemeinsam genutzte Platz setzt sich aus sechs Seiten und insgesamt 15 Portlets zusammen. Die Abbildung 6-1 zeigt den Inhalt der vierten Seite des Platzes sowohl aus Sicht des Portals der PAVONE AG als auch aus Sicht des Portals des GCCs.

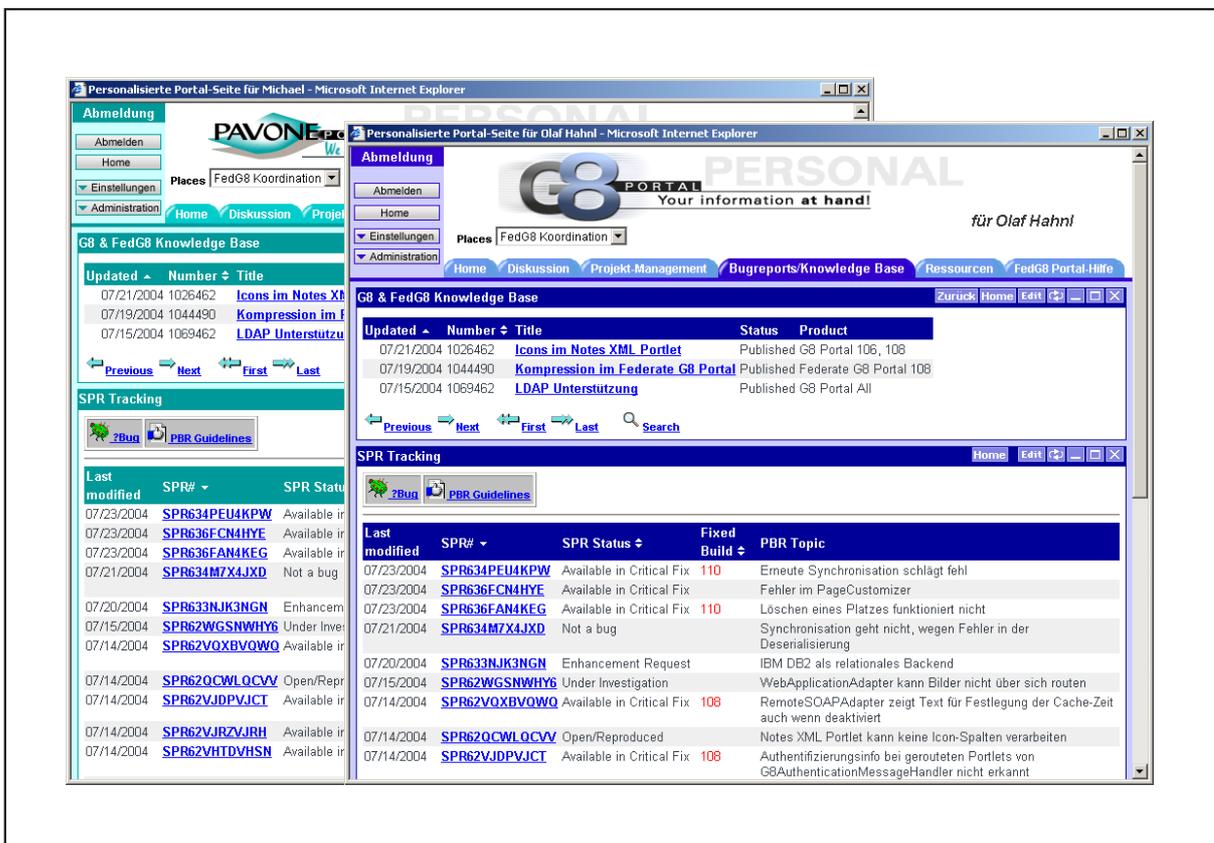


Abbildung 6-1: Ansicht des gekoppelten Platzes des Anwendungsszenarios

Thematisch finden sich in dem Platz alle kooperationsrelevanten Bereiche wieder:

1. Die Startseite (Home) enthält einen Gruppenkalender, der die Terminplanung unterstützt. Dazu kommen Portlets, die Nachrichten aus verschiedenen Bereichen anzeigen, z. B. aus dem Marketing und der Entwicklung.
2. Als wesentliches Element der internen Kommunikation dient das auf der zweiten Seite (Diskussion) eingebundene Diskussionsforum. Dieses steht allen Teilnehmern offen und deckt alle thematischen Bereiche ab.

3. Zur strukturierten Steuerung des Kooperationsprojekts ist die dritte Seite (Projekt-Management) vorgesehen. In ihr sollten basierend auf PAVONE Project Management die Projektplanungs- und Projektsteuerungsanwendungen integriert werden. Dies konnte nur teilweise realisiert werden. Die Portlet-basierte Anzeige der Projektpläne als einfache Textansichten ist umgesetzt. Weitergehende Visualisierungen waren aktuell nicht einzubinden, da diese auf Applets basieren, die, wie im folgenden Abschnitt beschrieben, Probleme bei der Verteilung bereiten. Zusätzlich sind Teile einer auf PAVONE Enterprise Office basierenden Office-Umgebung integriert. In dieser können Adressen von Partnern und Kunden und die mit ihnen geführte Korrespondenz abgelegt und verwaltet werden.
4. Als Feedback-Element für die Entwickler und Unterstützung für die Mitarbeiter, die sich mit der Installation und dem Einsatz des G8-Portals beschäftigen, stellt die vierte Seite (Bugreports/Knowledge Base) zwei Portlets bereit. Das erste Portlet dient dazu, neue Software Problem Reports (SPRs) erzeugen und nach bestehenden recherchieren zu können. Diese werden vom Entwicklungsteam untersucht und die Fehler ggf. im weiteren Verlauf der Entwicklung beseitigt. Das Knowledge Base Portlet zeigt Berichte, die Informationen zu verschiedenen Bereichen des G8-Portals enthalten. Hierzu gehören gleichermaßen Erklärungen zu bekannten Fehlern und Einschränkungen, Case Studies über den bisherigen Einsatz, Installationshinweise etc.
5. Über die fünfte Seite (Ressourcen) sind intern für das Projekt verfügbare Ressourcen zugreifbar. Dazu zählen z. B. Software-Tools, Lizenzschlüssel und Zugangsdaten zu den Verwaltungssystemen des Quellcodes.
6. Die letzte Seite (FedG8 Portal-Hilfe) stellt die internen Hilfedatenbanken zur Verfügung. Diese umfassen Informationen zur Administration der verschiedenen Portlets, des Portals selbst sowie für die Entwickler relevante Interna der Portalarchitektur.

6.4 Erfahrungen

Der Mehraufwand gegenüber der Installation eines nicht für die Kopplung vorgesehenen Portals beschränkte sich auf die initiale Konfiguration der eigentlichen Portalkopplung und war aufgrund der Unterstützung durch die Notes-basierte Administrationsumgebung gering.

Die Auswahl der für den Partner zur Verfügung gestellten Portlets stellte sich als problematischer dar als angenommen. Dies hing in besonderem Maße mit der bei beiden Partnern vorherrschenden Lotus-Domino-Infrastruktur zusammen. Die bisher ausschließlich für den loka-

len Einsatz vorgesehenen Portlets sind darauf beschränkt, eine Übersicht über die Dokumente einer Datenbank anzuzeigen. Wird ein Dokument selbst geöffnet oder soll ein neues angelegt werden, so wird ein neues Browser-Fenster geöffnet, das am Portal vorbei direkt mit dem Lotus-Domino-Server kommuniziert. Im konkreten Fall war es zwar möglich, den Lotus-Domino-Server auch direkt vom Partner aus anzusprechen – dies wird jedoch nicht immer der Fall sein –, aufgrund der fehlenden Vermittlung durch das Partnerportal bestand jedoch kein Sicherheitskontext, der es dem unbekanntem externen Benutzer ermöglicht hätte, die entsprechende Aktion durchzuführen. Aus diesem Grund mussten diese Portlets (z. B. zur Verwaltung der SPRs und zum Zugriff auf die Knowledge Base) auf einen anderen Typ umgestellt werden, der eine vollständige Darstellung und Ausführung von Aktionen innerhalb des Portlets erlaubt, aber andere spezifische Einschränkungen aufweist. Dieser Portlet-Typ agiert als Proxy zwischen der eigentlichen Informationsquelle und dem Portal. Dadurch ist es möglich, den Sicherheitskontext durchgängig zu übermitteln. Die notwendige Transformation der Inhalte, um diese vollständig innerhalb eines Portlets darstellen zu können, hat jedoch Grenzen; z. B. können Frames oder komplexer JavaScript-Code nicht korrekt verarbeitet werden. Ein ähnlich gelagertes Problem stellte die Einbindung von Applets dar. Diese konnten nicht verteilt werden, da es ohne Änderung der Applets selbst nicht möglich war, deren Abruf und ihre Kommunikation mit dem Ursprungsserver über das Portal umzuleiten. Ein letztes Problem, das sich wiederkehrend ergab, bestand darin, dass das Partnerportal zwar den Namen des Benutzers kennt, für den eine Aktion ausgeführt wird, die bestehenden Portlets aber nicht in der Lage sind, diesen korrekt an den Lotus-Domino-Server als Back-End-System zu übergeben. Dies hat zur Folge, dass alle Aktionen, die durch Benutzer des Partnerportals durchgeführt werden, unter der Pseudoidentität des Partnerportals erfolgen. Im Sinne der Vermeidung der Sichtbarkeit der internen Organisation (vgl. Abschnitt 4.4.1.2) kann dies sogar wünschenswert sein. Im konkreten Fall, bei dem eine enge Bindung mit großem Vertrauen bestand, wurde dies jedoch als Einschränkung empfunden.

Diese Problembereiche sind in weiten Teilen nicht neu und auch nicht ausschließlich spezifisch für gekoppelte Portale. Es handelt sich vielmehr um bekannte Fragestellungen im Zusammenhang mit Technologien zur vollständigen Integration externer Systeme innerhalb von Portlets. Bei gekoppelten Portalen ist eine Umgehung aber wesentlich erschwert oder sogar unmöglich, so dass sie verstärkt sichtbar werden. Als Zielsetzung auch bei der Entwicklung neuer Content-Adaptoren ist daher darauf zu achten, dass eine vollständige Abbildung einer Informationsquelle oder Anwendung durch das Portlet unter Vermittlung des Portals stattfindet. Der direkte Abruf von Informationen durch den Browser des Benutzers unter

Umgehung des Portals ist unbedingt zu vermeiden, wenn ein Portlet dieses Typs auch anderen Portalen zur Verfügung gestellt werden soll. Der in Abschnitt 5.2.3.1.3 herausgestellte transparente Einsatz von bestehenden Content-Adaptoren für das Anbieten von Portlets ist in den meisten Fällen möglich, kann abhängig vom angebotenen Content-Pool aber Erweiterungen erfordern. In diesem Sinne wurde exemplarisch der als Proxy fungierende Content-Adaptor um die Möglichkeit erweitert, aus dem übertragenen Benutzernamen ein Sicherheitstoken zu erzeugen. Dieses ermöglicht es, auf die Lotus-Domino-Server optional mit der tatsächlichen und nicht ausschließlich mit der Pseudoidentität zuzugreifen. Im Fall des Diskussions-Portlets war so z. B. eine eindeutige Zuordnung eines neu erstellten Diskussionsbeitrags zu einem Benutzer möglich. Dies vereinfachte einerseits die Kommunikation, andererseits konnte so sichergestellt werden, dass nur derjenige Benutzer das Dokument modifizieren kann, der dieses tatsächlich angelegt hat.

Auf der Ebene der verteilten gemeinsamen Nutzung eines Platzes traten keine wesentlichen Probleme auf. Es war zu beobachten, dass die Änderungshäufigkeit nach dem initialen Einrichten der Seiten, deren Layouts und der Portlets schnell abnahm. Zu Beginn war beiden Partnern parallel die Möglichkeit gegeben worden, ihre Portlets in den Platz einzubringen. Dies führte zu gleichzeitigen Änderungen, die jedoch nur selten zu wirklichen Konflikten führten, da bereits im Vorfeld Absprachen über den groben Aufbau des Platzes getroffen wurden, jeder Partner den ihm zugewiesenen Bereich bearbeitete und die Anzahl der Portlets gering war. Die wenigen auftretenden Konflikte konnten durch die integrierten automatischen Konfliktauflösungsstrategien weitgehend zur Zufriedenheit gelöst werden, erforderten wie erwartet teilweise jedoch den manuellen Eingriff eines Benutzers. Für die Startphase, besonders wenn mehrere Partner beteiligt sind und die Anzahl der Seiten und Portlets größer ist, eignet sich daher ein zweiter Ansatz zur Vermeidung von Konflikten (vgl. auch Abschnitt 4.5.4.3) besser. Bei diesem erhalten die Partner nacheinander das Recht, ihre Portlets in den Platz einzubringen. Nachdem der Ausgangszustand des Platzes erreicht ist, kann, wenn dies gewünscht wird, entsprechend mehreren oder allen Partnern das Recht zum Editieren zurückgegeben werden.

Aufgrund der abnehmenden Änderungshäufigkeit stellte die potenzielle Asynchronität der Meta-Struktur des Platzes zwischen den Partnern kein Problem dar. Eine mehrmals täglich ablaufende automatische Synchronisation war vollständig ausreichend, um den gemeinsamen Kontext zu erhalten und gleichzeitig eine unnötig große Last zu vermeiden. Es war zu beobachten, dass in den meisten Fällen nach wesentlichen Änderungen eine manuelle Synchronisation angestoßen wurde, die unmittelbar zu einem konsistenten Zustand führte. Die Effektivität

tät des Vorgehens ist jedoch bei Kopplungen mit mehr als zwei Portalen deutlich eingeschränkt und daher als Sonderfall und nicht als universelle Lösung zu betrachten.

Durch die delegierte Rechtevergabe an die lokalen Portale (vgl. Abschnitt 4.4.2.4) entstand für die Administratoren kein höherer Administrationsaufwand, als dies bei einem lokalen gemeinsam genutzten Platz der Fall ist. Für die Benutzer selbst war es – abgesehen von der oben dargestellten Problematik bei der Zuordnung der Pseudoidentität anstelle des konkreten Benutzernamens bei der Arbeit mit Notes-Dokumenten – praktisch transparent, dass sie mit einem zwischen zwei Partnern verteilten Platz und in ihm enthaltenen Remote-Portlets arbeiteten.

Bei der ersten installierten Version beeinträchtigten jedoch längere Verzögerungen bei der Anzeige von Remote-Portlets die Nutzbarkeit. Diese ließen sich nach einer Analyse im Wesentlichen auf zwei Faktoren zurückführen. Die Hardware der Testsysteme war bzgl. Prozessor und Arbeitsspeicher insgesamt dem unteren Limit der für den Betrieb von Portalen gegebenen Anforderungen zuzuordnen. Die Anbindung des Servers der PAVONE AG erfolgte gegenüber den bisherigen internen Tests im Rahmen der funktionellen Validierung nicht über ein lokales Netzwerk, sondern über das Internet mit geringeren Übertragungsraten und höheren Latenzzeiten. Dies führte zu einer merklichen Verlängerung der Übertragungszeiten. Konkrete Aussagen zu benötigter Minimalbandbreite und maximal tolerablen Latenzzeiten sind bisher wegen der beschränkten Aussagekraft des Szenarios nicht möglich. Da die verringerte Bandbreite gegenüber einem lokalen Netzwerk beim praktischen Einsatz realistisch ist, wurde eine Anpassung der Implementierung durchgeführt. Diese wurde durch die hohe Flexibilität des eingesetzten AXIS-Frameworks (vgl. Abschnitt 5.1.2.5) wesentlich erleichtert. Die Integration einer optionalen Kompression komprimiert die verschickten XML-basierten Web-Service-Nachrichten transparent um bis zu 50 % und mehr. Dies erfordert zwar sowohl beim Sender als auch beim Empfänger zusätzliche Prozessorleistung, führte aber bereits auf der vorhandenen Hardware zu einer ausreichenden Verkürzung der Antwortzeit. Nach dieser Maßnahme wurden die Verzögerungen beim Einsatz von Remote-Portlets für ein Testsystem allgemein als akzeptabel bewertet. Letzte Performance-Verbesserungen sind bei der aktuellen Implementierung durch die auf den Anwendungsfall abgestimmte Wahl von Caching-Einstellungen möglich.

Eine darüber hinausgehende weitere deutliche Verkürzung der Gesamtantwortzeit könnte die Weiterentwicklung des Basis-Frameworks des G8-Portals ermöglichen. Zurzeit wird die Ausgabe jedes Portlets sequenziell berechnet, so dass sich die Gesamtzeit zur Anzeige einer

Seite im Wesentlichen aus der Summe der Zeiten zur Berechnung der Anzeige aller Portlets der Seite ergibt. Dies führt bei den prinzipbedingt längeren Berechnungszeiten von Remote-Portlets zu deutlich längeren Gesamtantwortzeiten. Bei einer parallelen Abarbeitung wäre diese im Idealfall nur so lang wie die längste zur Generierung der Ausgabe eines Portlets benötigte Zeit. Gleichwohl dieser Idealfall praktisch nicht zu erreichen ist, kann die parallele Abarbeitung der Portlets die Transparenz der Nutzung von Remote-Portlets weiter verbessern. Gegebenenfalls zu berücksichtigen sind jedoch Auswirkungen der parallelen Abarbeitung auf die Prozessorauslastung.

Für die im gewählten Anwendungsszenario getestete Umgebung ergab sich insgesamt durch die grundsätzliche gemeinsame Nutzung der Informationsquellen und Anwendungen eine Reduktion der direkten E-Mail-Kommunikation und eine Verbesserung und Erleichterung der Zusammenarbeit in dem verteilten Projektteam.

6.5 Limitationen

Aufgrund der eingangs dargestellten Ausgangsbedingungen handelt es sich bei dem vorgestellten Anwendungsszenario um einen ersten, mit eingeschränkter Aussagekraft versehenen Schritt zur anwendungsorientierten Validierung der entwickelten Portalkopplung zwischen zwei oder mehreren G8-Portalen. Für die weitere Validierung ist die Ausweitung der Anzahl der beteiligten Partner und der Benutzergruppe bei jedem einzelnen Partner anzustreben. Verlässlichere Aussagen hinsichtlich qualitativer Erfahrungen sind erst von der Nutzung im operativen Tagesgeschäft zu erwarten. Die gelegentliche und punktuelle Nutzung, wie im Rahmen des Anwendungsszenarios, kann hier nur Tendenzen und Indizien liefern.

Auch funktionell wurden nicht alle Bereiche berücksichtigt. Die Verwendung gerouteter Portlets war ebenso wenig notwendig wie der Einsatz von Portlet-Ersetzung durch Portlet Type Identifier. Dies lässt sich unmittelbar auf die Beschränkung auf zwei Partner und den konkreten Fokus des Platzes zurückführen. Das Anwendungsszenario berücksichtigt aufgrund der Art der Kooperation die zweite Stufe des Kontinuums der Portalkopplung nicht. Deren Nutzen verspricht gerade bei wiederkehrenden Standardaufgaben, etwa der Bestellung von Waren oder dem Abfragen des Status des Versands der Waren beim Lieferanten, besonders hoch zu sein. Diese Vermutung ist ebenfalls in weiteren Untersuchungen zu überprüfen.

Ergänzend ist festzuhalten, dass Langzeiterfahrungen in der Nutzung fehlen. Dies trifft aber gleichermaßen auch auf den Einsatz nicht gekoppelter Portale zu.

6.6 Zusammenfassung

Das umgesetzte Anwendungsszenario hat im Rahmen einer zwar sowohl inhaltlich als auch personell eingeschränkten, praktischen Nutzung erste Erfahrungen über die Anwendbarkeit gekoppelter Portale ergeben. Diese lassen sich als durchweg positiv zusammenfassen, wobei auch bisher nicht oder nicht ausreichend berücksichtigte Problemfelder aufgedeckt wurden. Für die Nutzer ergab sich eine Verbesserung für das konkrete Anwendungsszenario, so dass das Grundanliegen der Kopplung von Portalen als erfüllt angesehen werden kann.

Gleichwohl muss eine weitere Erprobung Erkenntnisse über das genaue Nutzungsverhalten und daraus abzuleitende Verbesserungen bringen. Aufgrund der weitgehenden Transparenz der Kopplung waren jedoch, wie prognostiziert, keine negativen spezifischen, über die bereits vom Einsatz von Portalen hinausgehenden Benutzerreaktionen festzustellen. Dies bedeutet konkret, dass zu den bekannten Problemen der teilweise eingeschränkten Benutzerschnittstelle webbasierter Lösungen keine weiteren für die Kopplung spezifischen hinzukamen, welche die Akzeptanz einer derartigen Lösung negativ beeinflussen würden. Verbesserungen in diesem Bereich kommen nicht nur nicht gekoppelten Portalen, sondern gleichermaßen auch gekoppelten Portalen zugute.

Mit der erfolgreichen Durchführung einer ersten anwendungsorientierten Validierung wurden alle Teilaspekte des Forschungsziels im Verlauf der Kapitel 4, 5 und 6 berücksichtigt und stringent abgearbeitet.

7 Schlussbetrachtung und Ausblick

Die vorliegende Arbeit beschäftigt sich mit der Frage, wie das Potenzial der Nutzung von Portalen bei deren zunehmender intra- wie interorganisationaler Proliferation durch die Föderierung von Portalen nachhaltig gesichert werden kann. Das den Portalen grundsätzlich zukommende Potenzial, wird im Zusammenhang mit aktuellen Entwicklungen im Unternehmensumfeld dargestellt. Als Grundlage der Betrachtung der Föderation von Portalen selbst findet eine Auseinandersetzung mit der Bedeutung und den Eigenschaften des Begriffs Föderation an sich statt. Diese konzentriert sich im Wesentlichen auf Föderationen im Bereich der IT, bezieht jedoch gleichermaßen den Ursprung des Begriffs und dessen Verwendung in der Betriebswirtschaftslehre mit ein.

Basierend auf diesen Erkenntnissen kann die eigentliche Zielsetzung, die Föderierung von Portalen, verfolgt werden. Die Identifikation einer geeigneten Entwurfsmethode und die Herausarbeitung der gemeinsamen Basiskomponenten unterschiedlicher Portal-Frameworks ergänzen die vorbereitenden Darstellungen. Die Operationalisierung des Begriffs der Föderierung von Portalen und die Identifikation sowohl betriebswirtschaftlicher wie technischer Anforderungen an diese resultieren im Entwurf einer Kommunikations- und Synchronisationsarchitektur sowie eines Daten- und Funktionsmodells. Das so entstandene Modell wird informationstechnisch durch Erweiterung eines bestehenden Portal-Frameworks auf Basis der Web-Service-Technologie umgesetzt. Ergänzend zu der funktionalen findet eine anwendungsorientierte Validierung in Form eines mit einem Kooperationspartner praktisch umgesetzten Anwendungsszenarios statt.

7.1 Ausblick

Mit dieser Arbeit im Bereich der Kopplung von Portalen, die nach Kenntnis des Autors erstmals sowohl theoretische als auch praktische Aspekte umfasst, wird ein Grundstein gelegt, auf dem weitere Forschung aufbauen kann. Bis zur kommerziellen Verfügbarkeit von Produkten, die eine Kopplung von Unternehmensportalen ermöglichen, sind weitergehende und verstärkte Forschungen in zahlreichen Gebieten erforderlich. Einige bisher nicht oder nicht ausreichend gelöste Aspekte, vor allem technischer Art, sind schon jetzt bekannt, andere werden erst durch den weiteren Einsatz des entwickelten Konzepts und darauf basierenden Umsetzungen in größeren und unterschiedlich strukturierten Fallstudien aufgedeckt werden.

Die folgenden acht Abschnitte geben einen Überblick über die dem Autor bekannten und von ihm als besonders relevant eingestuften zukünftigen Forschungsgegenstände.

7.1.1 Soziologische und betriebswirtschaftliche Aspekte

Ein Themenkomplex, der zu neuen Fragestellungen führen wird, ist die soziologische Erforschung des Nutzungsverhaltens von Portalen und speziell von gekoppelten Portalen durch die Mitarbeiter als Plattform der gemeinsamen Informations- und Wissensverteilung sowie -nutzung. Mithilfe soziologischer Methoden kann untersucht werden, inwiefern die erwarteten Verbesserungen tatsächlich realisiert werden können bzw. welche Maßnahmen zu deren Realisierung notwendig sind. Diese Methoden wurden jedoch im Rahmen der vorliegenden Arbeit explizit ausgeklammert (vgl. Abschnitt 2.1.1.4 und 3.3) und daher nicht näher betrachtet.

Aus betriebswirtschaftlicher Sicht ist zudem zu betrachten, wie eine Verankerung der neuen Kooperations-, Koordinations- und Kollaborationsformen innerhalb der bestehenden Organisationsstrukturen möglich ist. Ohne ein entsprechendes Veränderungsmanagement (Change Management) ist die Etablierung der technischen Kopplung von Portalen ebenso zum Scheitern verurteilt wie dies bereits für die Einführung eines nicht gekoppelten Unternehmensportals gilt.

7.1.2 Berücksichtigung von Standardisierungen

Neben der grundsätzlichen Notwendigkeit, herstellerunabhängige Standards auf den Ebenen oberhalb der Portlets zu etablieren, ist in der nächsten Evolutionsstufe des vorliegenden Konzepts zur Kopplung von Portalen eine Weiterentwicklung mit dem Ziel der Integration des WSRP-Standards (vgl. [Thompson/Leue/Kropp 2003]) zur Einbindung entfernter Portlets anzustreben. Notwendige, nicht vorhandene Konstrukte sind als Erweiterungen des Standards zu integrieren bzw. sollten direkt in eine zukünftige Version des WSRP-Standards, evtl. als optionale Komponenten, einfließen.

Der Standard ist jedoch in der aktuellen ersten Version 1.0 selbst nur begrenzt produktiv einsetzbar. Wesentliche Aspekte und Funktionalitäten wurden von der Standardisierung vorerst ausgeklammert oder nur in einfacher Form berücksichtigt. Aufgrund der weitgehenden Neuheit des Ansatzes ist dies absehbar gewesen und nachvollziehbar. Er stellt jedoch die Grundlage für folgende Versionen dar, bei denen substanzielle Verbesserungen der Situation zu erwarten sind. Dies würde die Nutzbarkeit, die aktuell vor allem auf Technologiestudien be-

schränkt ist, auf die effektive Lösung faktisch vorhandener Probleme erweitern. Hierzu ist es u. a. notwendig, von Seiten der Anwenderunternehmen Druck auf die Hersteller auszuüben. Diese müssen dazu bewegt werden, die vom Markt benötigten Funktionalitäten zu integrieren.

7.1.3 Erweiterung des Synchronisationsmodells

Das vorgeschlagene Synchronisationsmodell weist ähnlich dem WSRP-Standard Einschränkungen auf. Da es sich ebenfalls um die erste Version einer gegenüber dem WSRP-Standard wesentlich komplexeren, auf eine herstellerübergreifende Nutzung ausgerichteten Gesamtarchitektur handelt, mussten bei der Konzeption Kompromisse eingegangen werden. Dies lässt sich vor allem auf das Fehlen von Standards für Seiten und Plätze zurückführen.

Im Rahmen der Weiterentwicklung des Synchronisationsmodells sollte die Flexibilität in den Bereichen Strukturierung von Seiten und Rechtevergabe verbessert werden. Die bisher angenommene eindimensionale, vollständig lineare Anordnung von Seiten zu einem Seitenregister erweist sich als zu starr, um komplexe Strukturen für die Benutzer leicht nachvollziehbar, abbildbar zu machen. Besser geeignet ist eine zweidimensionale, hierarchische Struktur aus Seiten, so wie sie z. B. im IBM WebSphere-Portal zur Anwendung kommt. In diesem stellt sich die lineare Anordnung als Sonderfall dar. Für Portale, welche die hierarchischen Seitenstrukturen nicht unterstützen, könnte daher ein Kompatibilitätsmodus vorgesehen werden, bei dem die hierarchische Struktur in die bisher verwendete lineare Struktur abgebildet wird. Das Synchronisationsmodell müsste ähnlich der Synchronisation des Layouts einer Seite um die dort dargestellten Sonderfälle erweitert werden. Da die grundsätzliche Konzeptionalisierung und Behandlung bereits verfügbar ist, stellt dies keine Einschränkung dar.

Der zweite wesentliche Punkt betrifft die Rechteverwaltung. Bisher wird vom Coordinator-Place ausschließlich für den gesamten Platz festgelegt, welche anderen Portale diesen Platz verändern dürfen. Diese Granularität ist für den praktischen Einsatz bei größeren Anwendergruppen jedoch zu unflexibel. Anzustreben ist, auf Ebene des Modells verschiedene Elemente mit weitergehenden Rechten versehen zu können. Unterscheiden lassen sich dabei sowohl platzweite Rechte als auch Rechte auf einzelnen Elementen. Zu den platzweiten, übergeordneten Rechten zählt z. B., ob sowohl für Seiten als auch für Portlets – unabhängig voneinander – festgelegt werden kann, ob diese grundsätzlich von einem Portal eingebracht, verändert oder gelöscht werden können. Auf der Ebene einzelner Elemente sollte es möglich sein, den Seiten verschiedene Rechte zuweisen zu können. Dies würde es erlauben, nur ausgewählten Portalen die Möglichkeit zu geben, einzelne Seiten zu editieren. Es kann zwischen dem Verändern der

Position und des Namens, des Layouts und der enthaltenen Portlets sowie dem Löschen der Seiten unterschieden werden. Gleiches ließe sich eine Ebene darunter auch für Zeilen- und Spalten-Container definieren. Bei gezielter Anwendung eines derart feingliederigen Berechtigungskonzepts ließen sich die durch sich widersprechende Änderungen entstehenden Konflikte auf technischer Grundlage minimieren. Auch hier wäre ein Kompatibilitätsmodus für Portale möglich, welche diese Form der Rechtevergabe nicht unterstützen, indem auf das bisher vorgesehene Schema zurückgegangen wird.

7.1.4 Integration der Inter-Portlet-Kommunikation

Eine Technologie, die zumindest in praktisch allen kommerziellen Portalprodukten der aktuellen Generation enthalten ist, ist die Inter-Portlet-Kommunikation. Dabei handelt es sich in der Regel um eine serverbasierte Technologie, die es erlaubt, Portlets miteinander interagieren zu lassen. Die visuelle Integration verschiedener Anwendungen durch die Anordnung von Portlets in Seiten wird so um eine funktionelle Integration erweitert. Die Auswahl einer Bestellung in einem Warenwirtschafts-Portlet kann dann gleichzeitig zur Anzeige des Kundenstamms und der Kundenumsätze in weiteren Portlets führen. Ein weiteres Beispiel ist die Initiierung einer Bestellung in einem Portlet durch die Auswahl eines Produkts im Produktkatalog in einem anderen Portlet.

Diese Funktionalität wird in verschiedenen Ausprägungen realisiert, aber jeweils ausschließlich durch herstellereigenspezifische APIs und die damit verbundene Programmierung. Die von Art und Umfang her sehr unterschiedlichen Implementierungsvarianten haben dazu geführt, dass die Inter-Portlet-Kommunikation weder in der ersten Version des Java Portlet API (vgl. [Java Community Process 2003]) noch der ersten Version des WSRP-Standards (vgl. [Thompson/Leue/Kropp 2003]) Berücksichtigung gefunden hat. Aufgrund der hohen Bedeutung, die der Inter-Portlet-Kommunikation sowohl von den Herstellern, aber vor allem auch von den Anwendern beigemessen wird, ist zu erwarten, dass sich dies mit der nächsten Version der Standards ändert. So würde es dann möglich, auch im Rahmen der Evolution des vorliegenden Konzepts (vgl. Abschnitt 7.1.2), die Inter-Portlet-Kommunikation zu integrieren. Damit ließe sich eine wesentliche funktionale und für Benutzer und den produktiven Einsatz bedeutsame Einschränkung aufheben.

7.1.5 Föderierung weiterer Portaldienste

Im Abschnitt 4.2.5 wurden ausgewählte Dienste vorgestellt, die von praktisch allen Portalen angeboten werden und denen eine hohe Bedeutung zukommt. Diese konnten in der ersten Version des entwickelten Konzepts nicht berücksichtigt werden. Im Folgenden wird ein Überblick darüber gegeben, welche Anforderungen und Möglichkeiten für eine Integration in eine der nächsten Versionen des Konzepts bestehen.

Die *Personalisierung* ist im Sinne des Kontinuums zur Portalkopplung auf die Personalisierung der Inhalte der Portlets beschränkt. Hierzu steht aktuell maximal der Benutzername zur Verfügung. Dies ist für eine auf den Benutzer abgestimmte Auswahl der Inhalte jedoch nicht ausreichend.

Ohne Kopplung von Portalen sind zwei Ansätze anzutreffen. Entweder erfasst das Portlet die notwendigen Profilinformatoren selbst oder es erfragt diese bei einem übergeordneten Portaldienst. Dieser stellt die zentral auf Ebene des Portals erhobenen Profilinformatoren zur Verfügung. Der erste Ansatz ist analog direkt im Fall gekoppelter Portale anwendbar. Der zweite, zu präferierende Ansatz vermeidet eine Mehrfacheingabe und Pflege von Profilinformatoren. Zu seiner Realisierung bei gekoppelten Portalen ist es jedoch notwendig, Profilinformatoren herstellerübergreifend präzise beschreiben zu können. Erst dadurch können die vom Portlet benötigten mit den lokal beim Nachfrager verfügbaren Informationen in Übereinstimmung gebracht werden.

Aktuell kommen jedoch keine solchen herstellerübergreifenden Standards zum Einsatz. Allgemeine portalunabhängige Ansätze und Standardisierungen, z. B. Dublin Core (vgl. [DCMI 2004]), stehen jedoch zur Verfügung und würden sich für eine Erweiterung des Konzepts anbieten. Weitere Fragestellungen ergeben sich bei der Steuerung, welche Profilinformatoren zwischen den beteiligten Portalen aus Sicherheits- und Datenschutzgründen überhaupt ausgetauscht werden sollen und dürfen.

Die im Zusammenhang mit der Sichtbarkeit der internen Organisation benannte Anforderung zur Filterung auf Ebene einzelner Informationseinheiten, z. B. Dokumenten (vgl. Abschnitt 4.4.1.2), lässt sich ebenfalls der Personalisierung zuordnen und kann mithilfe der gleichen Ansätze berücksichtigt werden.

Im Bereich der *Suche* ist besonders der Einsatz der brokerbasierten Suche für eine Kopplung von Portalen geeignet. Bei dieser sind die Suchfunktionen der gekoppelten Portale nur eine weitere externe Quelle, die zur Bildung des Suchergebnisses herangezogen werden. Zur

Anbindung dieser Suchfunktionen ist es zweckmäßig, unabhängig von Portalen, standardisierte Schnittstellen zum externen Zugriff auf die Suche zu definieren. Andernfalls sind entsprechende proprietäre Schnittstellen zu verwenden. Die rein indexbasierte Suche ist zur Kopplung nicht geeignet, da die Partner in den seltensten Fällen bereit sein werden, ihre Informationsquellen vollständig und direkt zu öffnen, so wie dies für eine indexbasierte Suche notwendig ist. Dies widerspricht zudem vollständig dem verfolgten Ansatz der weitgehenden Erhaltung der Autonomie bei der Kopplung. Stattdessen ist eine hybride Lösung anzustreben, die ggf. für lokale Quellen indexbasiert arbeitet, als Erweiterung jedoch die brokerbasierte Einbindung gekoppelter Portale erlaubt. Weitere Fragestellungen ergeben sich im Bereich der Weitergabe und Berücksichtigung der Zugriffsrechte der Benutzer bei der Generierung der Anzeige des Suchergebnisses.

Die Erreichung der Zielsetzung, eine gemeinsame intra- und interorganisationale Arbeitsumgebung für verteilte und virtuelle Teams zu schaffen, bei der besonders die gemeinsame Nutzung von Plätzen im Vordergrund steht, würde durch die Integration von *echtzeitkollaborativen Funktionalitäten* wesentlich unterstützt werden. Dies schließt alle Strukturebenen (Portal selbst, Plätze, Seiten und Portlets) eines Portals ein. Dem entgegen steht vor allem die produktspezifische Integration der Funktionalitäten. Die eingesetzten Techniken basieren grundsätzlich auf derselben technologischen Basis, setzen jedoch eigene Funktionsaufrufe ein. Die Standardisierung der Syntax und Semantik der Funktionsaufrufe, nicht deren Implementierung, würde ausreichen, um die Integration auf Seiten der Portaldarstellung zu erreichen. Daneben ist eine Kopplung der die echtzeitkollaborativen Funktionalitäten bereitstellenden unternehmensinternen Server erforderlich. Diese erlaubt die Interaktion mit den Benutzern der gekoppelten Portale. Entgegen eventueller Erwartungen ist diese Kopplung bereits möglich und standardisiert. Das offene und standardisierte Session Initiation Protocol (SIP) und das darauf aufbauende SIP for Instant Messaging and Presence Leveraging (SIMPLE) (vgl. [IETF 2004]) sind die Basis der Verbindung. Die Verfügbarkeit in Produkten ist zwar noch beschränkt, aktuelle Ankündigungen sprechen aber für eine zunehmende Verbesserung der Situation. Zu berücksichtigen sind jedoch die Auswirkungen, die sich durch eine derartige Kopplung auf die Sichtbarkeit der internen Organisation (vgl. Abschnitt 4.4.1.2) ergeben.

7.1.6 Nutzungs- und Abrechnungsverfahren

Grundlage der Nutzung praktisch aller IT-Systeme in mittleren bis großen Firmen ist die Verrechnung der Kosten im Rahmen von Fixkosten und darüber hinaus in Abhängigkeit der tatsächlichen Inanspruchnahme der Systeme. Dies wird in der entsprechenden Literatur unter den Begriffen Usage & Billing behandelt. Der praktische Einsatz einer Portalkopplung erfordert die Zurechenbarkeit und Verrechnung der entstehenden Kosten. Hierbei sind verschiedene Auslöser und Aktionen zu unterscheiden, die unterschiedlich gewichtet und mit monetären Größen versehen werden müssen. Abhängig davon, ob es sich um einseitige oder wechselseitige Nutzung handelt, sind entsprechende Clearing-Verfahren notwendig.

Ohne detaillierte Betrachtung ist es nicht möglich abzuschätzen, inwieweit eine Erweiterung des Daten- und Funktionsmodells um entsprechende Abrechnungsinformationen notwendig ist. So ist es vorstellbar, diese jeweils lediglich lokal bei ihrer Entstehung zu erfassen und in regelmäßigen Abständen ausschließlich ein Clearing durchzuführen, für das eine Erweiterung notwendig ist. Dies setzt jedoch ein hohes Maß an Vertrauen zwischen den Teilnehmern voraus, da standardisierte und akzeptierte Kontrollmöglichkeiten der Clearing-Grundlage fehlen. Daher wäre es alternativ möglich, jede relevante Funktion um entsprechende Kosteninformationen zu erweitern, die das Controlling und Auditing bei allen Partnern zulassen. Gleichmaßen ist festzulegen, welche Operation auf Basis welcher Messgrößen mit entsprechenden Kosten zu versehen sind. Übliche Messgrößen sind etwa die Anzahl der Nutzungen, die Zeit der Nutzung oder das übermittelte bzw. verarbeitete Datenvolumen.

7.1.7 Weiterentwicklung der Implementierung

Die bisher dargestellten Aspekte bezogen sich ausschließlich auf die Erweiterung oder Anpassung des Konzepts zur Kopplung von Portalen. Die Erweiterung des Konzepts sollte jedoch durch die Erweiterung der vorgestellten Implementierung begleitet werden. Dadurch lässt sich eine Validierung des Konzepts durchführen. Parallel bieten sich aber bereits im Rahmen des vorliegenden Konzepts Erweiterungsmöglichkeiten. Obwohl die vorgestellte Implementierung alle verbindlichen Bestandteile des Konzepts umsetzt, sind allgemein die Robustheit und damit einhergehend die Fehlerbehandlung sowie im Besonderen die Flexibilität des Portlet-Routings verbesserungsfähig.

Die Implementierung des Portlet-Routings verwendet, falls der Portlet-Provider nicht zur Verfügung steht, aktuell immer das erste Portal, das ihm Zugriff auf die Portlet-Definition bietet.

Ist dieses Portal zum Zeitpunkt des tatsächlichen Zugriffs nicht verfügbar, kann der Portlet-Inhalt nicht angezeigt bzw. die Aktion nicht ausgeführt werden. Dies trifft auch dann zu, wenn es einen weiteren Intermediär-Portlet-Provider gibt, über den der Portlet-Consumer auf das Portlet zugreifen könnte. Hier wäre eine Erweiterung derart sinnvoll, dass für alle Portlets alle Intermediär-Portlet-Provider gespeichert werden, die nach bestimmten Regeln beim Abruf herangezogen werden können. Eine der Regeln könnte das Kosten minimierende Routing sein. Hierbei würde immer die funktionsfähige Route ausgewählt, welche die geringsten Kommunikationskosten verursacht. Festzulegen ist, wie die Kosten definiert und gemessen werden. In Frage kommen u. a. unterschiedlich hohe Kosten, die bei der Nutzung unterschiedlicher Intermediär-Portlet-Provider anfallen, oder Kosten, die durch die Anzahl der Stationen bis zum Portlet-Provider entstehen.

7.1.8 Implementierung der Kopplung in einem anderen Portal-Framework

Die informationstechnische Validierung des entwickelten Konzepts erfolgt bisher ausschließlich anhand der Erweiterung des bestehenden G8-Portal-Frameworks. Daraus entstehende Portalkopplungen sind im Sinne der Taxonomie gekoppelter Portale (vgl. Abschnitt 4.3.3) bzgl. der Dimension der Heterogenität als homogene Portalkopplung zu identifizieren. In diesem Aspekt bleibt die Umsetzung hinter dem allgemeineren Konzept zurück. Die Erweiterung eines anderen Portal-Frameworks ist dazu geeignet, diese Beschränkung zu überwinden. Sie bietet gleichzeitig die Möglichkeit, bisher im Rahmen von homogenen Portalkopplungen unentdeckt gebliebene Schwachstellen des Modells zu identifizieren und anschließend zu beseitigen. Dies würde es ggf. auch erleichtern, Fallstudien durchzuführen, da nicht mehr die Beschränkung auf den ausschließlichen Einsatz des G8-Portals bestünde.

Für eine Erweiterung kommen ausschließlich im Quelltext verfügbare Portalimplementierungen in Frage. Durch das damit verbundene Ausscheiden der kommerziellen Produkte schränkt sich die Wahl auf Portal-Frameworks mit Open-Source-Lizenz ein. Aufgrund seiner Beliebtheit und Verbreitung sollte z. B. das Apache Jetspeed Portal-Framework (vgl. [Apache 2004]) in die engere Auswahl einbezogen werden.

7.2 Ergebnis und kritische Würdigung der Arbeit

Die in dieser Arbeit vorgestellten Konzepte, Ansätze und Umsetzungen zur Kopplung von Portalen sind der Startphase des Lebenszyklus der Entwicklung dieses Forschungs- und Technologiezweigs zuzuordnen. Als erste dem Autor bekannte umfassende wissenschaftliche Bearbeitung dieses Themas sind sie das Ergebnis eines dreijährigen Forschungs- und Entwicklungsvorhabens. Als solche sind die Ergebnisse als Beitrag in der Diskussion zur Überwindung von Problemen zu verstehen, die mit der Proliferation von Portalen verbunden sind. Ziel ist die Rückführung von Portalen auf ihr Kernversprechen, das Zurverfügungstellen des Single Point of Access.

Die Ergebnisse sind im Wesentlichen für drei Zielgruppen von Interesse:

- Für die wissenschaftliche Forschung besteht die Aufgabe darin, die in dieser Arbeit behandelten Aspekte zu vertiefen und die Erforschung der sich daraus ergebenden vielfältigen neuen Untersuchungsgegenstände, von denen einige im vorangegangenen Abschnitt 7.1 genannt wurden, fortzusetzen. Dies ist Voraussetzung und Fundament für die Weiterentwicklung des Forschungs- und Technologiezweigs im Sinne des Lebenszyklusgedankens.
- Aus Sicht der Anbieter von Portalen sollten das Konzept und seine Umsetzung bei der Weiterentwicklung der einzelnen Portal-Frameworks und bei Beratungen in einer zu gründenden Standardisierungsinitiative als Orientierung und Maßstab für die Standardisierung einer umfassenden und ganzheitlichen Portalkopplung dienen. Entsprechend der Feststellung in Abschnitt 4.4.2.2 sind nur die Anbieter selbst dazu in der Lage, die für eine Kopplung notwendigen Weiterentwicklungen vorzunehmen. Ohne dass diese sich auf gemeinsame Standards einigen, sind eine weite Verbreitung von Portalkopplungen sowie die Erreichung eines entsprechenden Reifegrads für deren praktischen Einsatz nicht zu erwarten.
- Für die Unternehmen, für welche die Proliferation von Portalen zunehmend zum Problem wird und die sich durch die Kopplung von Portalen dessen Lösung erhoffen, bietet das Konzept ebenfalls Potenziale. Sie können durch die vorliegende Arbeit bereits im Vorfeld einer Standardisierung und der kommerziellen Verfügbarkeit eine präzisere Vorstellung über die zukünftigen Möglichkeiten zur Kopplung von Portalen erhalten. Basierend darauf sind sie in der Lage, ihre Marktmacht zu nutzen und Druck auf die Hersteller von Portal-lösungen auszuüben. Durch die Verfügbarkeit einer praktisch einsetzbaren Implementierung ist es den Herstellern einerseits nicht möglich, die grundsätzliche Realisierbarkeit in Frage zu stellen. Andererseits sind somit für einen möglichen Standard Mindestanforderun-

gen definiert. Auch wenn das Problem der Portalproliferation für die Unternehmen noch nicht vorrangig ist, scheint es dringend geboten, dieses bereits jetzt an die Anbieter heranzutragen. Wie das weit weniger komplexe Beispiel des ausschließlichen Anbietens von Portlets zeigt, können sich Standardisierungsverfahren über Jahre hinziehen, bis ein Standard verabschiedet wird. Bis zur tatsächlichen Verfügbarkeit von standardkonformen Lösungen vergehen anschließend nochmals mehrere Monate.

Der von der vorliegenden Arbeit erhobene Anspruch, sowohl einen theoretischen als auch einen praktischen Beitrag zum Forschungs- und Technologiezweig gekoppelter und föderierter Portale zu leisten, wird im Folgenden durch eine zusammenfassende Darstellung der jeweiligen Beiträge untermauert.

Präzisierung von Begriffen

Der in den verfügbaren (wissenschaftlichen) Veröffentlichungen unreflektiert als leere Wort-hülse gebrauchte Begriff der Föderierung von Portalen (vgl. Abschnitt 2.2.2.3) wird anhand einer aus dem Bereich der föderierten Datenbanken stammenden Taxonomie präzisiert. Diese berücksichtigt die drei Dimensionen Verteilung, Heterogenität und Autonomie und spannt damit einen Raum verschiedener Ausprägungen von allgemein als *Portalkopplungen* zu bezeichnenden Verbindungen auf. Eine Föderation von Portalen ist in diesem Verständnis ausschließlich die Maximalausprägung der Dimension Autonomie (vgl. Abschnitt 4.3.3).

Umfassende Konzeptionalisierung

Orthogonal zur Einordnung einer Portalkopplung in die dargestellte Taxonomie ist die Aufstellung eines Kontinuums der Art einer Portalkopplung (vgl. Abschnitt 4.3.2) zu sehen. Das Kontinuum erlaubt die Entwicklung eines umfassenden Verständnisses darüber, was eine Portalkopplung auszeichnet und was deren Wert darstellt. Zur Wahrung der Praxisrelevanz orientiert es sich dabei an den Gemeinsamkeiten der betrachteten Portal-Frameworks. Das Kontinuum geht über alle bisherigen Ansätze im Bereich von Portalkopplungen weit hinaus. Diese berücksichtigen im besten Fall ausschließlich einen Bestandteil, nämlich das Anbieten von Portlets.

Ganzheitliche Anforderungsanalyse

Die unter Rückgriff auf die vorherigen Ergebnisse stattfindende Analyse von gleichermaßen betriebswirtschaftlichen wie technischen Anforderungen stellt eine umfassende Betrachtung aller relevanten Aspekte sicher (vgl. Abschnitt 4.4). Sie vermeidet eine Überbetonung oder

Auslassung des einen oder anderen Betrachtungsgegenstands. Durch die Ausweitung auf diese ganzheitliche Betrachtung findet eine deutliche Weiterentwicklung bestehender Ansätze im Bereich von Portalkopplungen statt. Diese fokussieren sich zumeist auf die rein technisch-funktionale Ebene, wobei selbst dort eine Beschränkung auf einzelne Aspekte vorherrschend ist.

Geschlossene Modellbildung

Ausgehend von der ganzheitlichen und umfassenden Auseinandersetzung mit Fragen der Kopplung von Portalen wird ein in sich geschlossenes und einheitliches Modell gebildet, das diese mit allen wesentlichen Facetten berücksichtigt (vgl. Abschnitt 4.5 und 4.6). Bisherige punktuell vorhandene Modelle gehen in diesem Gesamtmodell als Spezialfälle auf. Besondere Berücksichtigung beim Entwurf des Modells findet die Sicherstellung eines hohen Maßes an Flexibilität bei der Portalkopplung. Dies sorgt für die potenzielle Realisierbarkeit eines breiten Spektrums von verschiedenartigen Portalkopplungen, die durch unterschiedlichste Formen intra- wie interorganisationaler Zusammenarbeit entstehen können. Durch die konsequente Ausrichtung an den verschiedenen durch die Praxis gestellten Anforderungen geht die Bedeutung und Anwendbarkeit des Modells weit über den rein akademischen, wissenschaftlich-theoretischen Bereich hinaus.

Praktische Implementierung und Verfügbarkeit

Die praktische Implementierung stellt, für ein ausgewähltes Portal-Framework, eine vollständige informationstechnische Realisierung des eingeführten Modells für Portalkopplungen dar (vgl. Kapitel 5). Diese ist nicht auf einzelne ausgewählte Szenarien beschränkt, sondern ermöglicht durch die Berücksichtigung aller bei der Konzeption betrachteten Aspekte eine vollständige praktische Evaluierung. Erst nach einer umfassenden Umsetzung des Modells, die den praktischen Einsatz ermöglicht, kann durch Rückkopplung auf Basis der Erprobung in der Praxis die Etablierung eines Wirkungskreises zwischen Forschung und Praxis erfolgen. Dieser ist unabdingbar für eine optimierte Modellierung und die damit verbundene höhere Praxistauglichkeit. Damit hebt sich die Arbeit wesentlich von vergleichbaren Arbeiten ab, bei denen ein entwickeltes Konzept häufig nur in Ansätzen informationstechnisch umgesetzt und damit validierbar ist.

Kritische Anmerkungen

Die vorliegende Arbeit erreicht die eigene Zielsetzung in umfassender Weise, einige kritische Anmerkungen sind dennoch zwingend erforderlich:

- In dem Anspruch, Vorschläge für Standardisierungen im Bereich gekoppelter Portale zu machen, sind nur eingeschränkte Möglichkeiten gegeben, kommerzielle Portal-Frameworks detailliert auf deren Grundelemente hin zu untersuchen und somit Einblicke in deren Interna zu erlangen. Dies führt zwangsweise dazu, dass die Grundannahmen, auf denen die Vorschläge beruhen (vgl. Abschnitt 4.2), nicht so hinreichend präzise und allgemein gültig sein können, dass das vorliegende Konzept unverändert zur Kopplung von Portalen unterschiedlicher Hersteller herangezogen werden könnte. Zudem macht dies aktuell eine Beschränkung auf ausgewählte Basisfunktionalitäten notwendig, die hinter denen zurückbleiben, die von einem nicht gekoppelten Portal erwartet werden (vgl. auch die Ausführungen im vorhergehenden Ausblick). Wie bereits zuvor erläutert, ist das Konzept daher als Basisarbeit und Diskussionsgrundlage für eine durchzuführende Standardisierung zu interpretieren.
- Die Beschränkung der informationstechnischen Umsetzung und Validierung auf die Erweiterung des G8-Portals führt im Hinblick auf die Heterogenität zur Festlegung auf ausschließlich homogene Portalkopplungen (vgl. die Ausführungen im Abschnitt 7.1.8). Die Ausweitung auf heterogene Portalkopplungen ist jedoch zur umfassenden Validierung des Konzepts notwendig und kann ihrerseits Grenzen und Problemfelder des Modells offenbaren, die bisher nicht erkennbar sind.
- Obwohl die Validierung in der Praxis wie dargestellt aktuell schwierig ist (vgl. Abschnitt 6.1), so ist sie doch eine unabdingbare Grundlage für eine fundierte Bewertung der Praxistauglichkeit und der praktischen Relevanz der Arbeit. Die ersten Erfahrungen sind durchweg positiv, lassen jedoch noch keine ausreichenden Rückschlüsse auf allgemein gültige Aussagen zu. In diesem Bereich sind weitere Anstrengungen notwendig. Daher muss ein abschließendes Urteil offen bleiben.

Schlussbemerkung

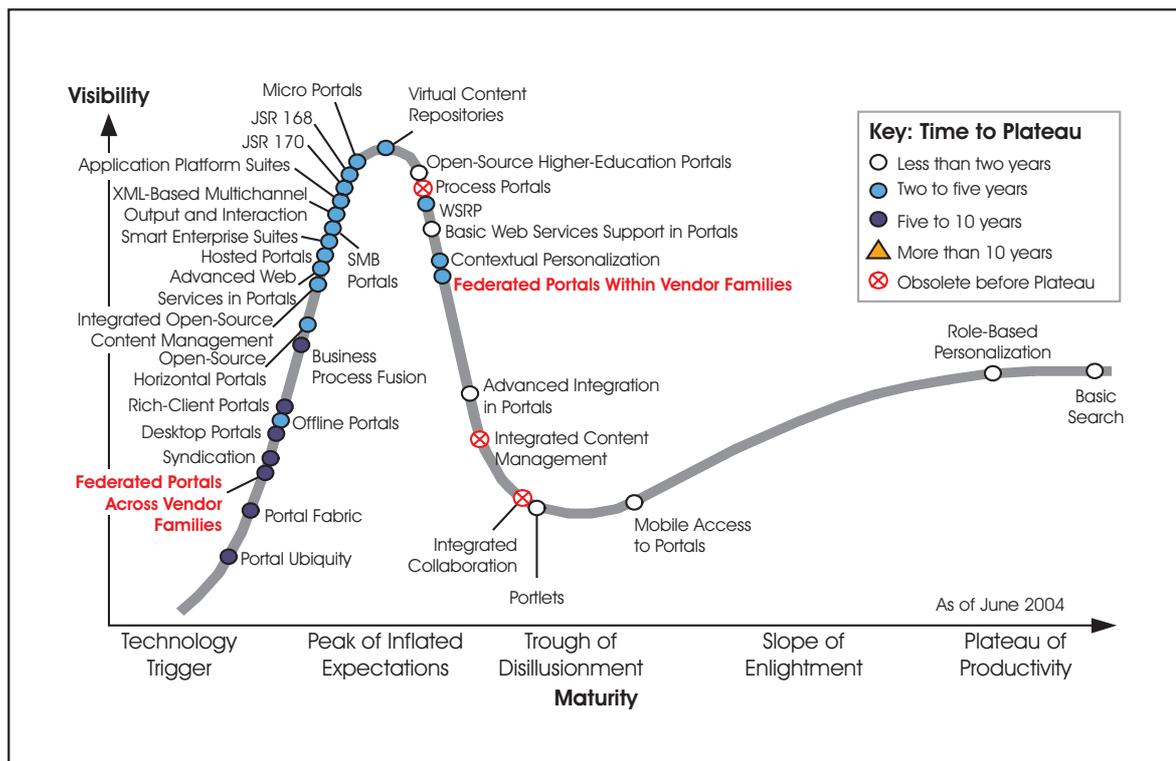


Abbildung 7-1: Hype Cycle for the Portal Ecosystem, 2004
([Phifer et al. 2004])

Abbildung 7-1 zeigt – analog zur Abbildung 2-7 in Abschnitt 2.2.2.3.2 – die Ergebnisse einer Studie von Phifer et al. (vgl. [Phifer et al. 2004]) zu verschiedenen Portalthemen, jedoch aktualisiert für das Jahr 2004. Einerseits fällt auf, dass zahlreiche neue Trends und Technologien zum Portalumfeld hinzugekommen sind. Im Rahmen der vorliegenden Arbeit besonders relevant ist, dass sich andererseits binnen eines Jahres die Positionierung der beiden Themen *Föderierung von Portalen eines Herstellers* und *Föderierung von Portalen verschiedener Hersteller* praktisch nicht verändert hat. Für den Bereich der Föderierung von Portalen eines Herstellers korrigierten die Analysten ihre Prognose für die Verfügbarkeit als Standardtechnologie gar von weniger als zwei Jahren auf nun 2-5 Jahre. Zusammengenommen lässt sich dieses Ergebnis nur derart interpretieren, dass, obwohl ein Interesse an beiden Themen vorhanden ist, im vergangenen Jahr kein substanzieller Fortschritt und Erkenntnisgewinn stattgefunden hat. Der Autor der vorliegenden Arbeit verbindet mit ihr die Hoffnung, einen Beitrag dazu zu leisten, diesen Zustand zu überwinden sowie die Diskussion und Entwicklung auf eine neue Stufe voranzubringen.

8 Literaturverzeichnis

[Alvesson 2000]

Alvesson, M.: Social Identity And The Problem of Loyalty In Knowledge-Intensive Companies, in: Journal of Management Studies; December 2000, Vol. 37, Issue 8, 2000, pp. 1101-1125.

[Amberg/Holzner/Remus 2003]

Amberg, M.; Holzner, J.; Remus, U.: Portal-Engineering – Anforderungen an die Entwicklung komplexer Unternehmensportale, in: 6. Internationale Tagung Wirtschaftsinformatik 2003, Dresden, 2003.

[ANSA 1991]

ANSA: A Systems Designer's Introduction to the Architecture, <http://www.ansa.co.uk/ANSATech/91/RC25300.pdf>, letzter Zugriff: 10.02.2004, 1991.

[Apache 2004]

Apache: Apache Portals – Jetspeed 1, <http://portals.apache.org/jetspeed-1/>, letzter Zugriff: 12.06.2004, 2004.

[Bajaj et al. 2003]

Bajaj, S.; Della-Libera, G.; Dixon, B.; Dusche, M.; Hondo, M.; Hur, M.; Kaler, C.; Lockhart, H.; Maruyama, H.; Nadalin, A.; Nagaratnam, N.; Nash, A.; Prafullchandra, H.; Shewchuk, J.: Web Services Federation Language (WS-Federation), <ftp://www6.software.ibm.com/software/developer/library/ws-fed.pdf>, letzter Zugriff: 04.03.2004, 2003.

[Basha/Irani 2002]

Basha, S. J.; Irani, R.: AXIS: The Next Generation of Java SOAP, Wrox Press Ltd, Birmingham, 2002.

[Bauer 2001]

Bauer, H.: Unternehmensportale: Geschäftsmodelle, Design, Technologien, Galileo Press, Bonn, 2001.

[Beasley et al. 1994]

Beasley, M.; Cameron, J.; Girling, G.; Hoffner, Y.; van der Linden, R.; Thomas, G.: Establishing Co-operation in Federated Systems, <http://www.ansa.co.uk/ANSATech/94/Primary/112802.pdf>, letzter Zugriff: 20.01.2004, 1994.

[Bell/LaPadula 1975]

Bell, D.; LaPadula, L.: Secure Computer Systems: Unified Exposition and Multics Interpretation, in: MTR-2997, (AY/W 020 445) July, The MITRE Corporation, Bedford, MA, 1975.

[Bobak 1996]

Bobak, A. R.: Distributed and Multi-Database Systems, Artech House, Boston, London, 1996.

[Borghoff/Schlichter 1998]

Borghoff, U. M.; Schlichter, J. H.: Rechnergestützte Gruppenarbeit: eine Einführung in verteilte Anwendungen – 2. Auflage, Springer, Berlin u.a., 1998, p. 557.

[Brophy/Venters 2001]

Brophy, R.; Venters, W.: Work, Workspace, and the Workspace Portal, in: Lecture Notes in Artificial Intelligence Volume 2117, 2001, pp. 421-431.

[Brousseau 1994]

Brousseau, E.: EDI and inter-firm relationships: toward a standardization of coordination processes?, in: Information Economics and Policy, Vol. 6, 1994, pp. 319-347.

[Buhl/Christ/Pape 2001]

Buhl, L.; Christ, J.; Pape, U.: Marktstudie Softwaresysteme für Enterprise-Application-Integration, Fraunhofer-Anwendungszentrum für Logistikorientierte Betriebswirtschaft, Paderborn, 2001.

[Bullinger et al. 2002]

Bullinger, H.-J. (Hrsg.); Eberhardt, C.-T.; Gurzki, T.; Hinderer, H.: Marktübersicht Portal Software für Business-, Enterprise-Portale und E-Collaboration, Fraunhofer IRB Verlag, 2002.

[Bullinger/Warnecke/Westkämper 2002]

Bullinger, H.-J.; Warnecke, H. J.; Westkämper, E. (eds.): Neue Organisationsformen im Unternehmen – Ein Handbuch für das moderne Management, Springer, Berlin, Heidelberg, New York, 2002.

[Buretta 1997]

Buretta, M.: Data Replication: Tools and Techniques for Managing Distributed Information, Wiley & Sons Inc., New York u.a., 1997.

[Burnett/Brookes-Roones/Keogh 2002]

Burnett, S.; Brookes-Rooney, A.; Keogh, W.: Brokering Knowledge in Organizational Networks: The SPN Approach, in: Knowledge and Process Management, Vol. 9 (2002), No. 1, 2002, pp. 1-11.

[Burris 1993]

Burris, B.: Technocracy at Work, State University of New York Press, Albany, New York, 1993.

[Busbach 1996]

Busbach, U.: Activity Coordination in Decentralized Working Environments, in: Dix, A.; Beale, R. (eds.): Remote Cooperation: CSCW Issues for Mobile and Teleworkers, Springer Verlag, London, 1996, pp. 95-112.

[Busse et al. 1999]

Busse, S.; Kutsche, R.-D.; Leser, U.; Weber, H.: Federated Information Systems: Concepts, Terminology and Architectures, in: Forschungsberichte des Fachbereichs Informatik – Bericht Nr. 99-9 – April, 1999.

[Buxmann et al. 2000]

Buxmann, P.; Weitzel, T.; Westarp, F. v.; König, W.: The Standardization Problem – An Economic Analysis of Standards in Information Systems, <http://www-i4.informatik.rwth-aachen.de/~jakobs/siit99/proceedings/Weitzel.DOC>, letzter Zugriff: 02.02.2004, 2000.

[Camarinha-Matos/Menzel/Cardoso 2003]

Camarinha-Matos, L. M.; Menzel, K.; Cardoso, T.: ICT support infrastructures and interoperability for VOs, in: Virtual Organisations Cluster – VOSTER WP4 D4.4, <http://www.uninova.pt/~cam/ev/VosterD44.pdf>, letzter Zugriff: 30.01.2004, 2003.

[Cauldwell et al. 2001]

Cauldwell, P.; Chawla, R.; Chopra, V.; Damschen, G.; Dix, C.; Hong, T.; Norton, F.; Ogbuji, U.; Olander, G.; Richmann, M. A.; Saunders, K.; Zaev, Z.: Professional XML Web Services, Wrox Press Ltd, Birmingham, 2001.

[Coase 1937]

Coase, R. H.: The Nature of the Firm, in: *Economica* 4, 1937, pp. 386-405.

[Collins 2001]

Collins, H.: Corporate Portals: Revolutionizing Information Access to Increase Productivity and Drive the Bottom Line, Amacon, New York, 2001.

[Confoy/Trabert 2003]

Confoy, M.; Trabert, O.: BP117 Getting on the Portal Fast Track: Solutions for Lotus Domino Users – Lotusphere 2003 conference presentation, IBM Corporation, Orlando, FL, USA, 2003.

[Conner/Prahalad 1996]

Conner, K. R.; Prahalad, C. K.: A Resource-based Theory of the Firm: Knowledge Versus Opportunism, in: *Organization Science*, Vol. 7, No. 5, September-October, 1996, pp. 477-501.

[Conrad 1997]

Conrad, S.: *Föderierte Datenbanksysteme : Konzepte der Datenintegration*, Springer, Berlin u. a., 1997.

[Conrad et al. 1999]

Conrad, S.; Hasselbring, W.; Hohenstein, U.; Kutsche, R.-D.; Roantree, M.; Saake, G.; Saltor, F.: Engineering Federated Information Systems – Report of EFIS '99 Workshop, in: *ACM SIGMOD Record*, 28, 1999.

[Conrad et al. 2002]

Conrad, S.; Hasselbring, W.; James, A.; Kambur, D.; Kutsche, R.-D.; Thiran, P.: Report on the EFIS 2001 Workshop, in: The Computer Journal, Vol. 45, No. 2, 2002.

[di Vimercati/Samarati 1996]

di Vimercati, S. D. C.; Samarati, P.: Access control in federated systems, in: New Security Paradigms Workshop – Proceedings of the 1996 workshop on New security paradigms, ACM Press, New York, 1996, pp. 87-99.

[Dadam 1996]

Dadam, P.: Verteilte Datenbanken und Client/Server-Systeme: Grundlagen, Konzepte und Realisierungsformen, Springer, Berlin u.a., 1996.

[Dangelmaier et al. 2002]

Dangelmaier, W.; Lessing, H.; Pape, U.; Rütter, M.: Klassifikation von EAI-Systemen, in: HMD, Heft 225, Juni 2002, dPunkt.verlag GmbH, Heidelberg, 2002, pp. 61-71.

[Date 1990]

Date, C. J.: An Introduction to Database Systems, Addison Wesley, Mass. u.a., 2000.

[Davie/Botha 2001]

Davie, R.; Botha, R.: The effect of heterogeneity and autonomy on federated database security, http://www.petech.ac.za/rbotha/Proj4/resources/R_Davie-AcademicPaper.pdf, letzter Zugriff: 29.01.2004, 2001.

[Davydov 2000]

Davydov, M. M.: EIP: The Second Wave, in: Intelligent Enterprise Magazine – Information Supply Chain – March 1, 2000 Vol. 3, No. 4, http://www.intelligententerprise.com/000301/supplychain.jhtml?_requestid=78391, letzter Zugriff: 29.01.2004, 2000.

[Davydov 2001]

Davydov, M. M.: Corporate portals and e-business integration, McGraw-Hill, New York u. a., 2001.

[DCMI 2004]

The Dublin Core Metadata Initiative (DCMI): Homepage der Dublin Core Metadata Initiative – Making it easier to find information, <http://dublincore.org/>, letzter Zugriff: 12.06.2004, 2004.

[De Laat 1999]

De Laat, P.: Research and Development Alliances: Ensuring Trust by Mutual Commitments, in: Ebers, M. (ed.): The formation of inter organizational networks, Oxford University Press, New York, 1999, pp. 146-173.

[Della-Libera et al. 2003]

Della-Libera, G.; Dixon, B.; Dusche, M.; Furguson, D.; Garg, P.; Hahn, T.; Hinton, H.; Hondo, M.; Kaler, C.; Leymann, F.; Lovering, B.; Maruyama, H.; Nadalin, A.; Nagaratnam, N.; Raghavan, V.; Shewchuk, J.: Federation of Identities in a Web Services World – A joint whitepaper from IBM Corporation and Microsoft Corporation, <ftp://www6.software.ibm.com/software/developer/library/ws-fedworld.pdf>, letzter Zugriff: 12.02.2004, 2003.

[Delphi Group 2000]

Delphi Group: Portal Design Primer: 68 Questions for Portal Planners, Delphi Group, <http://www.delphigroup.com/research/whitepapers.htm>, letzter Zugriff: 19.01.2004, 2000.

[Delphi Group 2003]

Delphi Group: The Value of Standards, Delphi Group, <http://www.delphigroup.com/research/whitepapers.htm>, letzter Zugriff: 19.01.2004, 2003.

[Denning 1982]

Denning, D. E.: Cryptography and Data Security, Addison-Wesley, Reading Massachusetts, 1982.

[Detlor 2000]

Detlor, B.: The corporate portal as information infrastructure: towards a framework for portal design, in: International Journal of Information Management, Vol. 20, Issue 2, April, 2000, pp. 91-101.

[Dias 2001]

Dias, C.: Corporate portals: a literature review of a new concept in Information management, in: International Journal of Information Management, Vol. 21, No. 4, Elsevier, 2001, pp. 269-287.

[Ditillo 2000]

Ditillo, A.: The relevance of information flows in knowledge-intensive firms, <http://www.sdabocconi.it/dr/file/wp74.pdf>, letzter Zugriff: 16.02.2004, 2000.

[Dittrich/Jonscher 1994]

Dittrich, K. R.; Jonscher, D.: Current Trends in Database Technology and Their Impact on Security Concepts, in: Database Security, VIII: Status and Prospects, Proceedings IFIP WG11.3 Working Conference, Bad Salzdetfurth, Germany, August, 1994.

[Drakos 2003]

Drakos, N.: “Introducing Microportals: One Portal, Multiple Personalities” — Microportals are an effective technology to generate on-demand portals, 2003.

[Ebers 1994]

Ebers, M.: Die Gestaltung interorganisationaler Informationssysteme – Möglichkeiten und Grenzen der transaktionskostentheoretischen Erklärung, in: Sydow, J.; Windeler, A. (Hrsg.): Management interorganisationaler Beziehungen – Vertrauen, Kontrolle und Informationstechnik, 1994, pp. 22-48.

[Eckel 1999]

Eckel, R.; Business Forum: Architecture for Federated Portals, in: Business Forum, <http://www.bizforum.org/whitepapers/infoimage-001.htm>, letzter Zugriff: 20.01.2004, 1999.

[Edmunds/Morris 2000]

Edmunds, A.; Morris, A.: The problem of information overload in business organisations: a review of the literature, in: International Journal of Information Management 20, Elsevier, 2000, pp. 17-28.

[Essmayr et al. 1995]

Essmayr, W.; Kastner, F.; Preishuber, S.; Pernul, G.; Tjoa, A. M.: Access Controls for Federated Database Environments – Taxonomy of Design Choices, in: In Proceedings of the Joint IFIP TC 6 and TC 11 Working Conference on Communications and Multimedia Security. Graz, Austria, Chapman and Hall, 1995.

[Faisst 1998]

Faisst, W.: Unterstützung virtueller Unternehmen durch Informations- und Kommunikationssystem – eine lebenszyklusorientierte Analyse, Inaugural-Dissertation Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen usw., 1998.

[Feather 1998]

Feather, J.: The information society: A study of continuity and change, Library Association, London, 1998.

[Firestone 2003]

Firestone, J. M.: Enterprise Information Portals and Knowledge Management, Butterworth-Heinemann, Amsterdam [u.a.], 2003.

[Fischer/Rensin/Rödig 2000]

Fischer, S.; Rensing, C.; Rödig, U.: Open internet security, Springer, Berlin u.a., 2000.

[Flehmig 2001]

Flehmig, M.: Integration und Personalisierung – Zur Realisierung essentieller Aspekte von Portal-Systemen, in: Proceedings international workshop „E-Commerce: Systemunterstützung für den Umgang mit Daten und Prozessen in vernetzten Anwendungsumgebungen“, Wien, September, 2001.

[Föcker/Lienemann 2000]

Föcker, E.; Lienemann, C.: Informationslogistische Dienste für Unternehmensportale, in: Wissensmanagement 3/00, 2000, pp. 18-22.

[Foss 1996a]

Foss, N. J.: Knowledge-based Approaches to the Theory of the Firm: Some Critical Comments, in: *Organization Science*, Vol. 7, No. 5, September-October, 1996, pp. 470-476.

[Foss 1996b]

Foss, N. J.: More Critical Comments on Knowledge-based Theories of the Firm, in: *Organization Science*, Vol. 7, No. 5, September-October, 1996, pp. 519-523.

[Franz 1999]

Franz, H.: The Impact of Computer Mediated Communication on Information Overload in Distributed Teams, in: *Proceedings of the 32nd Hawaii International Conference on System Sciences*, 1999.

[Frye 2002]

Frye, C.: IT still spending on portals, in: *ADTMag.com*, <http://www.adtmag.com/article.asp?id=6767>, letzter Zugriff: 20.01.2004, 2002.

[Gabler 1997]

Gabler: *Gabler Wirtschafts-Lexikon*, 14. Auflage, Gabler, Wiesbaden, 1997.

[Gaedke/Turowski 1999]

Gaedke, M.; Turowski, K.: Generic Web-Based federation of business application systems for e-commerce applications, in: Conrad, S.; Hasselbring, W.; Saake, G. (eds.): *Engineering Federated Information Systems (Proc. EFIS'99)*, Infix-Verlag, Sankt Augustin, 1999, pp. 25-42.

[Ghoshal/Moran 1996]

Ghoshal, S.; Moran, P.: Bad for practice: A critique of the transaction cost theory, in: *Academy of Management Review*; Jan96, Vol. 21, Issue 1, 1996, pp. 13-47.

[Gootzit/Phifer 2003]

Gootzit, D.; Phifer, G.: Gen-4 Portal Functionality: From Unification to Federation, in: *Gartner Report*, 2003.

[Graham et al. 2002]

Graham, S.; Simeonov, S.; Boubez, T.; Davis, D.; Daniels, G.; Nakamura, Y.; Neyama, R.: *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, Sams Publishing, Indianapolis, 2002.

[Grant 1996]

Grant, R. M.: Prospering in Dynamically-competitive Environments: Organizational Capability as Knowledge Integration, in: *Organization Science*, Vol. 7, No. 4, July-August, 1996, pp. 375-387.

[Grant 1997]

Grant, R. M.: The Knowledge-based View of the Firm: Implications for Management Practice, in: *Long Range Planning*, Vol. 30, No. 3, 1997, pp. 450-454.

[Grote 1990]

Grote, B.: Ausnutzung von Synergiepotentialen durch verschiedene Koordinationsformen ökonomischer Aktivitäten: zur Eignung der Transaktionskosten als Entscheidungskriterium, Lang Verlag, Frankfurt am Main [u. a.], 1990.

[Gurzki 2002]

Gurzki, T.: Vom Prozess zum Portal – Die Konzeption von wirtschaftlichen Portalen, http://www.gurzki.de/publications/prozessportale02/Gurzki_Konzeption_Wirtschaftliche_Portal.pdf, letzter Zugriff: 19.01.2003, 2002.

[Hahnl 2000]

Hahnl, O.: Personalisierte Portaltechnologien auf Basis einer prozessgetriebenen Groupware-Umgebung, Masters Thesis, University of Paderborn, Department of Business Computing 2, 2000.

[Handy 1995]

Handy, C.: Trust and the Virtual Organization, in: Harvard Business Review, May-June, 1995, pp. 40-50.

[Hasselbring 2000]

Hasselbring, W.: The Role of Standards for Interoperating Information Systems, in: Information Technology Standards and Standardization: A Global Perspective, K. Jakobs (ed.), Idea Group Publishing, 2000, pp. 116-130.

[Hasselbring et al. 2000]

Hasselbring, W.; van den Heuvel, W.-J.; Houben, G.; Kutsche, R.-D.; Rieger, B.; Roantree, M.; Subieta, K.: Research and Practice in Federated Information Systems – Report of the EFIS '2000 International Workshop, in: ACM SIGMOD Record, 29, 2000.

[Hazra 2002]

Hazra, T. K.: Building enterprise portals: principles to practice, in: International Conference on Software Engineering – Proceedings of the 24th international conference on Software engineering table of contents Orlando, Florida, ACM Press, New York, NY, USA, 2002, pp. 623-633.

[Heimbigner/McLeod 1981]

Heimbigner, D.; McLeod, D.: Federated Information Bases – A Preliminary Report, in: Infotech State of the Art Report: Database, Infotech State of the Art Reports, Vol. 9, Pergamon Infotech Limited, Maidenhead, UK, 1981, pp. 383-410.

[Heimbigner/McLeod 1985]

Heimbigner, D.; McLeod, D.: A federated architecture for information management, in: ACM Transactions on Information Systems (TOIS) Vol. 3, Issue 3 (July), ACM Press, New York, 1985, pp. 253-278.

[Heinrich 1993]

Heinrich, L. J.: Wirtschaftsinformatik – Einführung und Grundlegung, Oldenbourg Verlag GmbH, München, 1993.

[Heydebrand 1989]

Heydebrand, W.: New Organizational Forms, in: Work and occupations, 16, 1989, pp. 323-357.

[Hildreth 2002]

Hildreth, S.: Plug-and-Play Portal Standards on the Horizon, in: ebizq: The insider's guide to business Integration, <http://www.ebizq.net/topics/eai/features/1760.html>, letzter Zugriff: 16.01.2004, 2002.

[Hinderer 2002]

Hinderer, H.: Portale in der E-Business Strategie: Konzepte für den erfolgreichen Einsatz, http://portal.wiso.uni-erlangen.de/wps/portal/.cmd/ChangeExpandState/.exp/549/.miid/null/.exps/false/.def/true/.scr/Home/_s.155/464, letzter Zugriff: 19.01.2004, 2002.

[IBM 2003]

IBM: Guide to WebSphere Portal 5.0, <http://www-900.ibm.com/developerworks/cn/wsdd/download/pdf/portal5whitepaper.pdf>, letzter Zugriff: 14.05.2004, 2003.

[IBM 2004]

IBM: Business Portals: WebSphere Portal, <http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/portal>, letzter Zugriff: 12.06.2004, 2004.

[IETF 2004]

The Internet Engineering Task Force (IETF): SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), <http://www.ietf.org/html.charters/simple-charter.html>, letzter Zugriff: 28.06.2004, 2004.

[Jajodia/Wijesekera 2001]

Jajodia, S.; Wijesekera, D.: Security in Federated Database Systems, in: Information Security Technical Report Vol. 6, Issue 2 , 1 June, 2001, pp. 69-79.

[Java Community Process 2003]

Java Community Process: JSR-168 Portlet API 1.0 FR, <http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>, letzter Zugriff: 20.01.2004, 2003.

[Jin Kim/Chaudhury/Rao 2002]

Jin Kim, Y.; Chaudhury, A.; Rao, H. R.: A Knowledge Management Perspective to Evaluation of Enterprise Information Portals, in: Knowledge and Process Management Vol. 9, Issue 2, 2002, pp. 57-71.

[Jonscher/Dittrich 1993]

Jonscher, D.; Dittrich, K. R.: Access Control for Database Federations a discussion of the state-of-the-art, in: DBTA Workshop on Interoperability of Database Systems and Database Applications, Fribourg, Switzerland, October, 1993.

[Jonscher/Dittrich 1994]

Jonscher, D.; Dittrich, K. R.: An Approach For Building Secure Database Federations, in: Proceedings of 20th International Conference on Very Large Data Bases, 1994, pp. 24-35.

[Kaspar/Burghardt/Schumann 2002]

Kaspar, C.; Burghardt, M.; Schumann, M.: Konzept einer problemgerechten Informationsbedarfsdeckung im Rahmen wissensbezogener Portalstrategien, in: Workshop „Wissensmanagement mit Unternehmensportalen“, Fraunhofer IRB Verlag, 2002, pp. 177-180.

[Kawell et al. 1992]

Kawell, L.; Beckhardt, S.; Halvorsen, T.; Ozzie, R.; Greif, I.: Replicated Document Management in a group communication system, in: Marca, D.; Bock, G. (eds.): Groupware: Software for computer supported cooperative work, IEEE Computer Society Press, 1992, pp. 226-235.

[Keen 1990]

Keen, P. G. W.: Telecommunications and Organizational Choice, in: Fulk, J.; Steinfield, C. (eds.): Organizations and Communications Technology, Sage Publications, Newbury Park, CA, 1990, pp. 295-312.

[Kemper/Eickler 1999]

Kemper, A.; Eickler, A.: Datenbanksysteme – Eine Einführung, R. Oldenbourg Verlag, München, Wien, 1999.

[Klüber/Alt/Österle 2000]

Klüber, R.; Alt, R.; Österle, H.: Implementing Virtual Organizing in Business Networks: A Method of Inter-Business Networking, in: Malhotra Y. (ed.): Knowledge Management and Virtual Organizations: Theories, Practices, Technologies and Methods, Idea Group Publishing, Hershey u.a., 2000, pp. 43-62.

[Köszegi 2001]

Köszegi, S.: Vertrauen in virtuellen Unternehmen, Deutscher Universitätsverlag, Wiesbaden, 2001.

[Kogut/Zander 1996]

Kogut, B.; Zander, U.: What Firms Do? Coordination, Identity, and Learning, in: Organization Science, Vol. 7, No. 5, September-October, 1996, pp. 502-518.

[Kreger 2001]

Kreger, H.: Web Services Conceptual Architecture (WSCA 1.0), <http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>, letzter Zugriff: 25.02.2004, 2001.

[Kyte 2003]

Kyte, A.; Gartner Group: Personalized Portals Power Successful e-Procurement – Strategic Planning, 2000.

[Landau et al. 2003]

Landau, S.; Venezuela, C. C.; Ellison, G.; Hodges, J.; Kellomäki, S.; Kemp, J.; Linn, J.; Thompson, P.: Liberty ID-WSF Security and Privacy Overview, <http://www.projectliberty.org/specs/liberty-idwsf-security-privacy-overview-v1.0.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Langenscheidt 2003]

Langenscheidt: Langenscheidt Fremdwörterbuch Online Edition, Langenscheidt, Berlin und München, <http://www.langenscheidt.de/fremdwb/index.html>, letzter Zugriff: 29.07.2004, 2003.

[Letson 2001]

Letson, R.: A Closer Look at Portals and EAI, http://www.imagingmagazine.com/db_area/archs/2001/07/tfm0107f1.shtml, letzter Zugriff: 10.03.2004, 2001.

[Lewis 1996]

Lewis, D.: Dying for Information?, Reuters Business Information, London, 1996.

[Liberty Alliance Project 2001]

Liberty Alliance Project: Industry Leaders to form Network Identity Alliance – Liberty Alliance Project, in: Liberty Alliance Project Press Releases, <http://www.projectliberty.org/press/releases/2001-09-26.html>, letzter Zugriff: 05.03.2004, 2001.

[Liberty Alliance Project 2003a]

Liberty Alliance Project: Business Benefits of Federated Identity, <http://www.projectliberty.org/resources/whitepapers/LAP%20Business%20Benefits%20White%20Paper%20v4.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Liberty Alliance Project 2003b]

Liberty Alliance Project: Identity Systems and Liberty Specification Version 1.1 Interoperability, <http://www.projectliberty.org/resources/whitepapers/Liberty%20and%203rd%20Party%20Identity%20Systems%20White%20Paper.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Liberty Alliance Project 2003c]

Liberty Alliance Project: Introduction to the Liberty Alliance identity architecture, <http://www.projectliberty.org/resources/whitepapers/LAP%20Identity%20Architecture%20Whitepaper%20Final.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Linn et al. 2003]

Linn, J.; Boeyen, S.; Ellison, G.; Karhuluoma, N.; MacGregor, W.; Madsen, P.; Sengodan, S.; Shinkar, S.; Thompson, P.: Liberty Trust Models Guidelines, <http://www.projectliberty.org/specs/liberty-trust-models-guidelines-v1.0.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Linthicum 1999]

Linthicum, D. S.: Enterprise Application Integration, Addison-Wesley, Boston u. a., 1999.

[Linthicum 2001]

Linthicum, D. S.: B2B Application Integration – e-Business-Enable Your Enterprise, Addison-Wesley, Boston u.a., 2001.

[Linthicum 2004]

Linthicum, D. S.: Next Generation Application Integration – From Simple Information to Web Services, Addison-Wesley, Boston u.a., 2004.

[Loose/Sydow 1994]

Loose, A.; Sydow, J.: Vertrauen und Ökonomie in Netzwerkbeziehungen – Strukturations-theoretische Betrachtungen, in: Sydow, J.; Windeler, A. (Hrsg.): Management interorganisationaler Beziehungen – Vertrauen, Kontrolle und Informationstechnik, Westdeutscher Verlag, Opladen, 1994, pp. 160-193.

[Lotus 2000]

Lotus: Inside Notes – The Architecture of Notes and the Domino server, 2000.

[Mann 2002]

Mann, J.: Portals beget portals, in: ZDNet TechUpdate, <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2881090,00.html>, letzter Zugriff: 16.01.2004, 2002.

[McInturff 2003]

McInturff, T. J.: Creating the Optimal Portal, <http://www.loma.org/res-10-03-portal.asp>, letzter Zugriff: 11.03.2004, 2003.

[Mello 2002]

Mello, A.: Pull the plug on proliferating portals, in: ZDNet TechUpdate, <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2881466,00.html>, letzter Zugriff: 11.03.2004, 2002.

[Michaelis 1985]

Michaelis, E.: Organisation unternehmerischer Aufgaben – Transaktionskosten als Beurteilungskriterium, Lang Verlag, Frankfurt am Main [u. a.], 1985.

[Murray 1999]

Murray, G.: The Portal Is the Desktop, in: e-ProMag, <http://www.e-promag.com/eparchive/index.cfm?fuseaction=viewarticle&ContentID=166&websiteid=>, letzter Zugriff: 09.02.2004, 1999.

[Narth 2003]

Narth, P.: Helping Portals Work And Play Well Together, in: eBizQ, <http://www.ebizq.net/topics/portals/features/2501.html?pp=1>, letzter Zugriff: 20.01.2004, 2003.

[Nemiro 2000]

Nemiro, J. E.: The Glue That Binds Creative Virtual Teams, in: Malhotra, Y. (ed.): Knowledge Management and Virtual Organizations, Idea Publishing Group, Hershey, London, 2000, pp. 101-123.

[Neuburger 1994]

Neuburger, R.: Auswirkung von EDI auf die zwischenbetriebliche Arbeitsteilung – Eine transaktionskostentheoretische Analyse, in: Sydow, J.; Windeler, A. (Hrsg.): Management interorganisationaler Beziehungen – Vertrauen, Kontrolle und Informationstechnik, 1994, pp. 48-70.

[Nohlen/Schultze 1995]

Nohlen, D.; Schultze, R.-O. (eds.): Lexikon der Politik – Band I Politische Theorien, Beck, München, 1995.

[Nooteboom 1999]

Nooteboom, B.: Inter-firm alliances: analysis and design, Routledge, London u.a., 1999.

[Nurmi 1998]

Nurmi, R.: Knowledge-intensive firms, in: Business Horizons; May/June 1998, Vol. 41 Issue 3, 1998, pp. 26-33.

[OASIS 2002]

Organization for the Advancement of Structured Information Standards (OASIS); OASIS: UDDI Data Structure Reference V1.0 – UDDI Published Specification, 28 Juni 2002, <http://uddi.org/pubs/DataStructure-V1.00-Published-20020628.pdf>, letzter Zugriff: 25.02.2004, 2002.

[OASIS 2003]

Organization for the Advancement of Structured Information Standards (OASIS); OASIS: Extensible Access Control Markup Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, letzter Zugriff: 25.05.2004, 2003.

[Özsü/Valduriez 1991]

Özsü, T.; Valduriez, P.: Distributed Database Systems: Where Are We Now?, in: IEEE Computer, 24(8), 1991, pp. 68-78.

[Oracle 2002]

Oracle: Feature Overview – ORACLE 9i AS Portal, http://otn.oracle.com/products/portal/htdocs/30overview/portal_fov.html, letzter Zugriff: 14.05.2004, 2002.

[Oracle 2004]

Oracle: Enterprise Portals – Access and Organize Information and Applications with an Enterprise Portal, http://www.oracle.com/solutions/enterprise_portals/index.html, letzter Zugriff: 12.06.2004, 2004.

[Palmer 2003]

Palmer, N.: Portals and Collaboration, in: AIIM E-DOC Magazine, http://www.edocmagazine.com/vault_articles.asp?ID=25764, letzter Zugriff: 12.03.2004, 2003.

[Pfeiffer 1990]

Pfeiffer, H. K. C.: The Diffusion of Electronic Data Interchange, Institut für Wirtschaftsinformatik der Universität Bern, Bern, 1990.

[Phifer 2003]

Phifer, G.: The 'Uberportal': An Answer to the Multiple Portal Problem, in: Gartner Report, 2003.

[Phifer 2004]

Phifer, G.; Gartner Group: IHC Uses an SES to Bring Order to a Chaotic Intranet – Case Studies, 2004.

[Phifer et al. 2003]

Phifer, G.; Plummer, D. C.; Gootzit, D.; Hayward, S.; Drakos; Nikos; Valdes, R.; Gilbert, M. R.; Caldwell, F.; Latham, L.; Knox, R. E.; Natis, Y. V.; Pezzini, M.; Shegda, K. M.; Logan, D.; Gartner Group: Hype Cycle for Portal Ecosystems, 2003 – Strategic Analysis Report, 2003.

[Phifer et al. 2004]

Phifer, G.; Valdes, R.; Gootzit, D.; Drakos; Nikos; Shegda, K. M.; Caldwell, F.; Filho, W. A. D. A.; Hayward, S.; Knox, R. E.; Latham, L.; Lundy, J.; Natis, Y. V.; Pezzini, M.; Plummer, D. C.; Logan, D.; Andrews, W.; Gartner Group: Hype Cycle for the Portal Ecosystem 2004 – Strategic Analysis Report, 2004.

[Picot/Reichwald/Wigand 2001]

Picot, A.; Reichwald, R.; Wigand, R. T.: Die grenzenlose Unternehmung – Information, Organisation und Management, Gabler Verlag, Wiesbaden, 2001.

[PiroNet NDH 2002]

PiroNet NDH: Persönliche Portale bestimmen den Profit, <http://www.pironetndh.com/servlet/PB/menu/1002021>, letzter Zugriff: 19.01.2004, 2002.

[Ploch 2003]

Ploch, H.; G8 Project: Föderierte Portale als Integrationsplattform wissensintensiver Unternehmenskooperationen, Masters Thesis, University of Paderborn, Department of Business Computing 2, 2003.

[Plumtree 1999]

Plumtree: Corporate portals – A Simple View of a Complex World, http://www.plumtree.com/pdf/Corporate_Portal_White_Paper.pdf, letzter Zugriff: 09.02.2004, 1999.

[Plumtree 2002]

Plumtree: The Corporate Portal Market in 2002 – A Synthesis of New, Comprehensive Survey Results and Recent Analyst Research on How Organizations are Deploying Portals, <http://dagda.shef.ac.uk/inf201/plumtreesurvey.pdf>, letzter Zugriff: 19.12.2002, 2002.

[Powell 1996]

Powell, W. W.: Trust-based forms of governance, in: Kramer, R. M.; Rylar, T.R. (eds.): Trust in organizations: Frontiers of strategy and research, Sage, Thousand Oaks, CA, 1996, pp. 51-67.

[Probst/Raub/Romhardt 2003]

Probst, G.; Raub, S.; Romhardt, K.: Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen; 4. Auflage, Gabler Verlag, Wiesbaden, 2003.

[Puschmann 2003]

Puschmann, T.: Collaboration Portale – Architektur, Integration, Umsetzung und Beispiele, Dissertation, Universität St. Gallen, Difo-Druck, Bamberg, 2003.

[Puschmann/Alt 2004]

Puschmann, T.; Alt, R.: Process Portals – Architecture and Integration, in: Proceedings of the 37th Hawaii International Conference on System Sciences, 2004.

[Radeke 1996]

Radeke, E.: Federation and migration among database systems, Shaker, Aachen, 1996.

[Rahm 1994]

Rahm, E.: Mehrrechner-Datenbanksysteme – Grundlagen der verteilten und parallelen Datenbankverarbeitung, Addison-Wesley, Bonn, 1994.

[Reynolds/Koulopoulos 1999]

Reynolds, H.; Koulopoulos, T. M.: Enterprise knowledge has a face, in: Intelligent Enterprise, 2(5), 29-34, (March), http://www.intelligententerprise.com/db_area/archives/1999/993003/feat1.jhtml?_requestid=590696, letzter Zugriff: 04.02.2004, 1999.

[Riempp 1998]

Riempp, G.: Wide Area Workflow Management – Creating Partnerships for the 21st Century, Springer, Berlin u.a., 1998.

[Ring 1999]

Ring, P. S.: Processes Facilitating Reliance on Trust in Inter-Organizational Networks, in: Ebers, M. (ed.): The formation of inter organizational networks, Oxford University Press, New York, 1999, pp. 113-145.

[Roberts-Witt 2000]

Roberts-Witt, S. L.: Making Sense of Portal Pandemonium, <http://www.kmmag.com/articles/default.asp?ArticleID=84&Keywords=Making++AND+sense++AND+portal++AND+pandemonium>, letzter Zugriff: 09.02.2004, 2000.

[Rütschlin 2001]

Rütschlin, J.: Ein Portal – Was ist das eigentlich?, in: Bauknecht, K.; Brauner, W.; Mück, T. (Hrsg.). Informatik 2001: Wirtschaft und Wissenschaft in der Network Economy – Visionen und Wirklichkeit. Tagungsband der GI/OCG-Jahrestagung, 25.-28. September 2001, Universität Wien, 2001, pp. 691-696.

[Rupprecht-Däullary 1994]

Rupprecht-Däullary, M.: Zwischenbetriebliche Kooperation: Möglichkeiten und Grenzen durch neue Informations- und Kommunikationstechnologien, Dt. Universitäts-Verlag, Wiesbaden, 1994.

[SAP 2004]

SAP: SAP Enterprise Portal – Im 5. Gang aus dem Vollen schöpfen, <http://www.sap.com/germany/solutions/netweaver/enterpriseportal/index.asp>, letzter Zugriff: 12.06.2004, 2004.

[Schelp/Winter 2002]

Schelp, J.; Winter, R.: Enterprise Portals und Enterprise Application Integration – Begriffsbestimmung und Integrationskonzepte, in: HMD, Heft 225, Juni 2002, dPunkt.verlag GmbH, Heidelberg, 2002, pp. 6-20.

[Schneider 1985]

Schneider, D.: Die Unhaltbarkeit des Transaktionskostenansatzes für die „Markt oder Unternehmung“ Diskussion, in: Zeitschrift für Betriebswirtschaft, Vol. 55, Nr. 12, 1985, pp. 1237-1254.

[Schubert/Klein 1997]

Schubert, K.; Klein, M.: Das Politiklexikon, J.H.W. Dietz, Bonn, 1997.

[Serain 2002]

Serain, D.: Middleware and enterprise application integration, Springer, London, 2002.

[Sheth/Larson 1990]

Sheth, A. P.; Larson, J. A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases, in: ACM Computing Surveys (CSUR) Vol. 22, Issue 3 (September) – Special issue on heterogeneous databases, 1990, pp. 183-236.

[Shilakes/Tylman 1998]

Shilakes, C. C.; Tylman, J.; Merrill Lynch: Enterprise Information Portals, Merrill Lynch, 1998.

[Snell/Tidwell/Kulchenko 2002]

Snell, J.; Tidwell, T.; Kulchenko, P. (eds.): Webservice-Programmierung mit SOAP, O'Reilly Verlag, Köln, 2002.

[Starbuck 1992]

Starbuck, W. H.: Learning by Knowledge-Intensive Firms, in: Journal of Management Studies; November 1992, Vol. 29 Issue 6, 1992, pp. 713-741.

[Sydow/Winand 1998]

Sydow, J.; Winand, U.: Unternehmensvernetzung und -virtualisierung, in: Winand, U.; Nathusius, K. (Hrsg.): Unternehmensnetzwerke und virtuelle Organisationen, Schäffer-Poeschel, Stuttgart, 1998, pp. 11-31.

[Thompson/Leue/Kropp 2003]

Thompson, R.; Leue, C.; Kropp, A.; OASIS: Web Services for Remote Portlets Specification v 1.0, in: OASIS, <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>, letzter Zugriff: 16.01.2004, 2003.

[Tsichritzis/Klug 1978]

Tsichritzis, D.; Klug, A.: The ANSI/X3/SPARC DBMS framework report of the study group on database management systems, in: Information Systems, Vol. 3, No. 3, 1978, pp. 173-191.

[Ulrich/Hill 1979]

Ulrich, H.; Hill, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre, in: Raffèe, H., Abel, B.(Ed.). Wissenschaftstheoretische Grundfragen der Wirtschaftswissenschaften, Poeschel, Stuttgart, 1979, pp. 161-190.

[Veijalainen/Popescu-Zeletin 1988]

Veijalainen, J.; Popescu-Zeletin, R.: Multidatabase Systems in ISO/OSI Environment, in: Malagardis, N.; Williams, T. (eds.), Standards in Information Technology and Industrial Control, North-Holland, Netherlands, 1988, pp. 83-97.

[Visser et al. 1997]

Visser, P. R. S.; Jones, D. M.; Bench-Capon, T. J. M.; Shave, M. J. R.: An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability, in: AAAI 1997 Spring Symposium on Ontological Engineering, Stanford University, USA; also appeared as AAAI Technical Report SS-97-06, 1997.

[W3C 2001]

The World Wide Web Consortium (W3C); W3C: Web Services Description Language (WSDL) 1.1 – W3C Note 15 March 2001, <http://www.w3.org/TR/wsdl.html>, letzter Zugriff: 12.03.2004, 2001.

[W3C 2003a]

The World Wide Web Consortium (W3C); W3C: SOAP Version 1.2 Part 1: Messaging Framework – W3C Recommendation 24 Juni 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>, letzter Zugriff: 25.02.2004, 2003.

[W3C 2003b]

The World Wide Web Consortium (W3C); W3C: SOAP Version 1.2 Part 0: Primer – W3C Recommendation 24 June 2003, <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>, letzter Zugriff: 25.02.2004, 2003.

[W3C 2004a]

The World Wide Web Consortium (W3C); W3C: Web Services Glossary – W3C Working Group Note 11 February 2004, <http://www.w3.org/TR/2003/WD-ws-gloss-20030808/>, letzter Zugriff: 25.02.2004, 2004.

[W3C 2004b]

The World Wide Web Consortium (W3C); W3C: Web Services Architecture – W3C Working Group Note 11 February 2004, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, letzter Zugriff: 25.02.2004, 2004.

[Wason et al. 2003]

Wason, T.; Cantor, S.; Hodges, J.; Kemp, J.; Thompson, P.: Liberty ID-FF Architecture Overview, <http://www.projectliberty.org/specs/liberty-idff-arch-overview-v1.2.pdf>, letzter Zugriff: 10.02.2004, 2003.

[Watson-Manheim/Bélanger 2002]

Watson-Manheim, M. B.; Bélanger, F.: Support for Communication-Based Work Processes in Virtual Work, in: e-Service Journal – 2002 Vol. 1 Issue 3 (May), <http://search.epnet.com/direct.asp?an=7295653&db=buh>, letzter Zugriff: 12.12.2002, 2002, pp. 61-82.

[Webber/Dutton 2000]

Webber, D.; Dutton, A.: Understanding ebXML, UDDI, XML/EDI, http://www.xml.org/xml/feature_articles/2000_1107_miller.shtml, letzter Zugriff: 25.05.2004, 2000.

[White 1999]

White, C.: The enterprise information portal marketplace, in: Decision processing brief, DP-99-01, Database Associates International, Morgan Hill, CA, <http://www.decisionprocessing.com/papers/eip1.doc>, letzter Zugriff: 12.09.2003, 1999.

[Whyte et al. 1991]

Whyte, W. F. (ed.): Participatory action research, Sage, Newbury Park u.a., 1991.

[Wiederhold 1992]

Wiederhold, G.: Mediators in the Architecture of Future Information Systems, in: IEEE Computers, Vol. 25, No. 3, 1992, pp. 38-49.

[Williamson 1975]

Williamson, O. E.: *Markets and Hierarchies*, Free Press, New York, 1975.

[Williamson 1985]

Williamson, O. E.: *The Economic Institutions of Capitalism*, Free Press, New York, 1985.

[Williamson 1993]

Williamson, O. E.: *The Logic of Economic Organization*, in: Williamson, O.E.; Winter, S. G. (eds.): *The Nature of the Firm*, Oxford University Press, New York, Oxford, 1993, pp. 90-116.

[Wilson 1995]

Wilson, P.: *Unused Relevant Information in Research and Development*, in: *Journal of the American Society for Information Science*, Vol. 46, No. 1, 1995, pp. 45-51.

[Winch/Schneider 1993]

Winch, G.; Schneider, E.: *Managing the knowledge-based organization: The case of architectural practice*, in: *Journal of Management Studies*; November 1993, Vol. 30 Issue 6, 1993, pp. 923-938.

[Yang/Wijesekera/Jajodia 2001]

Yang, J.; Wijesekera, D.; Jajodia, S.: *Subject Switching Algorithms for Federated Databases*, in: submitted to 15th Annual IFIP WG 11.3 Working Conference on Database Security, 2001, pp. 61-74.

[Zhang et al. 2001]

Zhang, L.; Chan, S. C. F.; Ng, V. T. Y.; Yu, K. M.: *Enterprise Virtualisation: Concept, Methodology, and Implementation*, in: *International Journal of Advanced Manufacturing Technology – 2001* Vol. 18, Issue 3, 2001, pp. 217-234.

9 Anhang

Ergänzend zu der weitgehend abstrakten Darstellung der vier funktionalen Bestandteile der Portalkopplung im Rahmen des Abschnitts 4.6 werden im Folgenden die zugehörigen Schnittstellenoperationen mit deren Datentypen und Nachrichten detailliert aufgelistet. Diese bilden die Grundlage für eine informationstechnische Umsetzung des in Kapitel 4 vorgestellten Kopplungskonzepts und werden entsprechend bei der Kopplung zweier G8-Portale berücksichtigt (vgl. Abschnitt 5.2). Die Beschreibung erfolgt unter Verwendung der Web Service Description Language (vgl. Abschnitt 5.1.2.2). Dabei handelt es sich jedoch ausschließlich um Fragmente der WSDL-Definition des Daten- und Funktionsmodells, nicht jedoch um eine vollständige WSDL-Datei. Auf die Darstellung von Namespaces etc. wurde zur Verbesserung der Übersichtlichkeit an dieser Stelle verzichtet.

9.1 WSDL-Typen

```

<!-- -----
                                WSDL-Typen beim Portlet-Routing
----- -->

<!-- Beschreibung des Inhalts einer gerouteten Nachricht -->
<schema>
  <element name="routeMessage" type="xsd:anyType"/>
</schema>

<!-- Beschreibung einer Antwort bzgl des Inhalts einer gerouteten Nachricht -->
<schema>
  <element name="routeMessageReturn" type="xsd:anyType"/>
</schema>

<!-- -----
                                WSDL-Typen zum Anbieten von Portlets – 2. Schicht
----- -->

<!-- WSDL-Type zum Abruf von Portlet-Definitionen -->
<!-- Beschreibung einer angebotenen Portlet-Definition -->
<complexType name="PortletDefinitionData">
  <sequence>
    <element name="maxCount" nillable="true" type="xsd:string"/>
    <element name="portletCategory" nillable="true" type="impl:ArrayOf_xsd_string"/>
    <element name="portletDefKey" nillable="true" type="xsd:string"/>
    <element name="portletName" nillable="true" type="impl:ArrayOf_xsd_string"/>
    <element name="portletTypeIdentifier" nillable="true" type="xsd:string"/>
    <element name="providerPortalID" nillable="true" type="xsd:string"/>
    <element name="showOptions" type="xsd:int"/>
  </sequence>
</complexType>

```

```
        <element name="supportedLanguages" nillable="true" type="impl:ArrayOf_xsd_string"/>
    </sequence>
</complexType>

<!-- WSDL-Types zur Nutzung von Portlet-Instanzen -->

<!-- Beschreibung eines Portlets zur Erzeugung und Referenzierung -->
<complexType name="PortalPortletData">
    <sequence>
        <element name="creatorPortalID" nillable="true" type="xsd:string"/>
        <element name="defaultPortletName" nillable="true" type="xsd:string"/>
        <element name="deleted" type="xsd:boolean"/>
        <element name="ownerPortalID" nillable="true" type="xsd:string"/>
        <element name="portletDefKey" nillable="true" type="xsd:string"/>
        <element name="portletTypeIdentifier" nillable="true" type="xsd:string"/>
        <element name="position" type="xsd:int"/>
        <element name="positionModifiedTime" type="xsd:long"/>
        <element name="remotePortletKey" nillable="true" type="xsd:string"/>
        <element name="routedPortlet" type="xsd:boolean"/>
        <element name="servicePortalID" nillable="true" type="xsd:string"/>
        <element name="unknownPortlet" type="xsd:boolean"/>
    </sequence>
</complexType>

<!-- Beschreibung einer Anfrage an ein Portlet bei einem entfernten Aufruf -->
<complexType name="PortletRequestData">
    <sequence>
        <element name="baseURL" nillable="true" type="xsd:string"/>
        <element name="consumerPortalID" nillable="true" type="xsd:string"/>
        <element name="currentMode" type="xsd:int"/>
        <element name="defaultLanguage" nillable="true" type="xsd:string"/>
        <element name="device" type="xsd:int"/>
        <element name="language" nillable="true" type="xsd:string"/>
        <element name="pageKey" nillable="true" type="xsd:string"/>
        <element name="picturePath" nillable="true" type="xsd:string"/>
        <element name="portletKey" nillable="true" type="xsd:string"/>
        <element name="previousMode" type="xsd:int"/>
        <element name="remoteUserLogin" nillable="true" type="xsd:string"/>
        <element name="themePath" nillable="true" type="xsd:string"/>
        <element name="urlPortletKey" nillable="true" type="xsd:string"/>
        <element name="webAppPath" nillable="true" type="xsd:string"/>
        <element name="windowLayout" type="xsd:int"/>
        <element name="windowMode" type="xsd:int"/>
    </sequence>
</complexType>
```

<!-- Beschreibung einer Antwort eines Portlets nach einem entfernten Aufruf -->

```
<complexType name="PortletResponseData">
  <sequence>
    <element name="markup" nillable="true" type="xsd:string"/>
    <element name="session" nillable="true" type="xsd:string"/>
    <element name="specialActions" nillable="true" type="tns7:RemoteContainer"/>
  </sequence>
</complexType>
```

<!-- Beschreibung einer Zuordnung Name und Werte -->

```
<complexType name="RemoteContainer">
  <sequence>
    <element name="keys" nillable="true" type="impl:ArrayOf_xsd_string"/>
    <element name="values" nillable="true" type="impl:ArrayOfArrayOf_xsd_string"/>
  </sequence>
</complexType>
```

<!-- Beschreibung des HttpServletRequests für die Nutzung bei einem entfernten Portlet-Aufruf -->

```
<complexType name="RemoteHttpServletRequestBean">
  <sequence>
    <element name="authType" nillable="true" type="xsd:string"/>
    <element name="characterEncoding" nillable="true" type="xsd:string"/>
    <element name="contentLength" type="xsd:int"/>
    <element name="contentType" nillable="true" type="xsd:string"/>
    <element name="contextPath" nillable="true" type="xsd:string"/>
    <element name="headersContainer" nillable="true" type="tns7:RemoteContainer"/>
    <element name="method" nillable="true" type="xsd:string"/>
    <element name="parameterMapContainer" nillable="true" type="tns7:RemoteContainer"/>
    <element name="parametersContainer" nillable="true" type="tns7:RemoteContainer"/>
    <element name="pathInfo" nillable="true" type="xsd:string"/>
    <element name="pathTranslated" nillable="true" type="xsd:string"/>
    <element name="procotol" nillable="true" type="xsd:string"/>
    <element name="protocol" nillable="true" type="xsd:string"/>
    <element name="queryString" nillable="true" type="xsd:string"/>
    <element name="remoteAddr" nillable="true" type="xsd:string"/>
    <element name="remoteHost" nillable="true" type="xsd:string"/>
    <element name="remoteUser" nillable="true" type="xsd:string"/>
    <element name="requestURI" nillable="true" type="xsd:string"/>
    <element name="requestURLString" nillable="true" type="xsd:string"/>
    <element name="requestedSessionId" nillable="true" type="xsd:string"/>
    <element name="requestedSessionIdFromCookie" type="xsd:boolean"/>
    <element name="requestedSessionIdFromURL" type="xsd:boolean"/>
    <element name="requestedSessionIdFromUrl" type="xsd:boolean"/>
    <element name="requestedSessionIdValid" type="xsd:boolean"/>
    <element name="scheme" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>
```

```

    <element name="secure" type="xsd:boolean"/>
    <element name="serverName" nillable="true" type="xsd:string"/>
    <element name="serverPort" type="xsd:int"/>
    <element name="servletPath" nillable="true" type="xsd:string"/>
  </sequence>
</complexType>

<!-- -----
WSDL-Typen zum Anbieten von Seiten und Plätzen und zur gemeinsamen Nutzung von Plätzen – 3. Schicht
----- -->

<!-- Beschreibung eines Portal-Containers -->
<complexType name="PortalPortletContainerData">
  <sequence>
    <element name="childContainer" nillable="true" type="impl:ArrayOf_PortalPortletContainerData"/>
    <element name="containerKey" nillable="true" type="xsd:string"/>
    <element name="containerType" type="xsd:int"/>
    <element name="containerWidth" nillable="true" type="xsd:string"/>
    <element name="portlets" nillable="true" type="impl:ArrayOf_PortalPortletData"/>
    <element name="position" type="xsd:int"/>
    <element name="positionModifiedTime" type="xsd:long"/>
  </sequence>
</complexType>

<!-- Beschreibung der Container-Hierarchie einer Portal-Seite -->
<complexType name="PortalMainPortletContainerData">
  <sequence>
    <element name="deletedContainer" nillable="true" type="impl:ArrayOf_PortalPortletContainerData"/>
    <element name="deletedPortlets" nillable="true" type="impl:ArrayOf_PortalPortletData"/>
    <element name="firstLevelContainer" nillable="true" type="impl:ArrayOf_PortalPortletContainerData"/>
  </sequence>
</complexType>

<!-- Beschreibung einer Portal-Seite -->
<complexType name="PortalPageData">
  <sequence>
    <element name="container" nillable="true" type="tns7:PortalMainPortletContainerData"/>
    <element name="containerOrPortletsLastChanged" type="xsd:long"/>
    <element name="deleted" type="xsd:boolean"/>
    <element name="device" type="xsd:int"/>
    <element name="linkUrl" nillable="true" type="xsd:string"/>
    <element name="name" nillable="true" type="xsd:string"/>
    <element name="pageAttrib" type="xsd:int"/>
    <element name="pageKey" nillable="true" type="xsd:string"/>
    <element name="pagePropsLastChanged" type="xsd:long"/>
  </sequence>

```

```

        <element name="position" type="xsd:int"/>
    </sequence>
</complexType>

<!-- Beschreibung eines Portal-Platz -->
<complexType name="PortalPlaceData">
    <sequence>
        <element name="accessPlaceType" type="xsd:int"/>
        <element name="accessUserType" type="xsd:int"/>
        <element name="category" nillable="true" type="xsd:string"/>
        <element name="description" nillable="true" type="xsd:string"/>
        <element name="lastChanged" type="xsd:long"/>
        <element name="modified" type="xsd:boolean"/>
        <element name="name" nillable="true" type="xsd:string"/>
        <element name="otherConsumerPortals" nillable="true" type="xsd:string"/>
        <element name="pages" nillable="true" type="impl:ArrayOf_PortalPageData"/>
        <element name="pagesLastChanged" type="xsd:long"/>
        <element name="placeKey" nillable="true" type="xsd:string"/>
        <element name="placePropsLastChanged" type="xsd:long"/>
        <element name="remoteAdminRights" type="xsd:int"/>
        <element name="remotePlaceAdmin" nillable="true" type="xsd:string"/>
        <element name="syncTime" type="xsd:int"/>
    </sequence>
</complexType>

```

9.2 WSDL-Nachrichten

```

<!-- -----
                        WSDL-Nachrichten beim Portlet-Routing
----- -->

<!-- Nachricht zum Routing einer Anfrage bzgl. eines entfernten Portlets -->
<wsdl:message name="routeMessageRequest">
    <wsdl:part element="routeMessage" name="part"/>
</wsdl:message>

<!-- Nachricht als Antwort auf die Nachricht zum Routing einer Anfrage bzgl. eines entfernten Portlets -->
<wsdl:message name="routeMessageResponse">
    <wsdl:part element="impl:routeMessageReturn" name="routeMessageReturn"/>
</wsdl:message>

```

```
<!-- -----  
                WSDL-Nachrichten zur Sicherstellung der Basiskommunikation – 1. Schicht  
----- -->  
  
<!-- Nachricht zum Erzeugen einer Sitzung -->  
<wsdl:message name="createSessionResponse">  
    <wsdl:part name="createSessionReturn" type="soapenc:string"/>  
</wsdl:message>  
  
<!-- Nachricht als Antwort auf das Erzeugen einer Sitzung -->  
<wsdl:message name="createSessionRequest">  
</wsdl:message>  
  
<!-- Nachricht zum Beenden einer Sitzung -->  
<wsdl:message name="destroySessionRequest">  
</wsdl:message>  
  
<!-- Nachricht als Antwort zum Beenden einer Sitzung -->  
<wsdl:message name="destroySessionResponse">  
    <wsdl:part name="destroySessionReturn" type="xsd:boolean"/>  
</wsdl:message>  
  
<!-- -----  
                WSDL-Nachrichten zum Anbieten von Portlets – 2. Schicht  
----- -->  
  
<!-- WSDL-Nachrichten zum Abruf von Portlet-Definitionen -->  
<!-- Nachricht zum Abruf der verfügbaren Portlet-Definitionen -->  
<wsdl:message name="getAvailablePortletsRequest">  
    <wsdl:part name="device" type="xsd:int"/>  
</wsdl:message>  
  
<!-- Nachricht als Antwort mit den verfügbaren Portlet-Definitionen -->  
<wsdl:message name="getAvailablePortletsResponse">  
    <wsdl:part name="getAvailablePortletsReturn" type="impl:ArrayOf_PortletDefinitionData"/>  
</wsdl:message>  
  
<!-- Nachricht zur Nutzung von Portlet-Instanzen -->  
<!-- Nachricht zum Erzeugen einer neuen Portlet-Instanz -->  
<wsdl:message name="createPortletInstanceRequest">  
    <wsdl:part name="remoteUserLogin" type="soapenc:string"/>  
    <wsdl:part name="portletDefKey" type="soapenc:string"/>  
    <wsdl:part name="device" type="xsd:int"/>  
    <wsdl:part name="creatorPortalID" type="soapenc:string"/>  
    <wsdl:part name="consumerPortalID" type="soapenc:string"/>  
</wsdl:message>
```

```
</wsdl:message>

<!-- Nachricht als Antwort auf das Erzeugen einer neuen Portlet-Instanz -->
<wsdl:message name="createPortletInstanceResponse">
  <wsdl:part name="createPortletInstanceReturn" type="impl:ArrayOf_xsd_string"/>
</wsdl:message>

<!-- Nachricht zum Erzeugen eines Clones einer bestehenden Portlet-Instanz -->
<wsdl:message name="createPortletCloneRequest">
  <wsdl:part name="existingRemoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="newRemoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="portletKey" type="soapenc:string"/>
  <wsdl:part name="placeKey" type="soapenc:string"/>
  <wsdl:part name="device" type="xsd:int"/>
  <wsdl:part name="ownerPortalID" type="soapenc:string"/>
  <wsdl:part name="creatorPortalID" type="soapenc:string"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
  <wsdl:part name="copyPortletKey" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht als Antwort auf das Erzeugen eines Clones einer bestehenden Portlet-Instanz -->
<wsdl:message name="createPortletCloneResponse">
  <wsdl:part name="createPortletCloneReturn" type="impl:ArrayOf_xsd_string"/>
</wsdl:message>

<!-- Nachricht zum Löschen einer Portlet-Instanz -->
<wsdl:message name="destroyPortletInstanceRequest">
  <wsdl:part name="remoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="remotePortletKey" type="soapenc:string"/>
  <wsdl:part name="device" type="xsd:int"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort ob das Löschen einer Portlet-Instanz erfolgreich war -->
<wsdl:message name="destroyPortletInstanceResponse">
  <wsdl:part name="destroyPortletInstanceReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht zum Aufruf eines entfernten Portlets zur Erzeugung des Portlet-Inhalts -->
<wsdl:message name="getPortletMarkupRequest">
  <wsdl:part name="requestData" type="PortletRequestData"/>
</wsdl:message>
```

```
<!-- Nachricht als Antwort auf den Aufruf eines entfernten Portlets zur Erzeugung des Portlet-Inhalts -->
<wsdl:message name="getPortletMarkupResponse">
  <wsdl:part name="getPortletMarkupReturn" type="PortletResponseData"/>
</wsdl:message>

<!-- Nachricht zur Aktualisierung der Portlet-Eigenschaften eines entfernten Portlets -->
<wsdl:message name="updatePortletPropertiesRequest">
  <wsdl:part name="sourceRemoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="sourcePortletKey" type="soapenc:string"/>
  <wsdl:part name="sourceConsumerPortalID" type="soapenc:string"/>
  <wsdl:part name="destRemoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="destPortletKey" type="soapenc:string"/>
  <wsdl:part name="destConsumerPortalID" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort auf die Aktualisierung der Portlet-Eigenschaften eines entfernten Portlets -->
<wsdl:message name="updatePortletPropertiesResponse">
  <wsdl:part name="updatePortletPropertiesReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht zum Aufruf einer Aktion eines entfernten Portlets (ohne dass ein Input-Stream übergeben wird) -->
<wsdl:message name="invokePortletActionRequest">
  <wsdl:part name="remoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="remotePortletKey" type="soapenc:string"/>
  <wsdl:part name="device" type="xsd:int"/>
  <wsdl:part name="actionIdentifier" type="xsd:int"/>
  <wsdl:part name="container" type="RemoteContainer"/>
  <wsdl:part name="containerUp" type="RemoteContainer"/>
  <wsdl:part name="inputContainer" type="RemoteContainer"/>
  <wsdl:part name="contentBaseUrl" type="soapenc:string"/>
  <wsdl:part name="webAppPath" type="soapenc:string"/>
  <wsdl:part name="themePath" type="soapenc:string"/>
  <wsdl:part name="picturePath" type="soapenc:string"/>
  <wsdl:part name="consumerPortletKey" type="soapenc:string"/>
  <wsdl:part name="req" type="RemoteHttpServletRequestBean"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort auf den Aufruf einer Aktion eines entfernten Portlets (ohne dass ein Input-Stream übergeben wird) -->
<wsdl:message name="invokePortletActionResponse">
  <wsdl:part name="invokePortletActionReturn" type="soapenc:string"/>
</wsdl:message>
```

<!-- Nachricht zum Aufruf einer Aktion eines entfernten Portlets (bei Übergabe eines Input-Streams) -->

```
<wsdl:message name="invokePortletActionRequest1">
  <wsdl:part name="remoteUserLogin" type="soapenc:string"/>
  <wsdl:part name="remotePortletKey" type="soapenc:string"/>
  <wsdl:part name="device" type="xsd:int"/>
  <wsdl:part name="actionIdentifier" type="xsd:int"/>
  <wsdl:part name="container" type="RemoteContainer"/>
  <wsdl:part name="containerUp" type="RemoteContainer"/>
  <wsdl:part name="inputContainer" type="RemoteContainer"/>
  <wsdl:part name="contentBaseUrl" type="soapenc:string"/>
  <wsdl:part name="webAppPath" type="soapenc:string"/>
  <wsdl:part name="themePath" type="soapenc:string"/>
  <wsdl:part name="picturePath" type="soapenc:string"/>
  <wsdl:part name="consumerPortletKey" type="soapenc:string"/>
  <wsdl:part name="req" type="RemoteHttpServletRequestBean"/>
  <wsdl:part name="inputStream" type="apachesoap:DataHandler"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
</wsdl:message>
```

<!-- Nachricht als Antwort auf den Aufruf einer Aktion eines entfernten Portlets (bei Übergabe eines Input-Streams) -->

```
<wsdl:message name="invokePortletActionResponse1">
  <wsdl:part name="invokePortletActionReturn" type="soapenc:string"/>
</wsdl:message>
```

<!-- Nachricht zum Übertragen der Verantwortung für die Portlets mit dem PortletDef-Key auf ein anderes Portal -->

```
<wsdl:message name="transferPortletsToNewPortletOwnerRequest">
  <wsdl:part name="portletDefKey" type="soapenc:string"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
</wsdl:message>
```

<!-- Nachricht als Antwort auf das Übertragen der Verantwortung für die Portlets mit dem PortletDef-Key auf ein anderes Portal -->

```
<wsdl:message name="transferPortletsToNewPortletOwnerResponse">
  <wsdl:part name="transferPortletsToNewPortletOwnerReturn" type="xsd:boolean"/>
</wsdl:message>
```

<!-- Nachricht zum Übertragen der Verantwortung für ein Portlet auf ein anderes Portal -->

```
<wsdl:message name="transferPortletToNewOwnerRequest">
  <wsdl:part name="remotePortletKey" type="soapenc:string"/>
  <wsdl:part name="device" type="xsd:int"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
  <wsdl:part name="newConsumerPortalID" type="soapenc:string"/>
  <wsdl:part name="transferComplete" type="xsd:boolean"/>
</wsdl:message>
```

```
<!-- Nachricht als Antwort auf die Übertragen der Verantwortung für ein Portlet auf ein anderes Portal -->
<wsdl:message name="transferPortletToNewOwnerResponse">
</wsdl:message>

<!-- -----
WSDL-Nachrichten zum Anbieten von Seiten und Plätzen und zur gemeinsamen Nutzung von Plätzen – 3. Schicht
----- -->

<!-- Nachrichten zum Anbieten von Seiten -->
<!-- Nachricht zum Abruf der entfernt definierten angebotenen Seiten -->
<wsdl:message name="getAvailablePagesRequest">
  <wsdl:part name="currentPages" type="impl:ArrayOf_PortalPageData"/>
  <wsdl:part name="device" type="xsd:int"/>
</wsdl:message>

<!-- Nachricht als Antwort auf den Abruf der entfernt definierten angebotenen Seiten -->
<wsdl:message name="getAvailablePagesResponse">
  <wsdl:part name="getAvailablePagesReturn" type="impl:ArrayOf_PortalPageData"/>
</wsdl:message>

<!-- Nachrichten zum Anbieten von Plätzen und zur gemeinsamen Nutzung von Plätzen -->
<!-- Nachricht zum Erzeugen eines Platzes bei einem entfernten Portal -->
<wsdl:message name="createPlaceRequest">
  <wsdl:part name="pPlaceData" type="PortalPlaceData"/>
</wsdl:message>

<!-- Nachricht als Antwort auf das Erzeugen eines Platzes bei einem entfernten Portal -->
<wsdl:message name="createPlaceResponse">
  <wsdl:part name="createPlaceReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht zum Auffordern eines Consumer-Portals einen Consumer-Place zu löschen -->
<wsdl:message name="deleteConsumerPlaceRequest">
  <wsdl:part name="placeKey" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort auf das Auffordern eines Consumer-Portals einen Consumer-Place zu löschen -->
<wsdl:message name="deleteConsumerPlaceResponse">
  <wsdl:part name="deleteConsumerPlaceReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht teilt dem Coordinator-Portal mit, dass ein Consumer-Portal die Nutzung eines geteilten Platzes beendet-->
<wsdl:message name="terminateConsumerPlaceRequest">
  <wsdl:part name="placeKey" type="soapenc:string"/>
</wsdl:message>
```

```

<!-- Nachricht als Antwort, dass ein Consumer-Portal die Nutzung eines geteilten Platzes beendet -->
<wsdl:message name="terminateConsumerPlaceResponse">
  <wsdl:part name="terminateConsumerPlaceReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht zum Auffordern eines entfernten Portals zur Synchronisation mit dem Coordinator-Place -->
<wsdl:message name="requestPlaceSynchronisationRequest">
  <wsdl:part name="placeKey" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort auf das Auffordern eines entfernten Portals zur Synchronisation mit dem Coordinator-Place -->
<wsdl:message name="requestPlaceSynchronisationResponse">
  <wsdl:part name="requestPlaceSynchronisationReturn" type="xsd:boolean"/>
</wsdl:message>

<!-- Nachricht zur Synchronisation eines Consumer-Place mit dem Coordinator-Place -->
<wsdl:message name="syncConsumerPlaceRequest">
  <wsdl:part name="pPlaceData" type="PortalPlaceData"/>
</wsdl:message>

<!-- Nachricht als Antwort auf die Synchronisation eines Consumer-Place mit dem Coordinator-Place -->
<wsdl:message name="syncConsumerPlaceResponse">
  <wsdl:part name="syncConsumerPlaceReturn" type="PortalPlaceData"/>
</wsdl:message>

<!-- Nachricht neues Portal der Liste zugriffsberechtigter Portale auf einen Platz beim Consumer-Portal hinzuzufügen -->
<wsdl:message name="addConsumerPlaceAccessRightRequest">
  <wsdl:part name="placeKey" type="soapenc:string"/>
  <wsdl:part name="consumerPortalID" type="soapenc:string"/>
</wsdl:message>

<!-- Nachricht als Antwort neues Portal der Liste zugriffsberechtigter Portale auf einen Platz beim Consumer-Portal hin-->
<wsdl:message name="addConsumerPlaceAccessRightResponse">
  <wsdl:part name="addConsumerPlaceAccessRightReturn" type="xsd:boolean"/>
</wsdl:message>

```

9.3 WSDL-Operationen

```

<!-- -----
                        WSDL-Nachrichten beim Portlet-Routing
----- -->

<!-- Operation zum Routing einer Nachricht bzgl. eines entfernten Portlets -->
<wsdl:operation name="routeMessage">
  <wsdl:input message="impl:routeMessageRequest" name="routeMessageRequest"/>
  <wsdl:output message="impl:routeMessageResponse" name="routeMessageResponse"/>
</wsdl:operation>

```

```

<!-- -----
                        WSDL-Operationen zur Sicherstellung der Basiskommunikation – 1. Schicht
----- -->

<!-- Operation zum Aufbau einer Sitzung -->
<wsdl:operation name="createSession">
  <wsdl:input message="impl:createSessionRequest" name="createSessionRequest"/>
  <wsdl:output message="impl:createSessionResponse" name="createSessionResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zum Abbau einer Sitzung -->
<wsdl:operation name="destroySession">
  <wsdl:input message="impl:destroySessionRequest" name="destroySessionRequest"/>
  <wsdl:output message="impl:destroySessionResponse" name="destroySessionResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- -----
                        WSDL-Operationen zum Anbieten von Portlets – 2. Schicht
----- -->

<!-- WSDL-Operationen zum Abruf von Portlet-Definitionen -->
<!-- Operation zum Abruf der verfügbaren Portlet-Definitionen -->
<wsdl:operation name="getAvailablePortlets" parameterOrder="device">
  <wsdl:input message="impl:getAvailablePortletsRequest" name="getAvailablePortletsRequest"/>
  <wsdl:output message="impl:getAvailablePortletsResponse" name="getAvailablePortletsResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- WSDL-Operationen zur Nutzung von Portlet-Instanzen -->
<!-- Operation zum Erzeugen einer neuen Portlet-Instanz -->
<wsdl:operation name="createPortletInstance" parameterOrder="remoteUserLogin portletDefKey device
creatorPortalID consumerPortalID">
  <wsdl:input message="impl:createPortletInstanceRequest" name="createPortletInstanceRequest"/>
  <wsdl:output message="impl:createPortletInstanceResponse" name="createPortletInstanceResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zum Erzeugen eines Clones einer bestehenden Portlet-Instanz -->
<wsdl:operation name="createPortletClone" parameterOrder="existingRemoteUserLogin newRemoteUserLogin
portletKey placeKey device ownerPortalID creatorPortalID consumerPortalID copyPortletKey">
  <wsdl:input message="impl:createPortletCloneRequest" name="createPortletCloneRequest"/>
  <wsdl:output message="impl:createPortletCloneResponse" name="createPortletCloneResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

```

```
<!-- Operation zum Löschen einer Portlet-Instanz -->
<wsdl:operation name="destroyPortletInstance" parameterOrder="remoteUserLogin remotePortletKey device
    consumerPortalID">
    <wsdl:input message="impl:destroyPortletInstanceRequest" name="destroyPortletInstanceRequest"/>
    <wsdl:output message="impl:destroyPortletInstanceResponse" name="destroyPortletInstanceResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zum Aufruf eines entfernten Portlets zur Erzeugung des Portlet-Inhalts -->
<wsdl:operation name="getPortletMarkup" parameterOrder="requestData">
    <wsdl:input message="impl:getPortletMarkupRequest" name="getPortletMarkupRequest"/>
    <wsdl:output message="impl:getPortletMarkupResponse" name="getPortletMarkupResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zur Aktualisierung der Portlet-Eigenschaften eines entfernten Portlets -->
<wsdl:operation name="updatePortletProperties" parameterOrder="sourceRemoteUserLogin sourcePortletKey
    sourceConsumerPortalID destRemoteUserLogin destPortletKey destConsumerPortalID">
    <wsdl:input message="impl:updatePortletPropertiesRequest" name="updatePortletPropertiesRequest"/>
    <wsdl:output message="impl:updatePortletPropertiesResponse" name="updatePortletPropertiesResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zum Aufruf einer Aktion eines entfernten Portlets (ohne dass ein Input-Stream übergeben wird) -->
<wsdl:operation name="invokePortletAction" parameterOrder="remoteUserLogin remotePortletKey device
    actionIdentifier container containerUp inputContainer contentBaseUrl webAppPath themePath
    picturePath consumerPortletKey req consumerPortalID">
    <wsdl:input message="impl:invokePortletActionRequest" name="invokePortletActionRequest"/>
    <wsdl:output message="impl:invokePortletActionResponse" name="invokePortletActionResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>

<!-- Operation zum Aufruf einer Aktion eines entfernten Portlets (bei Übergabe eines Input-Streams) -->
<wsdl:operation name="invokePortletAction" parameterOrder="remoteUserLogin remotePortletKey device
    actionIdentifier container containerUp inputContainer contentBaseUrl webAppPath themePath
    picturePath consumerPortletKey req inputStream consumerPortalID">
    <wsdl:input message="impl:invokePortletActionRequest1" name="invokePortletActionRequest1"/>
    <wsdl:output message="impl:invokePortletActionResponse1" name="invokePortletActionResponse1"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation zum Übertragen der Verantwortung für die Portlets mit dem PortletDef-Key auf ein anderes Portal -->

```
<wsdl:operation name="transferPortletsToNewPortletOwner" parameterOrder="portletDefKey
    consumerPortalID">
    <wsdl:input message="impl:transferPortletsToNewPortletOwnerRequest"
        name="transferPortletsToNewPortletOwnerRequest"/>
    <wsdl:output message="impl:transferPortletsToNewPortletOwnerResponse"
        name="transferPortletsToNewPortletOwnerResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation zum Übertragen der Verantwortung für ein Portlet auf ein anderes Portal -->

```
<wsdl:operation name="transferPortletToNewOwner" parameterOrder="remotePortletKey device
    consumerPortalID newConsumerPortalID transferComplete">
    <wsdl:input message="impl:transferPortletToNewOwnerRequest"
        name="transferPortletToNewOwnerRequest"/>
    <wsdl:output message="impl:transferPortletToNewOwnerResponse"
        name="transferPortletToNewOwnerResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!--

WSDL-Operationen zum Anbieten von Seiten und Plätzen und zur gemeinsamen Nutzung von Plätzen – 3. Schicht

----->

<!-- Operation zum Anbieten von Seiten -->

<!-- Operation zum Abrufen der entfernt definierten angebotenen Seiten -->

```
<wsdl:operation name="getAvailablePages" parameterOrder="currentPages device">
    <wsdl:input message="impl:getAvailablePagesRequest" name="getAvailablePagesRequest"/>
    <wsdl:output message="impl:getAvailablePagesResponse" name="getAvailablePagesResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operationen zum Anbieten von Plätzen und zur gemeinsamen Nutzung von Plätzen -->

<!-- Operation zum Erzeugen eines Platzes bei einem entfernten Portal -->

```
<wsdl:operation name="createPlace" parameterOrder="pPlaceData">
    <wsdl:input message="impl:createPlaceRequest" name="createPlaceRequest"/>
    <wsdl:output message="impl:createPlaceResponse" name="createPlaceResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation zum Auffordern eines Consumer-Portals einen Consumer-Place zu löschen -->

```
<wsdl:operation name="deleteConsumerPlace" parameterOrder="placeKey">
    <wsdl:input message="impl:deleteConsumerPlaceRequest" name="deleteConsumerPlaceRequest"/>
    <wsdl:output message="impl:deleteConsumerPlaceResponse" name="deleteConsumerPlaceResponse"/>
    <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation die dem Coordinator-Portal mitteilt, dass ein Consumer-Portal die Nutzung eines geteilten Platzes beendet -->

```
<wsdl:operation name="terminateConsumerPlace" parameterOrder="placeKey">
  <wsdl:input message="impl:terminateConsumerPlaceRequest" name="terminateConsumerPlaceRequest"/>
  <wsdl:output message="impl:terminateConsumerPlaceResponse"
    name="terminateConsumerPlaceResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation zum Auffordern eines entfernten Portals zur Synchronisation mit dem Coordinator-Place -->

```
<wsdl:operation name="requestPlaceSynchronisation" parameterOrder="placeKey">
  <wsdl:input message="impl:requestPlaceSynchronisationRequest"
    name="requestPlaceSynchronisationRequest"/>
  <wsdl:output message="impl:requestPlaceSynchronisationResponse"
    name="requestPlaceSynchronisationResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation zur Synchronisation eines Consumer-Place mit dem Coordinator-Place -->

```
<wsdl:operation name="syncConsumerPlace" parameterOrder="pPlaceData">
  <wsdl:input message="impl:syncConsumerPlaceRequest" name="syncConsumerPlaceRequest"/>
  <wsdl:output message="impl:syncConsumerPlaceResponse" name="syncConsumerPlaceResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

<!-- Operation ein neues Portal der Liste zugriffsberechtigter Portale auf einen Platz beim Consumer-Portal hinzuzufügen -->

```
<wsdl:operation name="addConsumerPlaceAccessRight" parameterOrder="placeKey consumerPortalID">
  <wsdl:input message="impl:addConsumerPlaceAccessRightRequest"
    name="addConsumerPlaceAccessRightRequest"/>
  <wsdl:output message="impl:addConsumerPlaceAccessRightResponse"
    name="addConsumerPlaceAccessRightResponse"/>
  <wsdl:fault message="impl:RemotePortalException" name="RemotePortalException"/>
</wsdl:operation>
```

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder inhaltlich übernommene Stellen sind als solche gekennzeichnet.

Die Dissertation ist keine Gemeinschaftsleistung.

Hiermit erkläre ich, dass ich noch an keiner deutschen oder ausländischen Hochschule den Antrag auf ein Promotionsverfahren gestellt habe.

Hofgeismar, im Oktober 2004

Olaf Hahl