

Modellierung und Bewertung von Integration in Krankenhausinformationssystemen

Dissertation
zur Erlangung des akademischen Grades
Dr. rer. med.

an der medizinischen Fakultät
der Universität Leipzig

eingereicht von:
Dipl.-Inform. Med. Thomas Wendt
geboren am 08.09.1974

angefertigt an:
Institut für Medizinische Informatik, Statistik und Epidemiologie
der Universität Leipzig

Betreuer:
Prof. Dr. Alfred Winter

Beschluss über die Verleihung des Doktorgrades vom: 31.01.2006

Bibliographische Beschreibung

Thomas Wendt

Modellierung und Bewertung von Integration in Krankenhausinformationssystemen

Universität Leipzig, Dissertation

240 Seiten, 117 Literaturquellen ,
76 Abbildungen, 8 Tabellen,
2 Anlagen

Referat:

Die Dissertation behandelt die Modellierung und Bewertung der Integration von Anwendungssystemen eines Informationssystems.

Grundlage für die Modellierung ist das Metamodell $3LGM^2$, das hier zum $3LGM_A^2$ erweitert wird. Bei der Erweiterung werden Standards für Informationssystemarchitekturen berücksichtigt, um eine flexible Modellierung zu ermöglichen. Dabei wird auch eine vergleichende Übersicht über Rahmenwerke und zugehörige Standards für Informationssystemarchitekturen erarbeitet.

Aufbauend auf dem Metamodell $3LGM_A^2$ werden Bewertungsansätze für die Integration von Anwendungssystemen erarbeitet. Damit können Informationssysteme hinsichtlich der Erfüllung von Integrationsanforderungen, der Abhängigkeit von Anwendungssystemen und der Heterogenität der Integrationsinfrastruktur bewertet werden.

Für die systematische Betrachtung von Integrationsanforderungen werden Anforderungskategorien erarbeitet. Sie werden u. a. beim Entwurf eines Domänenkonzeptes verwendet, das die formale Analyse der Erfüllung von Integrationsanforderungen ermöglicht. Domänen sind dabei Mengen von Anwendungssystemen mit bestimmten Integrationsanforderungen bzw. mit einem bestimmten Integrationsstatus.

Für die Bewertung der Abhängigkeit und der Heterogenität werden verschiedene Kennzahlen, z. B. Abhängigkeitsgrade und Heterogenitätsgrade, entwickelt, die eine quantitative Bewertung der Integration ermöglichen.

Modellierung und Bewertung von Integration in Krankenhaus- informationssystemen

Thomas Wendt

Für Kathrin und Lukas

Danksagung

Die vorliegende Arbeit wurde am Institut für Medizinische Informatik, Statistik und Epidemiologie der Universität Leipzig angefertigt. Sie wurde von Herrn Professor Dr. Alfred Winter betreut, dem ich hiermit für die Überlassung des Themas sowie für die freundschaftliche und engagierte Begleitung und Unterstützung mit anregenden wissenschaftlichen Gesprächen herzlich danke.

Die Anfertigung der Arbeit wäre ohne die Unterstützung meiner Familie nicht möglich gewesen. Ich danke meiner Frau Kathrin für das Ertragen meines Rückzuges in unser Arbeitszimmer, für das ständige Motivieren zum „Dranbleiben“ und für das wiederholte sehr aufmerksame Korrekturlesen. Ich danke unserem kleinen Sohn Lukas für sein fröhliches Lachen, mit dem er mir viel Kraft geschenkt hat. Meinem Bruder Stefan danke ich für aufmerksames Korrekturlesen.

Meinem Kollegen Gert Funkat danke ich für aufmerksames Korrekturlesen trotz hoher Arbeitsbelastung und wertvolle methodische Hinweise.

Herrn Professor Dr. Wilhelm Hasselbring von der Universität Oldenburg danke ich für die Möglichkeit zum zweimaligen Vorstellen meiner Ergebnisse an seinem Institut mit anregenden Diskussionen.

Inhaltsübersicht

Inhaltsübersicht	I
Inhaltsverzeichnis	III
Kastenverzeichnis	IX
Abbildungsverzeichnis	XI
Tabellenverzeichnis	XV
A Einleitung	1
I Grundlagen	5
1 Bewertung der Integration: Ziel und Grundannahmen	7
2 Integrationsanforderungen	11
3 Informationssystemarchitekturen — Eine Einführung	21
4 Standards für Informationssystemarchitekturen	31
5 Geschäftsprozessmodellierung — Eine Einführung	77
II Architekturmodellierung	85
6 Einführung in das 3LGM ²	87
7 Flexible Architekturmodellierung — Überarbeitung des 3LGM ²	99
III Architekturbewertung	117
8 Theoretische Vorbereitung der Integrationsbewertung	119
9 Die Erfüllung von Integrationsanforderungen	129
10 Abhängigkeit von Anwendungsbausteinen	157
11 Heterogenitätsbewertung von Kommunikationsverbindungen	171
Z Abschluss	187
Anhang A Anforderungskatalog für die Informationsverarbeitung im Krankenhaus (Auszug)	i
Anhang B UML-Diagramme für die Ebenen des 3LGM ² _A	v
Literaturverzeichnis	ix
Stichwortverzeichnis	xv

Inhaltsverzeichnis

Inhaltsübersicht	I
Inhaltsverzeichnis	III
Kastenverzeichnis	IX
Abbildungsverzeichnis	XI
Tabellenverzeichnis	XV
A Einleitung	1
A.1 Gegenstand, Problematik, Motivation	1
A.1.1 Gegenstand	1
A.1.2 Problematik	2
A.1.3 Motivation	3
A.2 Zielsetzung und Fragestellung	3
A.2.1 Zielsetzung	3
A.2.2 Fragestellung	4
I Grundlagen	5
1 Bewertung der Integration: Ziel und Grundannahmen	7
1.1 Bewertungsziel	7
1.2 Grundannahmen der Architekturbewertung	8
2 Integrationsanforderungen	11
2.1 Vorbereitung	11
2.2 Kategorien von Integrationsanforderungen	13
2.2.1 Physische Integration	14
2.2.2 Datenintegration	14
2.2.3 Funktionale Integration	15
2.2.4 Semantische Integration	16
2.2.5 Kontextintegration	16
2.2.6 Präsentationsintegration	18
2.2.7 Zugriffsintegration	19
3 Informationssystemarchitekturen — Eine Einführung	21
3.1 Vorbemerkungen	21
3.2 Architektur: Komponenten und ihre Beziehungen	22

3.3	Architekturstile	24
3.4	Architekturstile für Informationssysteme — Metamodelle und Referenzmodelle	28
3.4.1	Architekturstile und Metamodelle	28
3.4.2	Architekturstile und Referenzmodelle	28
3.4.3	Bestimmung von Architekturstilen	30
4	Standards für Informationssystemarchitekturen	31
4.1	Orientierung durch Rahmenwerke	31
4.2	Technische Rahmenwerke für Informationssystemarchitekturen	32
4.2.1	Das OSI-Referenzmodell der ISO	32
4.2.2	Das Referenzmodell für offene verteilte Informationsverarbeitung der ISO	35
4.2.3	Die Object Management Architecture	40
4.2.4	Das Healthcare Information Framework	46
4.2.5	The Open Group Architectural Framework, Version 7	49
4.3	Weitere Standards für Integrationstechniken	53
4.4	Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen . .	54
4.4.1	Das Zachman-Rahmenwerk	55
4.4.2	Die Architektur integrierter Informationssysteme	56
4.4.3	Enterprise Application Integration	60
4.4.4	The Open Group Architectural Framework, Version 8	65
4.4.5	Enterprise Application Planning	68
4.5	Rahmenwerke und Architekturstile	72
4.6	Vergleichbarkeit von Architekturstandards	72
5	Geschäftsprozessmodellierung — Eine Einführung	77
5.1	Vorbemerkungen	77
5.2	Ereignisgesteuerte Prozessketten	77
5.3	Die Business Process Modeling Language	79
5.4	Das Semantische Objektmodell	81
II	Architekturmodellierung	85
6	Einführung in das 3LGM²	87
6.1	Vorbemerkungen	87
6.2	Die Ebenen des 3LGM ² und ihre Hauptklassen	87
6.2.1	Fachliche Ebene	88
6.2.2	Logische Werkzeugebene	88
6.2.3	Physische Werkzeugebene	92
6.3	Klassen für die Integrationsmodellierung	92
6.3.1	Klassen für die Modellierung der Speicherung von Informationen	92
6.3.2	Klassen für die Modellierung der Übermittlung von Informationen	94
6.4	Prozessmodellierung	95
6.5	Das 3LGM ² und Architekturstile	95

7	Flexible Architekturmodellierung — Überarbeitung des 3LGM²	99
7.1	Begriffe für die Modellierung von Komponenten auf der Basis des 3LGM ²	99
7.2	Überarbeitung des 3LGM ²	99
7.2.1	Ereignistypen	99
7.2.2	Schnittstellen und Operationen	102
7.3	Transformation von Nachrichtentypen	105
7.4	Dienste	105
7.4.1	Spezifikationen für Dienste	105
7.4.2	Modellierung von Diensten mit dem 3LGM ²	107
7.5	Kommunikationsverbindungen	107
7.6	Beispiele für die Modellierung verschiedener Architekturstile	108
7.6.1	Modellierung von HISA-basierten Architekturen	108
7.6.2	Modellierung von OMA-basierten Architekturen	110
7.6.3	Modellierung von Kommunikationsserver-basierten Architekturen	111
7.7	Einführung der Klasse Begriffssystem	113
III	Architekturbewertung	117
8	Theoretische Vorbereitung der Integrationsbewertung	119
8.1	Vorbemerkungen	119
8.2	Die Interpretation von Elementhierarchien	119
8.3	Mengen für Klassen des 3LGM _A ²	120
8.4	Prädikate zu Assoziationsbeziehungen des 3LGM _A ²	121
8.5	Zusammengesetzte Prädikate für die Unterstützung der Bewertung	122
8.5.1	Die Prädikate benötigt und unterstützt	122
8.5.2	Das Prädikat wird_angewendet_auf	123
8.5.3	Prädikate für die Analyse von Kommunikation	124
8.6	Formalisierung von Kommunikationsverbindungen	126
8.6.1	Kommunikationsverbindungen als Folgen von Kommunikationsbeziehungen	126
8.6.2	Vermittlung und Vermittlungstiefe	126
9	Die Erfüllung von Integrationsanforderungen	129
9.1	Mengen von Anwendungsbausteinen: Domänen	129
9.2	Ein Anwendungsszenario	130
9.3	Anforderungsdomeänen	133
9.3.1	Datendomeänen	133
9.3.2	Funktionale Domänen	134
9.3.3	Semantische Domänen	136
9.3.4	Kontextdomänen	136
9.3.5	Präsentationsdomänen	138
9.3.6	Zugriffsdomeänen	140
9.4	Kommunikationsdomänen	140
9.4.1	Domänen für die Übermittlung von Informationen	140
9.4.2	Domänen für den Aufruf von Funktionalität	142
9.5	Anwendung: Prüfung der Erfüllung von Integrationsanforderungen	143

9.5.1	Prüfung auf realisierte Datenintegration	143
9.5.2	Prüfung auf realisierte funktionale Integration	145
9.5.3	Prüfung auf realisierte semantische Integration	147
9.5.4	Prüfung auf realisierte Kontextintegration	150
9.5.5	Prüfung auf realisierte Präsentationsintegration	152
9.5.6	Prüfung auf realisierte Zugriffsintegration	152
10	Abhängigkeit von Anwendungsbausteinen	157
10.1	Vorbemerkungen	157
10.2	Informationale und funktionale Abhängigkeit	159
10.2.1	Der informationale Abhängigkeitsgrad	159
10.2.2	Der funktionale Abhängigkeitsgrad	159
10.3	Ausführungsabhängigkeit, transaktionale Abhängigkeit und Transaktionsstärke	160
10.3.1	Die Ausführungsabhängigkeit	160
10.3.2	Die transaktionale Abhängigkeit	161
10.3.3	Die Transaktionsstärke einer Transaktionsausführung	162
10.3.4	Kategorien von Transaktionsausführungen	162
10.4	Ein Anwendungsszenario aus dem Universitätsklinikum Leipzig	163
10.4.1	Vorstellung des Anwendungsszenarios	163
10.4.2	Kennzahlen zur Abhängigkeit für das Anwendungsszenario	167
11	Heterogenitätsbewertung von Kommunikationsverbindungen	171
11.1	Vorbemerkungen	171
11.2	Der absolute Heterogenitätsgrad von Kommunikationsverbindungen	171
11.2.1	Der absolute Heterogenitätsgrad von zwei Kommunikationsverbindungen	171
11.2.2	Der absolute Heterogenitätsgrad von n Kommunikationsverbindungen .	177
11.2.3	Interpretation des absoluten Heterogenitätsgrades	178
11.2.4	Problematik der Anwendung des Heterogenitätsgrades	178
11.3	Der relative Heterogenitätsgrad von Kommunikationsverbindungen	179
11.3.1	Der relative Heterogenitätsgrad	179
11.3.2	Interpretation des relativen Heterogenitätsgrades	180
11.4	Der kostenbewertete Heterogenitätsgrad von Kommunikationsverbindungen . .	181
11.4.1	Kosten von Kommunikationsverbindungen	181
11.4.2	Der kostenbewertete Heterogenitätsgrad	181
11.4.3	Interpretation des kostenbewerteten Heterogenitätsgrades	182
11.4.4	Bemerkungen zum kostenbewerteten Heterogenitätsgrad	183
11.5	Ein Anwendungsszenario aus dem Universitätsklinikum Leipzig	184
11.5.1	Ermittlung der Kommunikationsverbindungen als Folgen von Kommu- nikationsbeziehungen	184
11.5.2	Kennzahlen zur Heterogenität für das Anwendungsszenario	184
Z	Abschluss	187
Z.1	Vorbemerkungen	187
Z.2	Beantwortung der Fragen	187
Z.2.1	Fragen zu Ziel Z1	187
Z.2.2	Fragen zu Ziel Z2	190
Z.2.3	Fragen zu Ziel Z3	192

Z.2.4	Fragen zu Ziel Z4	193
Z.3	Diskussion	196
Z.3.1	Kommentierung der Ergebnisse	196
Z.3.2	Erfüllung der Ziele	197
Z.4	Potential für Evaluationen und für weiterführende Forschung	198
Anhang A	Anforderungskatalog für die Informationsverarbeitung im Krankenhaus (Auszug)	i
Anhang B	UML-Diagramme für die Ebenen des 3LGM_A²	v
	Literaturverzeichnis	ix
	Stichwortverzeichnis	xv

Inhaltsverzeichnis

Kastenverzeichnis

2.1	Grundbegriffe (1): Informationen, Daten, Wissen	11
2.2	Grundbegriffe (2): Informationssystem, Anwendungssystem, Integration	12
2.3	Exkurs zur semantischen Integration	17
3.1	Grundbegriffe (3): Definitionen für Architektur	21
3.2	Auszüge aus ISO/IEC 10746-2 (RM-ODP Foundations)	23
3.3	Grundbegriffe (4)	26
3.4	Grundbegriffe (5): Modelle und Referenzmodelle	29
4.1	Überblick über ISO-Standards zum RM-ODP	36
4.2	Überblick über ISO-Standards zum RM-ODP (Fortsetzung)	37
4.3	Ergänzende Standards zum RM-ODP: Spezifikation der in ISO/IEC 10746-3 vorgesehenen Trading Function in Standard ISO/IEC 13235-1 (Auszüge)	39
4.4	Überblick über OMG-Standards zur Architekturentwicklung	45
4.5	Beispiele für Methoden zur Datensicht und zur Steuerungssicht der ARIS	58
4.6	Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien	70
4.7	Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien (Fortsetzung 1)	71
4.8	Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien (Fortsetzung 2)	72
7.1	Das Weglassen der Konfigurationen im $3LGM_A^2$	104
9.1	Ereignistypen und Kommunikationsdomänen	143
11.1	Exkurs: Ausrichtung von Symbolsequenzen	172

Abbildungsverzeichnis

1.1	Dimensionen der Qualität von Informationssystemen (Quelle: [STYLIANOU and KUMAR 2000], S. 100, Abb. 1, S. 103, Tab. 1)	8
2.1	IT-Infrastruktur-Integrationsprofile des IHE IT Infrastructure Technical Framework, Vol. 1 (Quelle: [HIMSS/RSNA 2003a], Figure 2-1, S. 10)	13
2.2	Drei parallel genutzte Anwendungssysteme: ein Stationsmanagementsystem, ein OP-Dokumentationssystem und der Intranet-Server des Zentrallabors (Quelle: Bildschirmfoto von einem PC am UKL)	18
3.1	Ein Komponententyp (a), ein Konnektorentyp (b) und eine Konfiguration (c) (Quelle [ABOWD et al. 1995], S. 329-331, Abb. 3 - Abb. 5)	25
4.1	OSI-Referenzmodell (Quelle: [ISO/IEC JTC 1 1996c], S. 28)	32
4.2	Übersichtsgrafik zum generischen Schichtenkonzept (a) und Kommunikation von Einheiten der Schicht $N + 1$ über die Schicht N (b) (Quelle [ISO/IEC JTC 1 1996c], S. 7-8, Figure 3 und Figure 4)	34
4.3	OSI- und TCP/IP-Schichtenmodelle (nach [WASHBURN and EVANS 1994], S. 9, 157) . .	34
4.4	Das Object Management Architecture Reference Model (Quelle: [OMG 1997], Abb. 4-1)	41
4.5	Das Model Driven Architecture Pattern (a) (Quelle: [OMG 2003b], Abb. 2-2) und generische Unterscheidung von Sichtweisen auf Plattformen (b) (Quelle: [OMG 2003b], Abb. 6-6)	44
4.6	Die Schichten der Architektur von Informationssystemen im Gesundheitswesen (Quelle: [CEN TC251 1997], Abbildung 1)	47
4.7	Informationsmodell für die Subject of Care Healthcare Common Services (a) und funktionale Spezifikation (b) (Quelle: [CEN TC251 1997], S. 15)	48
4.8	TOGAF und seine Beziehungen zu anderen US-amerikanischen Rahmenwerken (Quelle: unbekannt)	49
4.9	Die Phasen der TOGAF Architecture Development Method und die Schritte der Phase C (Quelle: [THE OPEN GROUP 2001], Part II, Introduction, Figure 2)	51
4.10	Einfache (a) und detailliertere (b) Grafik zum TOGAF TRM (Quellen: [THE OPEN GROUP 2001], Part III, High-level Breakdown, Figure 1 und Part III, TRM in Detail, Figure 1)	52
4.11	Das Zachman-Rahmenwerk für Informationssystemarchitektur (Quelle: [ZACHMAN 1999])	55
4.12	Das ARIS-Haus (Quelle: [SCHEER 1998b], S. 46, Abbildung 20)	57
4.13	Metamodell der Fachkonzeptebene der Funktionssicht (Ausschnitt, Quelle: [SCHEER 1998a], S. 38, Abbildung 33)	57
4.14	Das ARIS-Phasenmodell (Quelle: [SCHEER 1998b], S. 39, Abbildung 16)	59
4.15	Vertikale Fragmentierung von Informationssystemen und horizontale Integration (Quelle: [HASSELBRING 2000], S. 34-35, Figure 1 und Figure 2)	60
4.16	Unterscheidung von Integrationsebenen in [HASSELBRING 2000] (a) sowie von EAI-Ebenen in [LINTHICUM 2000b] (b) und in [LONGO 2001] (c)	61
4.17	Einfaches Architektur-Referenzmodell für EAI (Quelle: [SCHMIDT 2002])	63
4.18	EAI-Architekturmuster (Quelle: [LUTZ 2000])	64
4.19	Die Phasen der TOGAF Architecture Development Method und die Schritte der Phase D (Quelle: [THE OPEN GROUP 2003], Part II, Introduction, Figure 2)	65

4.20	Architekturkontinuum (a) und Lösungskontinuum (b) in TOGAF, Version 8 (Quelle: [THE OPEN GROUP 2003], Part III, Enterprise Continuum in Detail, Architecture Continuum, Figure 1 und Solutions Continuum, Figure 3)	66
4.21	TOGAF TRM in TOGAF, Version 8, (a und b) und Ableitung der Elemente des TOGAF III-RM aus dem TOGAF TRM (c und d) (Quellen: [THE OPEN GROUP 2003], Part III, Foundation Architecture: Technical Reference Model, High-level Breakdown, Figure 1, und TRM in Detail, Figure 1, [THE OPEN GROUP 2003], Part III, Integrated Information Infrastructure Reference Model, High-Level View, Figure 2b und Figure 3)	67
4.22	Phasen des Enterprise Architecture Planning (Quelle: [SPEWAK and HILL 1992], S. 13, Figure 1.7)	69
5.1	Beispiel für eine EPK (Quelle: [LANGNER et al. 1997], S. 481, Abb. 1)	78
5.2	Beispiel für eine Prozessbeschreibung mit der BPML; die mit „wsdl“ beginnenden Definitionen sind WSDL-Definitionen für Dienste (Quelle: [BPMI BPML WG 2002], S. 67-70, Beispiel 5)	80
5.3	Beispiel für eine mit der BPMN erstellte Prozessbeschreibung (Quelle: [BPMI NOTATION WG 2004], S. 139, Abb. 87)	81
5.4	Ebenen der Unternehmensarchitektur im SOM-Ansatz (Quelle: [FERSTL and SINZ 1995], S. 212, Abb. 2)	82
5.5	Metamodell für die Geschäftsprozessmodellierung im SOM-Ansatz (Quelle: [FERSTL and SINZ 1995], S. 216, Abb. 7)	83
5.6	Beispiel für ein Interaktionsmodell (a) und ein Vorgangs-Ereignis-Modell (b) (Quelle: [FERSTL and SINZ 1995], S. 218-219, Abb. 10 u. 12)	84
6.1	Spezifikationen der fachlichen Ebene des 3LGM ² mit der UML (vgl. [WINTER et al. 2003], S. 546, Abb. 1) (a) und Auszug der fachliche Ebene des Informationssystems des UKL (b)	89
6.2	Spezifikationen der logischen Werkzeugebene des 3LGM ² mit der UML (vgl. [WINTER et al. 2003], S. 547, Abb. 3) (a) und Auszug der logischen Werkzeugebene des Informationssystems des UKL (b)	90
6.3	Spezifikationen der physischen Werkzeugebene des 3LGM ² mit der UML (vgl. [WINTER et al. 2003], S. 548, Abb. 5) (a) und Auszug der physischen Werkzeugebene des Informationssystems des UKL (b)	91
6.4	Auszug des Informationssystems des UKL in Drei-Ebenen-Darstellung	93
6.5	Kombinationen von Ereignistypen und Nachrichtentypen zur Beschreibung der Kommunikation zwischen den Anwendungsbausteinen Patientenverwaltungssystem und Kommunikationsserver am UKL	95
6.6	Ein Kommunikationsprozess zur Übermittlung von ADT-Informationen im Informationssystem des UKL	96
7.1	Fachliche Ebene des 3LGM _A ² ; hervorgehoben sind die überarbeiteten Klassen.	101
7.2	Logische Werkzeugebene des 3LGM _A ² ; hervorgehoben sind die neu eingeführte Klasse Operation und die überarbeiteten Assoziationsbeziehungen.	103
7.3	Eine Kommunikationsverbindung zur Übertragung von Falldaten im Informationssystem des UKL	108
7.4	Eine einfache OMA-basierte Architektur	109
7.5	Eine einfache OMA-basierte Architektur	111
7.6	Eine einfache Kommunikationsserverarchitektur	112
7.7	Erweiterte fachliche Ebene des 3LGM _A ² ; hervorgehoben sind die neu eingeführte Klasse Begriffssystem und die zuvor in Abschnitt 7.2.1 überarbeiteten Klassen.	113

7.8	Beispiele für die Zuordnung von Begriffssystemen zu greift_zu_auf-beziehungen: Bei der Interpretation und Bearbeitung von Diagnosen und Prozeduren werden die Diagnosenklassifikation <i>ICD10</i> , die Prozedurenklassifikation <i>OPS301</i> und das Abrechnungssystem <i>DRG</i> angewendet.	114
7.9	Erweiterte logische Werkzeugebene des $3LGM_A^2$ mit den Beziehungen der Klasse Begriffssystem zu den Klassen der logischen Werkzeugebene	115
7.10	Physische Werkzeugebene des $3LGM_A^2$	116
8.1	Interpretation von ist_Teil_von-Beziehungen	120
8.2	Beispiel für eine Kommunikationsverbindung mit Vermittlungsbeziehungen	127
9.1	Auszug aus der fachlichen Ebene des Informationssystems des UKL	130
9.2	Auszüge aus der logischen Werkzeugebene des Informationssystems des UKL	132
9.3	Datendomäne des Objekttyps <i>Fall</i>	134
9.4	Funktionale Domäne der Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i>	135
9.5	Zuordnung der Begriffssysteme <i>DRG</i> , <i>ICD10</i> und <i>OPS301</i> zu greift_zu_auf-Beziehungen (a) und semantische Domänen der Begriffssysteme (b)	137
9.6	Kontextdomäne des Objekttyps <i>Fall</i> und des physischen Datenverarbeitungsbausteines <i>PC NCH 15</i>	138
9.7	Präsentationsdomäne des physischen Datenverarbeitungsbausteines <i>PC NCH 15</i>	139
9.8	Übermittlungsdomäne des Objekttyps <i>Fall</i> , des Ereignistyps <i>NP1110</i> und des Anwendungsbausteines <i>Patientenverwaltungssystem</i>	141
9.9	Aufrufdomäne der Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i> , des Ereignistyps <i>DIAPROZ_init</i> und des Anwendungsbausteines <i>Verschlüsselungssystem</i>	142
9.10	Vergleich der Datendomäne des Objekttyps <i>Fall</i> und des Ereignistyps <i>NP1110</i> (a) mit der Übermittlungsdomäne des Objekttyps <i>Fall</i> , des Ereignistyps <i>NP1110</i> und des Anwendungsbausteines <i>Patientenverwaltungssystem</i> (b)	144
9.11	Vergleich der funktionalen Domäne der Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i> (a) mit der Aufrufdomäne der Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i> , des Ereignistyps <i>DIAPROZ_init</i> und des Anwendungsbausteines <i>Verschlüsselungssystem</i> (b)	146
9.12	Vergleich der semantischen Domäne des Begriffssystems <i>ICD10</i> (a) mit der Übermittlungsdomäne desselben Begriffssystems, des Ereignistyps <i>ICD_neu</i> und des Anwendungsbausteines <i>Verschlüsselungssystem</i> (b)	148
9.13	Vergleich der semantischen Domäne des Begriffssystems <i>ICD10</i> (a) mit der Aufrufdomäne der Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i> , des Ereignistyps <i>DIAPROZ_init</i> und des Anwendungsbausteines <i>Verschlüsselungssystem</i> (b)	149
9.14	Vergleich der Kontextdomäne des Objekttyps <i>Fall</i> , des physischen Datenverarbeitungsbausteines <i>PC NCH 15</i> und des Ereignistyps <i>Kontextwechsel_Fall</i> (a) mit der Übermittlungsdomäne des Objekttyps <i>Fall</i> , des Ereignistyps <i>Kontextwechsel_Fall</i> und des Anwendungsbausteines <i>KDMS-IS-H*MED</i> (b)	151
9.15	Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines <i>PC NCH 15</i> (a) mit der Übermittlungsdomäne des Objekttyps <i>Mitarbeiter des Pflegedienstes</i> , des Ereignistyps <i>ZUGR_update</i> und des Anwendungsbausteines <i>R/3-basierte Anwendungen</i> (b)	153
9.16	Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines <i>PC NCH 15</i> (a) mit der Aufrufdomäne der Aufgabe <i>Rechteprüfung MRZ</i> , des Ereignistyps <i>LOGIN_BBS</i> und des Anwendungsbausteines <i>LDAP-Benutzerverzeichnissystem</i> (b)	155
10.1	Lösungsspektrum zur Unterstützung heterogener Datenbanken (Quelle: [RAHM 1994], Abschnitt 10.4)	158

10.2	Alte Architektur im Anwendungsszenario aus dem UKL; hervorgehoben sind die Anwendungsbausteine der Datendomäne des Objekttyps <i>Fall</i> , des Ereignistyps <i>NP11I0</i> und des Anwendungsbausteines <i>Patientenverwaltungssystem</i>	164
10.3	Geplante Architektur im Anwendungsszenario aus dem UKL; hervorgehoben sind die Anwendungsbausteine der Datendomäne des Objekttyps <i>Fall</i> , des Ereignistyps <i>NP11I0</i> und Anwendungsbausteines <i>Patientenverwaltungssystem</i>	165
10.4	Anwendungsszenario aus dem UKL mit Kommunikationsserver (a) und mit ORB (b); hervorgehoben sind die Kommunikationsverbindungen zur Verteilung der Falldaten . .	166
11.1	Ausrichtung von zwei Kommunikationsverbindungen	174
11.2	Berechnung des Gleichheitswertes p von zwei ausgerichteten Kommunikationsverbindungen mit Vermittlungsverbindungen	175
11.3	Berechnung des Gleichheitswertes p von drei Kommunikationsverbindungen	177
11.4	Anwendungsszenario aus dem UKL mit Kommunikationsserver (a) und mit ORB (b); hervorgehoben sind die Kommunikationsverbindungen zur Verteilung der Falldaten . .	185
11.5	Ausgerichtete Kommunikationsverbindungen des Anwendungsszenarios; Berechnung der absoluten und der relativen Gleichheitswerte	186

Tabellenverzeichnis

3.1 Beispiele für Architekturstile (zusammengestellt aus [SHAW and GARLAN 1996], Abschnitte 2.2-2.5, S. 21-25)	28
4.1 Zuordnung von RM-ODP-Sichtweisen zu MDA-Sichtweisen (Quelle: [OMG 2003b], S. 3-1 - 3-2)	43
4.2 Übersicht über Rahmenwerke zu Informationssystemarchitekturen (Teil 1)	74
4.3 Übersicht über Rahmenwerke zu Informationssystemarchitekturen (Teil 2)	75
6.1 Beispiele für Ereignistypen im Kommunikationsstandard HL7 (Quelle: [HL7 1999]) . .	94
7.1 Beispiele für äquivalente Ereignistypen in den Kommunikationsstandards HL7 und SAP-HCM (vgl. Tabelle 6.1; Quellen: [HL7 1999] und [SAP 2005])	102
9.1 Elemente des Beispielmodells des Informationssystems des UKL	131
9.2 Elemente des Beispielmodells des Informationssystems des UKL (Fortsetzung 2)	133

A Einleitung

In vielen Krankenhäusern werden zur Zeit in großer Breite rechnerunterstützte Anwendungssysteme eingeführt. Sie dienen u. a. der Verschlüsselung von Diagnosen und Prozeduren, der OP-Dokumentation, der Arztbriefschreibung oder dem Erstellen und Verteilen von Befunden und Bildern.

Die damit entstehende nahezu flächendeckende Rechnerausstattung lässt in vielen medizinischen Abteilungen eines Krankenhauses Forderungen nach elektronischer Verfügbarkeit von Informationen anderer Abteilungen entstehen. Einzelne rechnerunterstützte Insellösungen sollen also miteinander verknüpft werden. Beispielsweise fordern Stationsärzte und Pflegekräfte Möglichkeiten der Einsicht in Laborbefunde von Rechnern auf der Station aus. Dazu muss der Zugriff auf das Befundungssystem des Labors bereitgestellt werden, oder die Laborbefunde müssen automatisch in ein auf den Stationsrechnern verfügbares Anwendungssystem übertragen werden.

Verwaltungsabteilungen, wie die Patientenverwaltung oder das Controlling, fordern die elektronische Verfügbarkeit von Daten, z. B. zur Weiterverarbeitung bei der Erstellung von Rechnungen und Statistiken. So werden seit dem Jahr 2000 große Anstrengungen im Zusammenhang mit der elektronischen Übermittlung von Diagnose- und Prozedurdaten für die Patientenabrechnung und für die Vorbereitung der Einführung der Diagnosis Related Groups (DRGs) unternommen.

Durch die Verfügbarkeit abteilungsspezifischer Anwendungssysteme unterschiedlicher Hersteller sind die Informationsmanager von Krankenhäusern oft einer Vielzahl von Informationssystemkomponenten gegenübergestellt. Diese unterscheiden sich hinsichtlich der zugrunde liegenden Implementierungskonzepte, der verwendeten Datenstrukturen oder der Gestaltung der Benutzungsschnittstellen. Trotzdem sollen sie aber möglichst ohne zusätzliche Handarbeit miteinander kommunizieren, um den Anwendern die Belastung des Abschreibens, der mehrfachen Anmeldung usw. zu ersparen.

A.1 Gegenstand, Problematik, Motivation

A.1.1 Gegenstand

Hauptgegenstand der Arbeit sind Krankenhausinformationssysteme. Die in dieser Arbeit entwickelten Lösungsansätze sind nicht grundsätzlich auf Krankenhausinformationssysteme beschränkt, aber durch die Arbeit mit und in diesen Systemen geprägt.

Das Management komplexer Informationssysteme erfordert umfangreiche Kenntnisse über die Informationssystemkomponenten und ihre Integration. Diese Arbeit entwickelt eine Modellierungsvorschrift für die Integration. Mit der Vorschrift soll es gelingen, *Integrationsanforderungen* verschiedener Kategorien, z. B.

- Datenintegration,
- funktionale Integration oder

A Einleitung

- Kontextintegration,

verschiedene *Integrationstechniken*, z. B.

- nachrichtenbasierte Kommunikation,
- Remote Function Call,
- Object Request Broker oder
- Kontextsynchronisierung,

unter Berücksichtigung verschiedener durch *Architekturstandards* vorgegebener *Architekturstile*, z. B.

- Store-&-Forward-Architektur,
- Object Management Architecture oder
- Healthcare Information Systems Architecture

formal zu beschreiben und leicht verständlich in Texten oder Grafiken darzustellen.

Die Modellierungsvorschrift wird nicht unabhängig von anderen Ansätzen entworfen, sondern als Erweiterung eines vorhandenen Metamodells, des 3LGM² ([WINTER et al. 2003]) zum 3LGM_A². Das 3LGM² definiert drei Ebenen mit verschiedenen Modellelementen, auf deren Basis Referenzmodelle und Modelle bestehender Informationssysteme erstellt werden können.

Auf der Basis der Modellierungsvorschrift werden Ansätze für die Bewertung der Integration von Anwendungssystemen entwickelt. Mit diesen Ansätzen soll ermöglicht werden

- die Erfüllung von Integrationsanforderungen formal zu überprüfen und
- die Architektur des Informationssystems hinsichtlich der Abhängigkeit von Anwendungssystemen sowie der Heterogenität der eingesetzten Integrationstechniken quantitativ zu bewerten.

A.1.2 Problematik

Kenntnisse über die Integration von Informationssystemkomponenten sind oft nicht oder nicht zusammenhängend dokumentiert. Dadurch ist u. a. der Einarbeitungsaufwand bei Mitarbeiterwechseln sehr hoch, oder bei Problemen müssen Informationen über die Integration aufwendig zusammengestellt werden.

Das Informationsmanagement wird weiterhin dadurch erschwert, dass

- Dokumentationen von Informationssystemkomponenten von den Herstellern bzw. Lieferanten auf spezielle, hauseigene Weise erstellt werden,
- Anforderungen an Informationssystemkomponenten von den Kunden, d. h. den Krankenhäusern, auf spezielle, hauseigene Weise erstellt werden und
- Dokumentationen moderner Integrationstechniken und -standards durch die herausgebenden Organisationen auf spezielle Weise erstellt werden.

Dadurch ist es schwer, die Vor- und Nachteile der Anwendung bestimmter Techniken und Standards für die Integration einzuschätzen und Entscheidungen für oder gegen die Einführung oder Ablösung von Komponenten verschiedener Hersteller zu treffen.

Diese Arbeit soll zur Lösung folgender Probleme beitragen:

Problem P1 Ein Vergleich von Architekturstandards, die verschiedenen Integrationstechniken zugrunde liegen, ist ohne ausführliche Auseinandersetzung mit diesen

Standards kaum möglich.

Problem P2 Für das Dokumentieren bzw. das Modellieren von Integrationsanforderungen und Integrationstechniken werden oft unterschiedliche Methoden und Vorgehensweisen gewählt. Dadurch können die betreffenden Dokumente bzw. Modelle nur schwer verglichen werden.

Problem P3 Auswahlentscheidungen für oder gegen bestimmte Integrationstechniken können oft nicht oder nur schwer hinsichtlich der Erfüllung von Integrationsanforderungen oder der Beeinflussung der Integrationsinfrastruktur getroffen werden.

Problem P4 Mit vorhandenen Werkzeugen zur Informationssystemmodellierung können Integrationsanforderungen und ihre Erfüllung nur schwer modelliert und bewertet werden.

A.1.3 Motivation

Am Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE) der Universität Leipzig wird das in Abschnitt A.1 genannte Metamodell 3LGM² weiterentwickelt. Wesentliche Fragen der Weiterentwicklung des Modells betreffen die Integration von Informationssystemkomponenten. Diese Arbeit soll zu dieser Weiterentwicklung beitragen.

Gleichzeitig ist der Autor dieser Arbeit in Projekten zur Integration von Anwendungssystemen am Universitätsklinikum Leipzig (UKL) tätig. Innerhalb der Projekte treten im Zusammenhang mit Integrationsfragestellungen immer wieder Probleme auf, die sich unter den in Abschnitt A.1.2 genannten Problemen zusammenfassen lassen. Eine Unterstützung der Analyse und Planung der Integrationsinfrastruktur könnte die Durchführung von Integrationsprojekten am UKL und an anderen Klinika erleichtern. Die in den Projekten gesammelten Erfahrungen können bei der Entwicklung von Lösungsansätzen helfen.

A.2 Zielsetzung und Fragestellung

A.2.1 Zielsetzung

Diese Arbeit hat die folgenden Ziele:

Ziel Z1 (zu P1 - P3) Es soll dargestellt werden, welche Anforderungen bzgl. der Integration Integrationsanforderung von Informationssystemkomponenten typischerweise in Krankenhäusern gestellt werden und welche Techniken zu ihrer Erfüllung verwendet werden können.

Ziel Z2 (zu P1) Es soll eine vergleichende Übersicht über Architekturstandards erstellt werden.

Ziel Z3 (zu P2) Es soll eine Überarbeitung des 3LGM² entworfen werden, welche die Modellierung der Erfüllung von Integrationsanforderungen und die Bewertung der Verwendung unterschiedlicher Integrationstechniken erlaubt. Dabei sollen vorhandene Architekturstandards berücksichtigt werden.

Ziel Z4 (zu P3) Es soll mit Hilfe des überarbeiteten 3LGM² untersucht werden, ob verschiedene Integrationstechniken ausgetauscht werden können, ohne dass dabei Einbußen an Funktionalität entstehen.

Zu Problem P4 ist kein spezielles Ziel angegeben, da die Werkzeugentwicklung in dieser Arbeit nicht behandelt wird. Mit dem in [WENDT et al. 2004] beschriebenen Werkzeug 3LGM²-Baukasten können 3LGM²-konforme Modelle erstellt und analysiert werden können. Es kann als Grundlage für ein Modellierungswerkzeug dienen, das der in dieser Arbeit vorgenommenen Überarbeitung des 3LGM² entspricht und die Anwendung der ebenfalls in dieser Arbeit entwickelten Bewertungsansätze ermöglicht. Die Ergebnisse dieser Arbeit können als Spezifikation für die Werkzeug(weiter)entwicklung verwendet werden.

A.2.2 Fragestellung

Um die Ziele aus Abschnitt A.2.1 zu erfüllen, sollen u. a. die folgenden Fragen beantwortet werden:

1. Fragen zu Ziel Z1:
 - F1.1** Welche Anforderungen werden an die Integration von Informationssystemkomponenten gestellt?
 - F1.2** Welche Techniken können zur Erfüllung der Integrationsanforderungen verwendet werden und welche Vor- bzw. Nachteile haben sie?
2. Fragen zu Ziel Z2:
 - F2.1** Welche bekannten Architekturstandards gibt es und welche Charakteristika haben sie?
 - F2.2** Wie stehen die Integrationsanforderungen mit den Architekturstandards in Beziehung?
 - F2.3** Welchen Integrationstechniken liegen die Architekturstandards zugrunde?
3. Fragen zu Ziel Z3:
 - F3.1** Wie stehen die in Frage F2.1 genannten Architekturstandards mit dem 3LGM² in Beziehung?
 - F3.2** Wie muss das 3LGM² überarbeitet werden, um die in Frage F1.1 genannten Integrationsanforderungen modellieren zu können?
 - F3.3** Wie muss das 3LGM² überarbeitet werden, um die in Frage F1.2 genannten Integrationstechniken modellieren sowie Vor- und Nachteile bestimmen zu können?
4. Fragen zu Ziel Z4:
 - F4.1** Welche Gemeinsamkeiten und Unterschiede bestehen zwischen den in Frage F2.1 genannten Architekturstandards?
 - F4.2** Wie können die in Frage F2.3 genannten Integrationstechniken ausgetauscht werden und welche Einbußen oder Gewinne an Funktionalität entstehen dabei?

Teil I
Grundlagen

1 Bewertung der Integration: Ziel und Grundannahmen

Die Bewertung des Einsatzes bestimmter Integrationstechniken nimmt, nach ausführlicher Vorbereitung, einen großen Teil der vorliegenden Arbeit ein. Die im Abschnitt A.2.2 der Einleitung genannten Fragen, insbesondere die Fragen F1.2, F3.3 und F4.2, führen zur Erörterung der Bewertungsthematik. Bevor in den weiteren Kapiteln von Teil I und im Teil II Antworten zu den übrigen Fragen und wesentliche Grundlagen und Vorbereitungen für die Bewertungsansätze von Teil III erarbeitet werden, beschreibt dieses Kapitel das Bewertungsziel und Grundannahmen für die Bewertung. Diese Ausführungen leiten die weiteren Kapitel dieser Arbeit.

1.1 Bewertungsziel

Für die Bewertung von Informationssystemen oder einzelnen Komponenten werden oft betriebswirtschaftliche Kriterien formuliert. Als Ziel von Evaluationen wird z. B. der Kenntnisgewinn über die Änderungen im Kosten-Nutzen-Verhältnis genannt. Kosten werden dann z. B. über Gerätekosten, Materialverbräuche, Behandlungszeiten, Kosten von Nebenwirkungsbehandlungen usw. quantifiziert.

Auch andere Evaluationsziele, die die Behandlungsqualität und die Zufriedenheit von Patienten, aber auch die Zufriedenheit des medizinischen Personals in den Mittelpunkt stellen, werden vermehrt gesetzt. Zugehörige Bewertungskriterien umfassen u. a. quantitative Elemente wie Wartezeiten, Längen von Krankenhausaufenthalten, das Auftreten von Komplikationen oder Nebenwirkungen, Zeiten für die Bereitstellung von Akten, aber auch qualitative Elemente wie die subjektive Zufriedenheit von Patienten mit der Behandlung oder die subjektive Bewertung der Benutzbarkeit von Anwendungssystemen durch Personal.

Die genannten Evaluationsziele und -kriterien sollen hier nicht in Frage gestellt werden. Der Schwerpunkt dieser Arbeit liegt jedoch auf einem weiteren, in der Literatur bisher kaum diskutierten Evaluations- bzw. Bewertungsziel: der **Bewertung der Architekturqualität eines Informationssystems**.

Da die Architektur unmittelbar mit der Integration von Informationssystemkomponenten zusammenhängt, kann man auch von *Bewertung der Integrationsqualität* sprechen.

Informationssystembewertung für das Informationsmanagement

Über die bereits genannten Evaluationskriterien kann die Architektur eines Informationssystems *mittelbar* bewertet werden, u. a. hinsichtlich der Erfüllung der Nutzeranforderungen. Bisher sind jedoch für das Informationsmanagement keine geeigneten *unmittelbaren formalen*

Ziel

Bewertung der Architekturqualität (=Integrationsqualität) eines Informationssystems

Grundannahmen der Architekturbewertung

GA1: Bei der Bewertung der Architektur stehen folgende Fragen im Vordergrund:

- Welche Integrationsanforderungen bestehen?
- Sind die Integrationsanforderungen erfüllt?
- Wie groß ist die Abhängigkeit zwischen Anwendungssystemen?
- Wie komplex ist die Architektur des Informationssystems?

GA2: Mit der Anzahl unterschiedlicher Integrations-techniken steigt die Komplexität und sinkt die Beherrschbarkeit eines Informationssystems.

1 Bewertung der Integration: Ziel und Grundannahmen

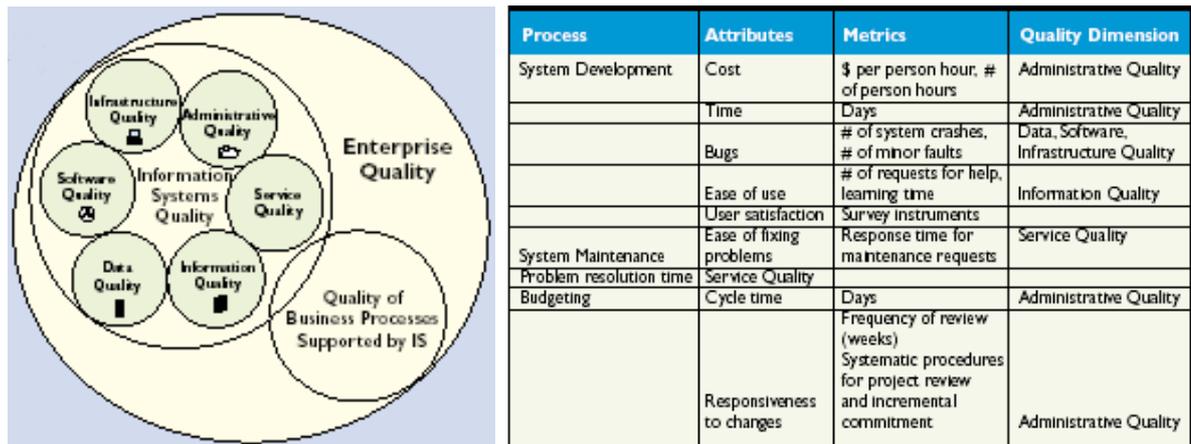


Abbildung 1.1: Dimensionen der Qualität von Informationssystemen (Quelle: [STYLIANOU and KUMAR 2000], S. 100, Abb. 1, S. 103, Tab. 1)

Bewertungsansätze für die Erfüllung von Nutzeranforderungen und für die Bewertung der Komplexität und Wartbarkeit von Informationssystemen bekannt.

Die im Teil III dieser Arbeit entwickelten Bewertungsansätze wurden für das Management von Informationssystemen entworfen. Sie sollen die strategische Bewertung und Planung von Informationssystemen unterstützen, z. B. bei der Ermittlung unnötiger Komplexität oder beim Vergleich von alternativen Lösungsansätzen. Spezielle technisch orientierte Bewertungsansätze, z. B. Quality of Service, werden nicht behandelt.

1.2 Grundannahmen der Architekturbewertung

Ähnlich mathematischen Axiomen, nur weniger formal, werden hier für die Architekturbewertung Grundannahmen getroffen, auf deren Basis die Bewertungsansätze der Kapitel 9 bis 11 entwickelt werden:

GA1: Bei der Bewertung der Architektur stehen folgende Fragen im Vordergrund:

- Welche Integrationsanforderungen bestehen?
- Sind die Integrationsanforderungen erfüllt?
- Wie groß ist die Abhängigkeit zwischen Anwendungssystemen und wo besteht Potential zur Verringerung der Abhängigkeit?
- Wie komplex ist die Architektur des Informationssystems und wo besteht Vereinfachungspotential?

GA2: Mit der Anzahl unterschiedlicher verwendeter Integrationstechniken steigt die Komplexität und sinkt die Beherrschbarkeit eines Informationssystems. Das gilt insbesondere dann, wenn mehrere Integrationstechniken zur Erfüllung derselben Integrationsanforderung(en) angewendet werden.

Von den zur Grundannahme GA1 genannten Fragen, die bei der Architekturbewertung im Vordergrund stehen, werden die ersten beiden hier ohne weitere Kommentierung als grundlegende Fragen angenommen.

Mit den beiden anderen Fragen wird unterstellt, dass die Komplexität des Informationssystems im Mittelpunkt der unmittelbaren formalen Bewertung stehen sollte. Die Auswahl stützt sich auf die immer wieder in der Informationsmanagementpraxis auftretende Forderung nach einfachen Architekturen und Wiederverwendung von Integrationstechniken. In Arbeiten wie [CLEGG 2000] und [HAYNE and POLLARD 2000] wurden verschiedene Kriterien für gutes Management, insbes. auch das Systemdesign, herausgearbeitet. Darin finden sich u. a. Forderungen nach Vereinfachung:

„Principle 11: Systems should be simple and make problems visible.“ ([CLEGG 2000], S. 471)

„A systems analyst at a large public transit company summarized as follows: 'We have a really mixed bag of development tools at the moment. We go everywhere from COBOL and FORTRAN all the way up to 4GL's such as PowerHouse and FoxPro. So we are trying to support too many different types of applications and too many different languages.'“ ([HAYNE and POLLARD 2000], S. 78)

Beide zitierte Werke nennen auch andere Forderungen bzw. Kriterien. Viele davon beziehen sich auf die Gestaltung von Prozessen im Informationsmanagement, z. B. auf die Berücksichtigung sozialer Aspekte, das Projektmanagement oder die strategische Ausrichtung. Sie sind ebenso wichtig, sollen hier aber nicht behandelt werden. Hier steht das Ergebnis von Informationsmanagement-Projekten, d. h. der Zustand des Informationssystems, im Vordergrund, genauer: die Qualität der Architektur bzw. der Integration.

In [STYLIANOU and KUMAR 2000] werden verschiedene Dimensionen bzw. Aspekte der Qualität beschrieben. Bezogen auf diese Qualitätsdimensionen behandelt die vorliegende Arbeit die administrative Qualität (administrative quality) und die Betreuungsqualität (service quality) (Abbildung 1.1).

Das 3LGM² als Beschreibungssprache für Informationssysteme

Für die Bewertung von Informationssystemen wird eine Erweiterung des Metamodells 3LGM², das in Kapitel 7 erarbeitete 3LGM_A², als Beschreibungssprache zugrunde gelegt. Informationen über Informationssysteme, die für die in dieser Arbeit erarbeiteten Bewertungsansätze benötigt werden, können mit dem 3LGM_A² ausgedrückt werden. Aus diesen „primären“ Informationen können weitere Informationen zur Bewertung abgeleitet werden. Letzteres wird in den Kapiteln von Teil III ausführlich beschrieben.

2 Integrationsanforderungen

2.1 Vorbereitung

Die in diesem Kapitel vorgestellten Kategorien von Integrationsanforderungen sind wichtige Grundlage für die Bewertung der Integration von Informationssystemkomponenten hinsichtlich der Erfüllung von Integrationsanforderungen. Sie werden in vielen Kapiteln dieser Arbeit, hauptsächlich jedoch im Kapitel 9, verwendet. Dort werden die Modellierung von Integrationsanforderungen und die Prüfung der Erfüllung der Anforderungen behandelt.

Integrationsanforderungen werden in dieser Arbeit als Forderungen an die Leistungsfähigkeit bzw. Funktionalität aus Anwendersicht formuliert. Diese Anforderungen sind von technisch formulierten Anforderungen zu unterscheiden, wie etwa Skalierbarkeit, Offenheit oder Fehler-toleranz (siehe z. B. [EMMERICH 2003], S. 20-26).

Ein einfaches Beispiel für Integrationsanforderungen ist der Wunsch, bei parallel genutzten Anwendungssystemen Benutzerkennung und Passwort, sofern erforderlich, nur einmal eingeben zu müssen und nicht getrennt für jedes der Anwendungssysteme. Eine weiteres Beispiel ist der Wunsch nach automatischer Verfügbarkeit von Befunden aus einem zentralen Klinikumslabor oder einer radiologischen Abteilung auf einem Stationsrechner. Ein drittes Beispiel ist der Wunsch nach automatischer Verfügbarkeit von bei der Patientenaufnahme erfassten Patienten- und Falldaten im Laborinformationssystem.

Kasten 2.1: Grundbegriffe (1): Informationen, Daten, Wissen

Die drei Bezeichnungen *Information*, *Daten* und *Wissen* haben ihren festen Platz in der Umgangssprache und es gibt persönliche Vorstellungen davon, für welche Begriffe sie stehen. In [LEINER et al. 1997], S. 22 werden sie wie folgt definiert (siehe auch DIN 44300):

„*Information* ist die Kenntnis über bestimmte Sachverhalte oder Vorgänge.“

„*Daten* sind Gebilde aus Zeichen oder kontinuierliche Funktionen (z. B. Tonsignale), die aufgrund bekannter oder unterstellter Abmachungen Informationen darstellen. Daten sind die Grundlage oder das Ergebnis eines Verarbeitungsschrittes.“

„*Wissen* ist die Kenntnis über den in einem Fachgebiet zu gegebener Zeit vorhandenen Konsens hinsichtlich Terminologie, regelhafter Zusammenhänge und Handlungsrichtlinien. Wissen ist demnach auch *Information* im weiteren Sinne.“

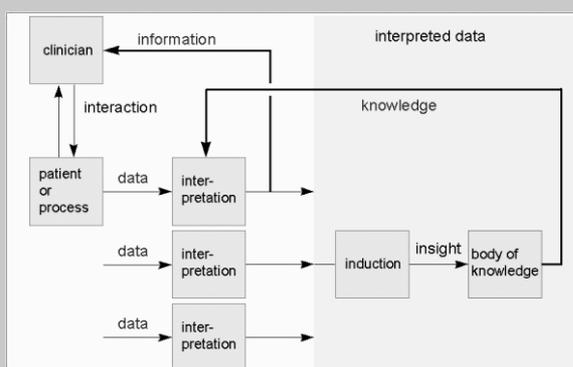


Abbildung: Daten, Informationen und Wissen in der Patientenversorgung (Quelle: [VAN BEMMEL and MUSEN 1997], Figure 1.1)

Kasten 2.2: Grundbegriffe (2): Informationssystem, Anwendungssystem, Integration

Ein *Informationssystem* sei nach [HAUX et al. 1998], S. 18

„das (...) Teilsystem eines Unternehmens, das aus den *informationsverarbeitenden* Aktivitäten und den an ihnen beteiligten menschlichen und maschinellen Handlungsträgern in ihrer informationsverarbeitenden Rolle besteht.“

Diese Definition ist eine Verallgemeinerung der in [WINTER et al. 1998] erarbeiteten Definition für Krankenhausinformationssysteme.

In der Regel können innerhalb eines Informationssystems Komponenten identifiziert werden, die die Erfüllung ausgewählter Aufgaben unterstützen und in diesem Zusammenhang technisch oder organisatorisch eine Einheit bilden. Sie werden hier nach [HAUX et al. 1998], S. 18 als *Anwendungssysteme* bzw. nach [WINTER et al. 2001] als *Anwendungsbausteine* bezeichnet. Typische Anwendungssysteme in diesem Sinne sind ein Patientenverwaltungssystem, ein Radiologieinformationssystem oder ein Archivverwaltungssystem.

Mit der Verwendung der Bezeichnung „Anwendungsbaustein“ anstelle von „Anwendungssystem“ wird betont, dass eine Komponente bestimmte Funktionen zu Verfügung stellt, aber ohne die Integration mit anderen Komponenten nicht oder nur wenig sinnvoll verwendet werden kann.

Integration ist in [DROSDOWSKI 1994], S. 643 erklärt als

„1. [Wieder]herstellung einer Einheit [aus Differenziertem]; Vervollständigung. 2. Einbeziehung, Eingliederung in ein größeres Ganzes; (...) 3. Zustand, in dem sich etwas befindet, nachdem es integriert worden ist; (...) 4. Berechnung eines Integrals; (...)“

In dieser Arbeit wird Integration im Sinne der dritten Erklärung als Zustand, nicht als Tätigkeit oder Vorgang, verstanden.

Es lassen sich viele weitere Beispiele für Integrationsanforderungen nennen. Da eine vollständige Aufzählung unmöglich ist und nicht jede einzelne Anforderung diskutiert werden kann, werden hier die folgenden Anforderungskategorien unterschieden, denen die meisten Anforderungen zugeordnet werden können:

- physische Integration,
- Datenintegration,
- funktionale Integration,
- semantische Integration,
- Kontextintegration,
- Präsentationsintegration und
- Zugriffsintegration.

Die Aufteilung in die genannten Kategorien wurde hauptsächlich auf der Basis der Werke [HAUX et al. 2004], [VAN BEMMEL and MUSEN 1997], [IMBI/MI 2001] (siehe Anhang A) und [OLSEN 1995] vorgenommen, unter zusätzlicher Berücksichtigung von [HIMSS/RSNA 2003a] und [HEIMBIGNER and MCLEOD 1985]. [HIMSS/RSNA 2003a] ist dabei der erste Teil des IHE IT Infrastructure Technical Framework (IHE ITI TF) der Initiative *Integrating the Healthcare Enterprise (IHE)*. Er spezifiziert Integrationsanforderungen für die Integration radiologischer Anwendungssysteme mit anderen Anwendungssystemen¹. Für die hier vorgenommene Einteilung von Integrationsanforderungen ist dabei hauptsächlich die darin angegebene Einteilung

¹ Der IHE IT Infrastructure Technical Framework, Vol. 1 ([HIMSS/RSNA 2003a]) darf nicht mit dem IHE Technical Framework, Vol. 1 ([HIMSS/RSNA 2003b]) verwechselt werden. Während letzteres Werk hauptsächlich Integrationsanforderungen innerhalb radiologischer Organisationseinheiten spezifiziert, enthält der hier zitierte IHE IT Infrastructure Technical Framework Spezifikationen für die Integration radiologischer Anwendungssysteme mit den Anwendungssystemen anderer Organisationseinheiten.

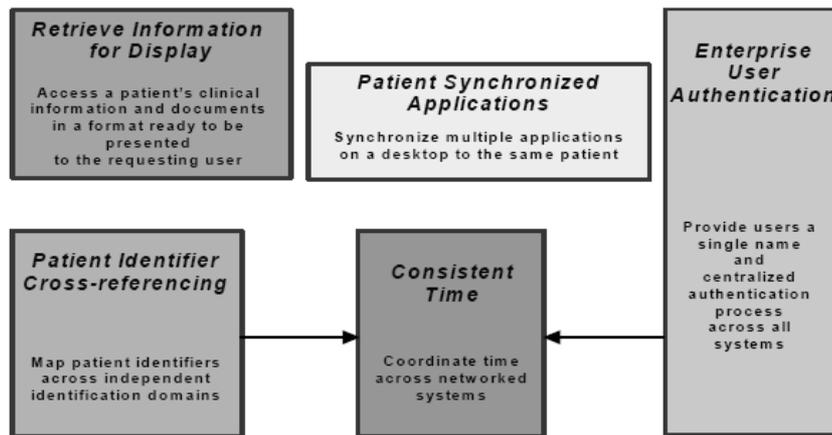


Abbildung 2.1: IT-Infrastruktur-Integrationsprofile des IHE IT Infrastructure Technical Framework, Vol. 1 (Quelle: [HIMSS/RSNA 2003a], Figure 2-1, S. 10)

lung von Nutzeranforderungen in Integrationsprofile relevant (Abbildung 2.1; [HIMSS/RSNA 2003a], S. 10-14).

Die in den Abschnitten zu den einzelnen Anforderungskategorien angegebenen zahlreichen Zitate aus den genannten Werken sollen die Vielfalt der vorhandenen Einteilungen und Definitionen widerspiegeln. Die hier gewählte Einteilung der Kategorien entstand durch Auswahl und Zusammenfassung.

2.2 Kategorien von Integrationsanforderungen

Die folgenden Definitionen beschreiben Kategorien von Integrationsanforderungen. Beispiele für einzelne Anforderungen, die sich den Kategorien zuordnen lassen, sind

- die Forderung nach Einbindung von Tablet-PCs mit Funknetz in das Klinikumsnetzwerk, um mobilen Datenzugriff zu ermöglichen,
- die Forderung nach automatischer Übermittlung von Patientenstammdaten und Falldaten von einem Patientenverwaltungssystem an andere Anwendungssysteme,
- die Forderung nach Einbindung eines Diagnosen- und Prozedurenverschlüsselungsmoduls in unterschiedliche Anwendungssysteme,
- die Forderung nach Bereitstellung und automatischer Anwendung eines aktuellen Diagnosenkataloges auf der Basis der jeweils gültigen Version der Diagnosenklassifikation ICD10 mit Beginn ihrer Gültigkeit in allen betroffenen Anwendungssystemen,
- die Forderung nach automatischer Patientenauswahl auf dem Bildserver der Radiologie, abhängig vom ausgewählten Patienten im Stationsmanagementsystem,
- die Forderung nach einheitlicher Präsentation von Befunden an Stations- und OP-Arbeitsplätzen oder
- die Forderung nach gleichen Benutzerkennungen für PC- bzw. Domänenanmeldungen, das Stationsmanagementsystem sowie den radiologischen Bildserver.

Hinweis: Im Text der Arbeit wird mit Bezug auf Anforderungskategorien oft „die Forderung nach ...“ oder „Anforderung ...“ geschrieben, um lange Formulierungen wie „Anforderungen

aus der Kategorie ...“ zu vermeiden.

2.2.1 Physische Integration

Definition 2.1: Physische Integration ist gegeben, wenn die für jeglichen Datenaustausch notwendige physische Infrastruktur vorhanden ist.

Ende — Definition

Die angegebene Definition wird gestützt durch die Definition in [OLSEN 1995]:

„Technical integration means that you can use various application programs (in multi-session mode) from a terminal or workstation anywhere in the hospital, no matter where the relevant application program resides. It also means that the technical infrastructure (network, etc.) for communicating data from one computer to another, inside or outside the hospital, is available. (...)“ ([OLSEN 1995])

Zur physischen Integration existieren heute verbreitete Standards wie z. B. die IEEE-Standards 802.X für Rechnernetzwerke.

Zur Anforderungskategorie der physischen Integration gehört, zumindest im weiteren Sinne, das Integrationsprofil „Consistent Time“ im IHE ITI TF. Die Synchronisation der Systemzeit verschiedener Rechner oder anderer physischer Komponenten ist wichtige Voraussetzung für das Zusammenarbeiten von Betriebssystemen und Anwendungssystemen:

„*Consistent Time Profile* defines mechanisms to synchronize the time base between multiple actors and computers. Various infrastructure, security, and acquisition profiles require use of a consistent time base on multiple computers. (...)“ ([HIMSS/RSNA 2003a], S. 13)

Hinweis: Auch die Realisierung verschiedener Anwendungssysteme auf ein und demselben Rechner ist physische Integration.

2.2.2 Datenintegration

Definition 2.2: Datenintegration ist gegeben, wenn bestimmte Daten, z. B. Krankenhausfalldaten, nur einmal erfasst werden müssen, auch wenn die dadurch repräsentierten Informationen bei der Arbeit mit mehreren Anwendungssystemen benötigt werden.

Ende — Definition

Die angegebene Definition wird gestützt durch die Definitionen in [HAUX et al. 2004], [VAN BEMMEL and MUSEN 1997] und [HEIMBIGNER and MCLEOD 1985]:

„*Data integration* is guaranteed in a hospital information system when data which have been recorded are available wherever they are needed, without having to be reentered. (...)“ ([HAUX et al. 2004], S. 127)

„*Data integration*. This means that data registered in one application are available to another application, if necessary and provided that it is not conflicting with confidentiality. This prevents repeated recording of the same data and reduces the risk of mistakes.“ ([VAN BEMMEL and MUSEN 1997], S. 351)

„—*Data communication*. Each component has a collection of data, and other components may be interested in accessing some portion of those data. Exchanging the data is the primary activity in a federation, and so a mechanism to support data sharing is essential to the operation of a federation.“ ([HEIMBIGNER and MCLEOD 1985], S. 255)

Die Definition in [OLSEN 1995] bestätigt zunächst die vorhergehenden Definitionen und fordert zusätzlich für den Zustand der vollständigen Datenintegration die einmalige Speicherung jedes Datums:

„Data integration means that a particular data item — once recorded — is immediately available for all relevant and authorized purposes performed by any relevant application. Programs can access data from other applications allowing data to be validated against each other at the time of recording in order to ensure a high data quality. Full data integration requires that each data item is stored in only one logical place, which is ‘known’ by all relevant applications. All data will, however, not necessarily have to be placed in the same physical database nor on the same data server. (...)“ ([OLSEN 1995])

Die einmalige Speicherung wird durch die Definition dieser Arbeit nicht gefordert.

Diese Anforderungskategorie korrespondiert mit dem Integrationsprofil „Retrieve Information for Display“ im IHE ITI TF:

„*Retrieve Information for Display* enables simple and rapid access to patient information for better care. It supports access to existing persistent documents in well-known presentation formats such as CDA, PDF, JPEG, etc. It also supports access to specific key patient-centric information such as allergies, current medications, summary of reports, etc. for presentation to a clinician. (...)“ ([HIMSS/RSNA 2003a], S. 12)

2.2.3 Funktionale Integration

Definition 2.3: Funktionale Integration ist gegeben, wenn Funktionen, die in mehreren Anwendungssystemen benötigt werden, möglichst nur einmal implementiert sind und aus den Anwendungssystemen aufgerufen werden können; dazu gehören u. a. das Präsentieren von Befunddokumenten am Bildschirm und das Verschlüsseln von Diagnosen und Prozeduren.

Ende — Definition

Die angegebene Definition wird gestützt durch die Definitionen in [VAN BEMMEL and MUSEN 1997], [HEIMBIGNER and MCLEOD 1985] und [OLSEN 1995]:

„*Functional integration* means that functions of different applications are available to the qualified user within one user environment.“ ([VAN BEMMEL and MUSEN 1997], S. 351)

„—*Transaction sharing*. A component may not wish to share its data directly, but rather to share operations upon its data. This may be the case if the data are sensitive or have consistency constraints attached to them. In any case, components must be able to define transactions that can be invoked by other components.“ ([HEIMBIGNER and MCLEOD 1985], S. 255)

„Functional integration means that the user can use a function from one application program in connection with other applications in a seamless way.“ ([OLSEN 1995])

Relevanz der funktionalen Integration für Anwendungssystemnutzer

Für die Nutzer von Anwendungssystemen kann es zunächst ohne Bedeutung sein, ob eine bestimmte Funktionalität, z. B. das Suchen von Patientendaten, nur einmal in einem Anwendungssystem implementiert ist und dann von mehreren anderen Anwendungssystemen genutzt werden kann, oder ob die betreffende Funktionalität mehrfach implementiert ist. In letzterem Fall muss „nur“ Datenintegration zwischen den betreffenden Anwendungssystemen gegeben sein (vgl. auch die Definition von Funktionsintegration in [LEHMANN and MEYER ZU BEXTEN 2002], S. 517).

Im Fall der Mehrfachimplementierung einer bestimmten Funktionalität in mehreren Anwendungsbausteinen ist u. U. neben der erforderlichen Datenintegration auch Präsentationsintegration (siehe Abschnitt 2.2.6) oder ggf. auch Kontextintegration (siehe Abschnitt 2.2.5) zu realisieren. Das gilt, wenn die betreffende Funktionalität die Interaktion mit den Anwendungssystemnutzern einschließt. Der mit der Realisierung der Integration verbundene Aufwand kann bei funktionaler Integration u. U. geringer sein.

Dualität von Datenintegration und funktionaler Integration

Zwischen Datenintegration und funktionaler Integration besteht eine Dualitätsbeziehung bezüglich des Zugriffs auf Informationen:

1. Wenn zwischen verschiedenen Anwendungssystemen funktionale Integration hinsichtlich einer bestimmten Funktionalität gegeben ist, dann müssen die mit dieser Funktionalität verarbeiteten Informationen, genauer: die die Informationen repräsentierenden Daten, nicht zwischen den betreffenden Anwendungssystemen abgeglichen werden. Der Zugriff erfolgt nur über die gemeinsam genutzten Funktionen, wodurch eine einmalige Speicherung genügt.
2. Wenn die in verschiedenen Anwendungssystemen benötigte Funktionalität mehrfach implementiert ist, dann kann durch Datenintegration prinzipiell dieselbe Leistungsfähigkeit der Anwendungssysteme wie bei funktionaler Integration erreicht werden.

Im zweiten Fall ist, wie bereits im Zusammenhang mit der Relevanz der funktionalen Integration beschrieben, mit erhöhtem Integrationsaufwand für zusätzliche Präsentationsintegration und Kontextintegration zu rechnen.

2.2.4 Semantische Integration

Definition 2.4: Semantische Integration ist gegeben, wenn in unterschiedlichen Anwendungssystemen dieselben Begriffssysteme verwendet werden, d. h. wenn Daten gleich interpretiert werden.

Ende — Definition

Die Verwendung von Begriffssystemen wird im Kasten „Exkurs zur semantischen Integration“ erörtert. Zur semantischen Integration gehört u. a. der Abgleich von Diagnose- und Prozedurkatalogen zwischen verschiedenen Anwendungssystemen.

Die angegebene Definition wird gestützt durch die Definition in [OLSEN 1995]:

„Semantic integration means that the sender and the receiver in a communication of data use the same terminology, data definitions and reference model, in order to ensure full and correct understanding of the semantic content.“ ([OLSEN 1995])

Zu dieser Anforderungskategorie gehört auch das Integrationsprofil „Patient Identifier Cross-referencing“ im IHE ITI TF:

„The *PIX profile* supports the cross-referencing of patient identifiers from multiple Patient Identifier Domains. These cross-referenced patient identifiers can then be used by “identity consumer” systems to correlate information about a single patient from sources that “know” the patient by different identifiers. (...)“ ([HIMSS/RSNA 2003a], S. 12)

Wenn es nicht möglich ist, dieselben Begriffssysteme zu verwenden, dann muss durch geeignete Funktionalität eine gegenseitige Übersetzung der Begriffssysteme ermöglicht werden.

2.2.5 Kontextintegration

Definition 2.5: Kontextintegration ist gegeben, wenn beim parallelen Arbeiten mit mehreren Anwendungssystemen ein bestimmter Kontext, z. B. Anmeldekontext oder Patientenkontext, nur einmal hergestellt werden muss und dann automatisch zwischen den Anwendungssystemen abgeglichen wird.

Kasten 2.3: Exkurs zur semantischen Integration

Semantische Integration von Anwendungssystemen soll sicherstellen, dass die Bedeutung von bestimmten Daten, d. h. ihre Interpretation, in diesen Anwendungssystemen dieselbe ist. Die Daten sollen also dieselben Informationen repräsentieren.

Legt man Definitionen aus der Logik zugrunde, dann ist eine Interpretation eine Abbildung von Symbolen auf Objekte der realen Welt. Zu den Objekten zählen dabei nicht nur Gegenstände, sondern auch Aktionen, Informationen oder Funktionen. Bei der Implementierung von Anwendungssystemen (Software Engineering, Customizing) werden spezielle Modelle der realen Welt erstellt, um die Daten, welche Informationen über die reale Welt repräsentieren, in geeigneten Strukturen speichern und verarbeiten zu können.

Für viele Informationen z. B. für Laborergebnisse, Diagnosen oder Angaben zur Aufnahmeart, wird die beabsichtigte Interpretation der Daten dadurch sichergestellt, dass zu den zu interpretierenden „Arbeits“-Daten zusätzliche Daten, die erklärende Informationen repräsentieren, in den Anwendungssystemen gespeichert werden. Beispiele sind Referenzbereichskataloge inkl. zugehöriger Bewertungen für Laborergebnisse oder Diagnosenkataloge mit Diagnosenkürzeln und zugehörigen erklärenden Langtexten. Diese zusätzlichen Daten stellen als Texte und/oder Bilder Bezüge zu Objekten der realen Welt her.

Eine Sammlung von erklärenden Daten für bestimmte Informationen wird hier **Begriffssystem** genannt^a (nach [LEINER et al. 1997], S. 38 und 125). In Begriffssystemen kann sich, wie z. B. im ICD10-Diagnosenkatalog, neben der Bereitstellung von Erklärungstexten eine Systematik der enthaltenen Symbole und folglich eine Systematik der Objekte der realen Welt ausdrücken.

Mit den bisherigen Erklärungen heißt *semantische Integration* von bestimmten Anwendungssystemen, dass in diesen Anwendungssystemen dieselben Begriffssysteme zur Interpretation von Daten zur Verfügung stehen oder dass, bei Verwendung von verschiedenen Begriffssystemen, die betreffenden Begriffssysteme durch spezielle Funktionalität aufeinander abgebildet werden.

Abgrenzung zur semantischen Datenmodellierung und zur semantischen Integrität

In der Informatik ist der Begriff semantische Datenmodellierung bekannt. Die semantische Datenmodellierung ist eine Methode zur Modellierung von Informationen, die sehr oft zur Erstellung von konzeptionellen Datenmodellen verwendet wird. Konzeptionelle Datenmodelle formalisieren die reale Welt zunächst unabhängig von Implementierungstechniken, i. d. R. mit Hilfe von Klassen und ihren Beziehungen. Aus konzeptionellen Datenmodellen werden, abhängig von Implementierungstechniken, logische Datenmodelle gebildet, z. B. relationale oder objektorientierte Datenmodelle. Beim Übergang von einem konzeptionellen Datenmodell zu einem relationalen Datenmodell müssen beispielsweise n:m-Beziehungen aufgelöst oder Spezialisierungsbeziehungen durch geeignete Relationen ersetzt werden ([VOSSEN 2000], S. 72-79 (Abschnitt 4.3), [BALZERT 1996], S. 137-162 (Kapitel 2)).

Bestimmte Beziehungen im logischen Datenmodell, die für eine adäquate Abbildung der realen Welt notwendig sind, z. B. Fremdschlüsselbeziehungen, nennt man auch semantische Integritätsbedingungen. Oft wird beispielsweise festgelegt, dass Objekte nur über Elemente aus einem Begriffsverzeichnis beschrieben werden dürfen, z. B. durch eine Vorschrift zur Dokumentierung von Diagnosen nach ICD10. Dafür werden im logischen Datenmodell semantische Integritätsbedingungen, typischerweise Fremdschlüsselbeziehungen, angegeben.

^a Auch wenn in den erklärenden Daten nur ein einziger Begriff beschrieben wird, können diese Daten als Repräsentation eines Begriffssystems betrachtet werden.

Abbildung 2.2 zeigt ein Beispiel für drei hinsichtlich des Anmeldekontextes und des Patientenkontextes abzugleichende Anwendungssysteme.

Die angegebene Definition wird gestützt durch die Definition in [HAUX et al. 2004]:

„*Contextual integration* means that the context is preserved when the application component is changed, e.g., at a healthcare professional workstation. (...)“ ([HAUX et al. 2004], S. 128)

Diese Anforderungskategorie korrespondiert mit dem Integrationsprofil „Patient Synchronized Applications“ im IHE ITI TF:

„*Patient Synchronized Applications* supports viewing data for a single patient among otherwise independent and unlinked applications on a user’s workstation. Its implementation reduces the repetitive tasks of

2 Integrationsanforderungen

selecting the same patient in multiple applications. It also improves patient safety by reducing the chance of medical errors caused by viewing the wrong patient's data. (...)“ ([HIMSS/RSNA 2003a], S. 12)

Sie überdeckt teilweise das Integrationsprofil „Enterprise User Authentication“ im IHE ITI TF (vgl. Abschnitt 2.2.7).

2.2.6 Präsentationsintegration

Definition 2.6: Präsentationsintegration ist gegeben, wenn Eingabe- und Präsentationselemente für die gleichen Daten in unterschiedlichen Anwendungssystemen gleich oder ähnlich gestaltet sind und deshalb nicht jeweils neu erlernt werden müssen.

Ende — Definition

Die angegebene Definition wird gestützt durch die Definitionen in [HAUX et al. 2004], [VAN BEMMEL and MUSEN 1997] und [OLSEN 1995]:

„Presentation integration is guaranteed when different application components represent data as well as user interfaces in a unified way. (...)“ ([HAUX et al. 2004], S. 128)

The screenshot displays three overlapping application windows on a PC screen:

- Top Window (Intranet-Browser SAP 51):** Shows patient information for Universitätsklinikum Leipzig, A&R, Institut für Laboratoriumsmedizin, Klinische Chemie und Molekulare Diagnostik. Patient name is redacted. Address: Liebigstr. B.2, 04103 Leipzig. Lab results for various tests are listed on the right.
- Middle Window (MCCProd: OP-Dokumentation):** Shows a table of patient appointments in the Neurochirurgie department. The selected patient is B2103, with a diagnosis of Glioblastom Frontallapp.
- Bottom Window (MCCProd: OP-Dokumentation):** Shows detailed patient data and a table of operating room times. The patient's name is redacted. The table lists various procedures and their durations.

Abbildung 2.2: Drei parallel genutzte Anwendungssysteme: ein Stationsmanagementsystem, ein OP-Dokumentationssystem und der Intranet-Server des Zentrallabors (Quelle: Bildschirmfoto von einem PC am UKL)

„*Presentation integration* implies that data from various applications are presented to the user in an adequate and consistent way. Especially for dynamically changing data, this is not self-evident.“ ([VAN BEMMEL and MUSEN 1997], S. 351)

„User interface integration means that features like screen dialogue, function keys, help functions, macros, presentation layout etc., are roughly the same in all applications.“ ([OLSEN 1995])

2.2.7 Zugriffsintegration

Definition 2.7: Zugriffsintegration ist gegeben, wenn die Verwaltung von Benutzerkennungen und Zugriffsrechten für unterschiedliche Anwendungssysteme einheitlich erfolgt.

Ende — Definition

Die Definition aus [HAUX et al. 2004] schließt die Bereitstellung von Anwendungssystemen ein und ist allgemeiner als die Definition dieser Arbeit:

„*Access integration* is guaranteed when the application components needed for the completion of a certain task can be used where they are needed. (...)“ ([HAUX et al. 2004], S. 128)

Diese Anforderungskategorie korrespondiert mit dem Integrationsprofil „Enterprise User Authentication“ im IHE ITI TF:

„*Enterprise User Authentication (EUA)* – Defines a means to establish one name per user that can then be used on all of the devices and software that participate in this integration profile. It greatly facilitates centralized user authentication management and provides users with the convenience and speed of a single sign-on. This profile leverages Kerberos (RFC 1510) and the HL7 CCOW standard (user subject). (...)“ ([HIMSS/RSNA 2003a], S. 12)

3 Informationssystemarchitekturen — Eine Einführung

3.1 Vorbemerkungen

„**Ar|chi|tek|tur** *die*; -, -en (aus gleichbed. *lat.* architectura): 1. a) (ohne Plur.) Baukunst [als wissenschaftliche Disziplin]; b) Baustil. 2. der nach den Regeln der Baukunst gestaltete Aufbau eines Gebäudes.“ ([DROSDOWSKI 1994], S. 134)

Mit dieser Erklärung aus einem Fremdwörterbuch könnte der Baukunst die Informatik als Kunst des Aufbaus von Informationssystemen gegenübergestellt werden. Der „nach den Regeln der Baukunst gestaltete Aufbau eines Gebäudes“ entspräche dann dem nach den Regeln der Informatik gestalteten Aufbau eines Informationssystems.

DIE Definition für *Architektur* existiert nicht. In den meisten Definitionen spiegelt sich die Idee der Zerlegung eines Systems in Komponenten mit bestimmten Beziehungen wider (siehe Kasten „Grundbegriffe (3)“, vgl. [BACHMANN et al. 2000], S. 1-2).

Jede Definition für Architektur, die aus einem Satz oder nur wenigen Sätzen besteht, kann die Komplexität des Architekturbegriffes, auch für Informationssysteme, kaum fassen. Die Betrachtung der Architektur von Informationssystemen kann bezüglich sehr unterschiedlicher Fragestellungen und unter Berücksichtigung sehr unterschiedlicher Details erfolgen.

Eine Unterstützung bei der Orientierung innerhalb dieser Vielfalt geben Rahmenwerke zu Informationssystemarchitekturen. Sie beschreiben Vorgehens-Referenzmodelle für die Architekturentwicklung, verschiedene Sichten auf die Architektur oder auch Architektur-Referenzmodelle. Rahmenwerke unterstützen das Management von Informationssystemen, d. h. die Initiierung und Steuerung von Projekten, das Auswählen von Analyse- und Entwurfsmethoden, das Treffen von Auswahlentscheidungen für oder gegen bestimmte zur Auswahl stehende Komponenten usw. Im Kapitel 4 werden einige Rahmenwerke vorgestellt.

Kasten 3.1: Grundbegriffe (3): Definitionen für Architektur

Standard IEEE 1471-2000 „Recommended Practice for Architectural Description of Software-Intensive Systems“:

„Architecture: the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.“ ([HILLIARD 2000])

The Open Group Architectural Framework:

„In TOGAF, “Architecture” has two meanings depending upon its contextual usage:

1. A formal description of a system, or a detailed plan of the system at component level to guide its implementation.
2. The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.“ ([THE OPEN GROUP 2001], Part I, TOGAF FAQ)

Spezifikation der Unified Modeling Language (UML):

„Architecture: The organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.“ ([OMG 2003c], Glossary-2)

3.2 Architektur: Komponenten und ihre Beziehungen

Das Reference Model for Open Distributed Processing (RM-ODP) der International Organization for Standardization (ISO) wurde auf der Grundlage ausführlicher Definitionen der bei der Architekturbeschreibung zulässigen Begriffe entwickelt. Der Standard *ISO/IEC 10746-2 (RM-ODP Foundations)* definiert beispielsweise die im Standard *ISO/IEC 10746-3 (RM-ODP Architecture)* verwendeten Begriffe zur Architekturbeschreibung¹. Darin werden für die Architekturbeschreibung u. a.

- die Begriffskategorie *basic modelling concepts* mit Begriffen wie *Objekt (object)*, *Aktion (action)*, *Schnittstelle (interface)*, *Verhalten (behaviour)* usw.,
- die Begriffskategorie *specification concepts* mit Begriffen wie *Komposition (composition)*, *Verfeinerung (refinement)*, *Typ (type)*, *Subtyp (subtype)* usw. und
- die Begriffskategorie *structuring concepts* mit
 - Begriffen zur Organisation von Objekten, z. B. *Gruppe (group)*, *Konfiguration (configuration)* und *Domäne (domain)*,
 - Begriffen zur Beschaffenheit von Systemen und Objekten, z. B. *Transparenz (transparency)* und *Vertrag (contract)*,
 - Begriffen zur Bezeichnung, z. B. *Namensraum (name space)* und *Namensgraph (naming graph)*,
 - Begriffen zur Verhaltensweise von Objekten, z. B. *initiierendes Objekt (initiating object)* und *Versagen (failure)*, und
 - Begriffen zum Management von Systemen, z. B. *Anwendungsmanagement (application management)* und *Managementinformation (management information)*

unterschieden (Kasten „Auszüge aus ISO/IEC 10746-2 (RM-ODP Foundations)“). Die übrigen Kategorien definieren Grundlagenbegriffe wie *Entität (entity)*, *Aussage (proposition)* oder *Satz (sentence)* und Begriffe zur Konformitätsbeschreibung wie *Portabilität (portability)* ([ISO/IEC JTC 1 1996b], S. 2-18, siehe auch Abschnitt 4.2.2).

Die Begriffskategorien aus dem Standard *ISO/IEC 10746-2 (RM-ODP Foundations)* implizieren, dass die Architektur eines Informationssystems durch die Spezifikation von Komponenten (im Standard: „Objects“) und die Spezifikation von Beziehungen zwischen diesen Komponenten festgelegt ist. Zur Spezifikation von Komponenten werden neben den Basis-Modellierungsbegriffen (*basic modelling concepts*) die Spezifikationsbegriffe (*specification concepts*) verwendet. Für die Spezifikation von Beziehungen werden neben den Basis-Modellierungsbegriffen die Strukturbegriffe (*structuring concepts*) verwendet. Die verschiedenen Unterkategorien der Strukturbegriffe lassen erkennen, dass Architekturbeschreibungen durch die Nutzung vieler unterschiedlicher Begriffe sehr vielfältig sein können.

Definitionen für den Architekturbegriff und für Architekturbeschreibungen, die ein präzises Verständnis und infolgedessen einen Vergleich verschiedener Architekturkonzepte und Architekturstile ermöglichen, wurden auch ausführlich in den Arbeiten der School of Computer Science der Carnegie Mellon University (Pittsburgh, USA) entwickelt (siehe u. a. [GARLAN and SHAW 1994], [ABOWD et al. 1995], [SHAW and GARLAN 1995], [GARLAN et al. 1997] und

¹ Ein weiteres Beispiel für Begriffsdefinitionen, die einer bestimmten Architekturbeschreibung zugrundegelegt werden, ist der Abschnitt 5 im Standard *ISO/IEC 7498-1*. Er definiert wichtige Begriffe für die Architekturbeschreibung in den Abschnitten 6 und 7 desselben Standards, in denen das bekannte OSI-Referenzmodell definiert wird ([ISO/IEC JTC 1 1996c], siehe auch Abschnitt 4.2.1).

Kasten 3.2: Auszüge aus ISO/IEC 10746-2 (RM-ODP Foundations)

„8 Basic modelling concepts

The detailed interpretation of the concepts defined in this clause will depend on the specification language concerned, but these general statements of concepts are made in a language-independent way to allow the statements in different languages to be interrelated.

The basic concepts are concerned with existence and activity: the expression of what exists, where it is and what it does.

8.1 Object: A model of an entity. An object is characterized by its behaviour (...) and, dually, by its state (...). An object is distinct from any other object. An object is encapsulated, i.e. any change in its state can only occur as a result of an internal action or as a result of an interaction (...) with its environment (...). (...)

9 Specification concepts

9.1 Composition

a) (Of objects) – A combination of two or more objects yielding a new object, at a different level of abstraction. The characteristics of the new object are determined by the objects being combined and by the way they are combined. The behaviour of a composite object is the corresponding composition of the behaviour of the component objects.

b) (Of behaviours) – A combination of two or more behaviours yielding a new behaviour. The characteristics of the resulting behaviour are determined by the behaviours being combined and the way they are combined. (...)

(zitiert aus [ISO/IEC JTC 1 1996b], S. 3-5)

[SHAW and GARLAN 1996]). Diese und viele andere Arbeiten zur Architektur sind hauptsächlich auf Softwarearchitekturen bezogen. Sie können jedoch auch allgemeiner auf Informationssystemarchitekturen angewendet werden, wenn softwarespezifische Details weggelassen werden. Sie bilden eine wesentliche Grundlage für die folgenden Ausführungen.

Viele Werke geben eine kurze Definition für Architektur an (vgl. Kasten „Grundbegriffe (3)“), die dann durch ergänzende Definitionen und ausführliche Erklärungen zu einer umfangreichen Beschreibungssprache bzw. Modellierungsmethode ausgebaut werden. Der folgende Satz aus [SHAW and GARLAN 1996]² sei hier Ausgangspunkt für die Definition des Architekturbegriffes³:

„The *architecture of a software system* defines that system in terms of computational components and interactions among those components.“ ([SHAW and GARLAN 1996], S. 3)

Der Satz ähnelt den in Kasten „Grundbegriffe (3)“ angegebenen Definitionen für Architektur. In allen Beschreibungen und Definitionen werden neben den Komponenten eines Systems auch die Beziehungen zwischen diesen Komponenten als wesentliche Elemente der Architektur genannt (vgl. Kasten „Grundbegriffe (3)“):

„(...) components and interactions among those components.“ (aus oben zitiertem Satz)

„(...) components, their relationships to each other (...).“ ([HILLIARD 2000])

„(...) structure of components, their interrelationships, (...).“ ([THE OPEN GROUP 2001], Part I, TOGAF FAQ)

„(...) organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. (...).“ ([OMG 2003c], Glossary-2)

² Originalbeitrag ist [GARLAN and SHAW 1994]

³ Diese Definition wird auch im Model Driven Architecture Guide der Object Management Group referenziert ([OMG 2003b], vgl. Abschnitt 4.2.3).

„(...) we find *components*, both primitive and composite; *rules of composition* that allow the construction of nonprimitive components, or systems; and *rules of behaviour* that provide semantics for the system.“
([SHAW and GARLAN 1996], S. 4)

Auf der Basis der bisherigen Ausführungen wird Architektur von Informationssystemen hier folgendermaßen definiert:

Definition 3.1 Die **Architektur** eines Informationssystems ist die Beschreibung seiner Komponenten und der Beziehungen zwischen den Komponenten.

Ende — Definition

Universell anwendbare Definitionen bzw. Vorgehensweisen zur Ermittlung von Komponenten und Beziehungen innerhalb einer Architektur existieren bisher nicht. Die Kriterien für die Abgrenzung von Komponenten und für die Beschreibung ihrer Beziehungen hängen von

- der Art des Systems, dessen Architektur betrachtet wird,
- den Zielen der Architekturbetrachtung und
- den eingenommenen Sichtweisen

ab. Bei der Betrachtung der Architektur eines Informationssystems können beispielsweise Informationsklassen der verarbeiteten Informationen die Rolle der Komponenten einnehmen. Die Beziehungen zwischen den Informationsklassen nehmen dann die Rolle der Beziehungen zwischen den Komponenten ein. Die Rolle der Komponenten kann aber z. B. auch durch Klassen in einem objektorientierten Softwareentwurf eingenommen werden; die Verknüpfungen von Objekten, die durch die Festlegung von gegenseitigen Methodenaufrufen entstehen, z. B. in Sequenzdiagrammen, nehmen dann die Rolle von Interaktionsbeziehungen zwischen den Komponenten ein (siehe z. B. [EMMERICH 2003], S. 48-50).

Architekturentwicklungs- bzw. Architekturweiterentwicklungsprojekte verlangen also nach einer Festlegung von Sichtweisen auf die Architektur und von Zielen der Architekturbetrachtung, abhängig von der Art des betrachteten Systems. Darauf aufbauend können dann die möglichen bzw. sinnvollen Komponententypen und Beziehungstypen festgelegt werden. Ein solches Vorgehen ist, wenn keine Empfehlungen oder Vorgaben genutzt werden, für die meisten Projekte zu aufwendig. Deshalb wurden von verschiedenen Organisationen aber auch Einzelaufbauern Rahmenwerke für Informationssystemarchitekturen entwickelt. Sie sind Orientierungshilfen in der theoretischen Vielfalt von Sichtweisen, Komponententypen usw. In Kapitel 4 werden einzelne Rahmenwerke für Informationssystemarchitekturen vorgestellt.

3.3 Architekturstile

Wenn Beschreibungen von Komponenten und ihren Beziehungen, d. h. Architekturen, nicht nur für ein einzelnes, sondern für mehrere Systeme gelten, dann werden die betreffenden Architekturen auch als Architekturstile bezeichnet. Anstelle von „Stil“ wird oft auch die Bezeichnung *Muster* (pattern) oder *Entwurfsmuster* verwendet ([SHAW and GARLAN 1996], S. 19, [GAMMA et al. 1997], S. 2-4).

In [GAMMA et al. 1997] wird, mit Bezug zur objektorientierten Entwicklung, der Begriff Entwurfsmuster folgendermaßen definiert:

„Entwurfsmuster (...) sind Beschreibungen zusammenarbeitender Objekte und Klassen, die maßgeschneidert sind, um ein allgemeines Entwurfsproblem in einem bestimmten Kontext zu lösen.“ ([GAMMA et al. 1997], S. 4)

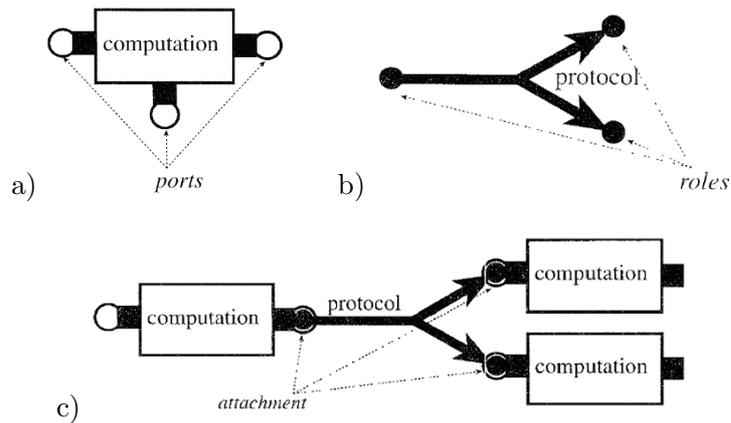


Abbildung 3.1: Ein Komponententyp (a), ein Konnektorentyp (b) und eine Konfiguration (c) (Quelle [ABOWD et al. 1995], S. 329-331, Abb. 3 - Abb. 5)

In der Literatur zu softwarebezogenen Architekturstilen wird einleitend oft auch die vom „echten“ Architekten Christopher Alexander in seinem Werk „A pattern language : towns, buildings, construction“ ([ALEXANDER 1977]) angegebene Beschreibung des Begriffes Muster zitiert:

„Jedes Muster beschreibt ein in unserer Umwelt beständig wiederkehrendes Problem und erläutert den Kern der Lösung für dieses Problem, so dass sie diese Lösung beliebig oft anwenden können, ohne sie jemals ein zweites Mal gleich auszuführen.“ (übersetzt aus [ALEXANDER 1977])

In [SHAW and GARLAN 1996] (siehe auch [GARLAN 1995]) werden Architekturstile über die durch sie angegebenen

- Definitionen von Sprachen zur Beschreibung von Komponenten und ihren Beziehungen⁴ und
- Regeln für die Nutzung der Sprachen

charakterisiert:

„An architectural style (...) defines a family of such systems in terms of a pattern of structural organization. More specifically, an architectural style defines a *vocabulary* of components and connector types, and a set of *constraints* on how they can be combined.“ ([SHAW and GARLAN 1996], S. 20)

In [ABOWD et al. 1995] (siehe auch [SHAW and GARLAN 1996], S. 129-146) wird das, was unter einem Architekturstil zu verstehen ist bzw. was zur Beschreibung eines Architekturstils gehört, folgendermaßen präzisiert (Abbildung 3.1):

1. Zur Beschreibung von Architekturen wird eine formale abstrakte Syntax benötigt. Sie ist unabhängig von konkreten Architekturstilen. Die grundlegenden syntaktischen Elemente sind⁵

⁴ In [SHAW and GARLAN 1996] wie auch im nachfolgend referenzierten [ABOWD et al. 1995] sind die Beziehungen auf Interaktionsbeziehungen beschränkt. Andere Beziehungstypen, z. B. Aggregation oder Vererbung, werden nicht berücksichtigt.

⁵ In [ABOWD et al. 1995] werden nicht die Bezeichnungen Komponententyp (component type) und Konnektorentyp (connector type) verwendet, sondern Komponente (component) und Konnektor (connector). In der Definition der Konfigurationen wird jedoch deutlich, dass Typen gemeint sind, die dann in Konfigurationen ausgeprägt werden.

Kasten 3.3: Grundbegriffe (4)

Definitionen für *Modell*

Definition aus [SCHÜTTE 1998] „Grundsätze ordnungsmäßiger Referenzmodellierung“:

„Ein Modell ist das Ergebnis einer Konstruktion eines Modellierers, der für Modellnutzer Elemente eines Originals zu einer Zeit als relevant mit Hilfe einer Sprache deklariert.“ ([SCHÜTTE 1998], S. 59)

Definition aus [HAUX et al. 2004] „Strategic Information Management in Hospitals“:

„A simplified depiction of reality or excerpts of it. Models are usually adapted to answer certain questions or to solve certain tasks.“ ([HAUX et al. 2004], S. 239)

Einfache Modelle von Informationssystemen können z. B. aus Listen von Anwendungsbausteinen, aus Tabellen von Aufgaben der Informationsverarbeitung oder aus Grafiken von Rechnern und ihren Netzwerkverbindungen bestehen.

Erklärung und Definition für *Metamodell*

Um Modellierung zu leiten und vergleichbar zu machen, aber auch um auf der Basis von existierenden Modellen übergreifende Tatsachen zu formulieren, werden Metamodelle geschaffen. Das in dieser Arbeit erweiterte 3LGM² ist ein Metamodell für Informationssysteme.

Im Kasten „Grundbegriffe (1)“ wurden die Begriffe Information und Wissen definiert. Mit ihnen kann für die Beziehung zwischen Modellen und Metamodellen von Informationssystemen folgende Erklärung angegeben werden: Während ein Modell Tatsachenkenntnisse über ein Informationssystem darstellt, enthält ein Metamodell Wissen zur Interpretation von Modellen und gibt ggf. Vorschriften zur Erstellung von Modellen an.

Layer	Description	Example
meta-metamodel	The infrastructure for a metamodeling architecture. Defines the language for specifying metamodels.	MetaClass, MetaAttribute, MetaOperation
metamodel	An instance of a meta-metamodel. Defines the language for specifying a model.	Class, Attribute, Operation, Component
model	An instance of a metamodel. Defines a language to describe an information domain.	StockShare, askPrice, sellLimitOrder, StockQuoteServer
user objects (user data)	An instance of a model. Defines a specific information domain.	<Acme_SW_Share_98789>, 654.56, sell_limit_order, <Stock_Quote_Svr_32123>

Abbildung: Meta-Metamodell, Metamodell, Modell, Anwendungsdaten (Quelle: [OMG 2003c], Abschnitt 2.2.1)

Definition aus [HAUX et al. 2004] „Strategic Information Management in Hospitals“:

„Language for describing models of a certain class. It usually describes the modeling framework, which consists of the modeling syntax and semantics (...), the representation of the objects (...), and (sometimes) the modeling rules (e.g., the modeling steps).“ ([HAUX et al. 2004], S. 238)

Definitionen für *Referenzmodell*

Definition aus [WINTER et al. 1999] „Referenzmodelle für die Unterstützung des Managements von Krankenhausinformationssystemen“ (vgl. [LEHMANN and MEYER ZU BEXTEN 2002], S. 489):

„Sei eine Klasse S von Sachverhalten gegeben. Ein Modell R ist *Referenz für eine Klasse S* oder R ist *Referenzmodell für die Klasse S* genau dann, wenn R ein allgemeines Modell ist, das

- als Grundlage für die Konstruktion spezieller Modelle der Sachverhalte der Klasse S oder
- als Vergleichsobjekt für Modelle von Sachverhalten der Klasse S

dienen kann.“ ([WINTER et al. 1999], S. 176)

Definition aus [HAUX et al. 2004] „Strategic Information Management in Hospitals“:

„Model pattern for a certain class of aspects. On the one hand, these model patterns can help to derive more specific models through modifications, limitations, or add-ons (generic reference models). On the other hand, these model patterns can be used to directly compare models e.g. concerning their completeness (nongeneric reference models).“ ([HAUX et al. 2004], S. 245)

- Komponententypen, die die aktiven Einheiten eines Systems beschreiben und für die Interaktion mit anderen Komponenten Schnittstellentypen (im Original: „ports“)⁶ bereitstellen,
 - Konnektorentypen, die die Verknüpfung von Schnittstellen und damit das Beschreiben der Interaktion von Komponenten ermöglichen, und
 - Konfigurationen, die das (syntaktische) Ausprägen und Zusammensetzen von Komponententypen und Konnektorentypen ermöglichen.
2. Für einen konkreten Architekturstil wird die Bedeutung der Elemente der Syntax durch ein semantisches und eine Abbildung aus der Syntax in das semantische Modell festgelegt. Durch das semantische Modell wird definiert, welche Komponententypen tatsächlich zu Architekturen des betreffenden Architekturstils gehören und welche Einschränkungen es für die Ausprägungen der Komponententypen und der Konnektorentypen sowie für ihre Zusammensetzung gibt.
 3. Ergänzend kann die Syntax für einen bestimmten Architekturstil durch spezielle Festlegungen eingeschränkt werden, und es können Vereinfachungen für bestimmte syntaktische Strukturen definiert werden.

Der in [ABOWD et al. 1995] beschriebene Ansatz zur Formalisierung von Architekturen und Architekturstilen wird u. a. in Arbeiten zur Architekturbeschreibungssprache *ACME* weitergeführt (siehe z. B. [GARLAN et al. 1997]).

Auf der Basis der Ausführungen dieses Abschnitts wird hier der Begriff *Architekturstil* für die folgenden Kapitel definiert:

Definition 3.2 Architekturstile für Informationssystemarchitekturen werden durch

- Komponententypen,
- Konnektorentypen und
- Konfigurationen, die Komponententypen und Konnektorentypen ausprägen und zusammensetzen,

beschrieben. Für die Beschreibung müssen

- eine Syntax und
- die zugehörige Semantik

definiert sein. Über die Syntax wird festgelegt, wie Ausprägungen von Komponenten und Konnektoren notiert werden. Über die Semantik werden die zum Architekturstil gehörenden Komponenten- und Konnektorentypen sowie Regeln für die Bildung von Konfigurationen festgelegt.

Ende — Definition

Muster bzw. Entwurfsmuster im Sinne der am Anfang dieses Abschnitts zitierten Definitionen von Alexander bzw. Gamma entsprechen Konfigurationen im Sinne der Ausführungen von [ABOWD et al. 1995]. Muster können insofern als Architekturstile angesehen werden, als sie i. d. R. mit einer speziellen grafischen oder textuellen Syntax definiert werden, für die wiederum, implizit oder explizit, eine bestimmte Semantik festgelegt wurde.

⁶ In [ABOWD et al. 1995] wird darauf hingewiesen, dass „ports“ eine Verallgemeinerung des verbreiteten Schnittstellenbegriffes sind. Hier wurde die Bezeichnung Schnittstelle bzw. Schnittstellentyp beibehalten.

	Pipes und Filter	Objekt-orientierte Organisation	Ereignisbasierte, implizite Aufrufe	Geschichtete Systeme
Komponententypen	Filter	Objekte	Module	Komponenten auf den verschiedenen Schichten
Konnektorentypen	Pipes	Funktionsaufruf, Prozeduraufruf	Bindungen zwischen Ereignistypen und aufzurufenden Prozeduren	→ sind definiert in den Protokollen für die Interaktion zwischen den Ebenen

Tabelle 3.1: Beispiele für Architekturstile (zusammengestellt aus [SHAW and GARLAN 1996], Abschnitte 2.2-2.5, S. 21-25)

In [SHAW and GARLAN 1996] werden Beispiele für Architekturstile vorgestellt. Tabelle 3.1 nennt vier dieser Beispiele und die zugehörigen Komponententypen- und Konnektorentypen.

3.4 Architekturstile für Informationssysteme — Metamodelle und Referenzmodelle

Die im vorhergehenden Abschnitt referenzierte Literatur ist auf Softwaresysteme bzw. ihre Architektur bezogen. Es wurde bereits darauf hingewiesen, dass Teile der darin enthaltenen Ansätze zur Architekturbeschreibung hier allgemeiner auf Informationssystemarchitekturen angewendet werden. Die Verallgemeinerung besteht in der Berücksichtigung von Aspekten, die zunächst nicht ausdrücklich auf Softwarearchitekturen bezogen sind, wie z. B. die Organisationsstruktur eines Unternehmens oder die Informationen, die in einem Unternehmen verarbeitet werden. Ein Werk, das eine den genannten Werken vergleichbare ausführliche Diskussion des Architekturbegriffes mit ausdrücklichem Bezug zu Informationssystemen enthält, war bei der Erstellung dieser Arbeit nicht bekannt.

3.4.1 Architekturstile und Metamodelle

In semantischen Modellen von Architekturstilen werden Komponententypen und Konnektorentypen beschrieben. Das semantische Modell ist ein Metamodell für die Beschreibung der letztlich interessierenden Architekturen konkreter Informationssysteme (vgl. Kasten „Grundbegriffe (4)“).

Ein Beispiel für semantische Modelle, d. h. Metamodelle, für Informationssysteme ist das in der Einleitung (Kapitel A) genannte 3LGM². Es wird in Kapitel 6 vorgestellt und danach für eine ausführliche Diskussion der Bewertung von Architekturen verwendet. Weitere Beispiele für Metamodelle sind das Object Model der Object Management Group (siehe Abschnitt 4.2.3) und das Metamodell der Architektur integrierter Informationssysteme (siehe Abschnitt 4.4.2).

3.4.2 Architekturstile und Referenzmodelle

Im Zusammenhang mit Informationssystemarchitekturen wird sehr oft von Referenzmodellen (siehe Kasten „Grundbegriffe (5)“) gesprochen bzw. geschrieben. Es werden beispielsweise

- Vorgehens-Referenzmodelle für die Entwicklung bzw. Weiterentwicklung von Architekturen,

Kasten 3.4: Grundbegriffe (5): Modelle und Referenzmodelle

Definition und Erklärung für *Informationsmodell* und *Informations-Referenzmodell*

Definition für *Informationsmodell* aus [BECKER and SCHÜTTE 1996] „Handelsinformationssysteme“:

„Ein Informationsmodell ist das immaterielle Abbild des betrieblichen Objektsystems aus Sicht der in diesem verarbeiteten Informationen für Zwecke des Informationssystem- und Organisationsgestalters.“ ([BECKER and SCHÜTTE 1996], S. ????)

Informationsmodelle entsprechen konzeptionellen Datenmodellen in der Softwareentwicklung, insbesondere in der Datenbankentwicklung. Dort beschreiben konzeptionelle Datenmodelle unabhängig von der Implementierungstechnik bzw. dem verwendeten Datenbankmanagementsystem die zu verarbeitenden Informationen. Sie werden dann in technikabhängige logische Datenmodelle überführt (siehe z. B. [VOSSEN 2000], S. 72-79 (Abschnitt 4.3)).

Definition: Ein *Informations-Referenzmodell* ist ein Informationsmodell für eine Klasse von Informationssystemen oder von Anwendungsbereichen.

Definition für *Vorgehens-Referenzmodell*

Definition aus [LEHMANN and MEYER ZU BEXTEN 2002] „Handbuch der Medizinischen Informatik“ (vgl. [WINTER et al. 1999], S. 178):

„Ein Vorgehens-Referenzmodell ist ein allgemeines Modell für eine Klasse von Vorgehensweisen z. B. bei Projekten. Aus dem Vorgehens-Referenzmodell kann die Vorgehensweise in einem speziellen Projekt z. B. in Form eines Projektplans abgeleitet werden.“ ([LEHMANN and MEYER ZU BEXTEN 2002], S. 492)

Definition und Erklärung für *Architekturmodell* und *Architektur-Referenzmodell*

Definition: Ein *Architekturmodell im weiteren Sinn* ist ein Modell, das Komponenten und ihre Beziehungen beschreibt.

Mit der im Abschnitt 3.2 nach Definition 3.1 beschriebenen allgemeinen Auffassung von Komponenten und ihren Beziehungen ist auch die soeben angegebene Definition sehr allgemein. Damit können beispielsweise auch Informationsmodelle als Architekturmodelle bezeichnet werden.

Für diese Arbeit wird der Begriff *Architekturmodell* hier durch die Einschränkung der Komponenten auf ausführbare (oder handelnde) Komponenten bzw. auf funktionale Einheiten enger gefasst. Entsprechende Modelle werden hier als *Architekturmodelle im engeren Sinn* bzw. einfach als *Architekturmodelle* bezeichnet:

Definition: Ein *Architekturmodell im engeren Sinn* ist ein Modell, das ausführbare oder handelnde Komponenten bzw. funktionale Einheiten und die Beziehungen zwischen den Komponenten bzw. Einheiten beschreibt.

Zu Architekturmodellen (im engeren Sinn) werden hier auch solche Modelle gerechnet, die bestimmte Funktionalitäten zu funktionalen Einheiten wie z. B. Diensten oder Dienstgruppen zusammenfassen. Sie schreiben i. d. R. nicht ausdrücklich vor, dass jede funktionale Einheit durch genau eine ausführbare oder handelnde Komponente des Informationssystems realisiert wird; die Abgrenzung von ausführbaren oder handelnden Komponenten erfolgt jedoch oft auf der Basis von funktionalen Einheiten.

Definition: Ein *Architektur-Referenzmodell* ist ein Architekturmodell für eine Klasse von Informationssystemen oder von Anwendungsbereichen.

- Informations-Referenzmodelle für die in bestimmten Anwendungsbereichen verarbeiteten Informationen oder
- Architektur-Referenzmodelle für bestimmte Aufteilungen von Informationssystemen in ausführbare (oder handelnde) Komponenten

bereitgestellt.

Architektur-Referenzmodelle entsprechen Mustern bzw. konkreten Konfigurationen von Komponenten entsprechend den Ausführungen in Abschnitt 3.3.

3.4.3 Bestimmung von Architekturstilen

In [ABOWD et al. 1995] wird für jeden der vorgestellten Architekturstile eine spezielle Semantik bzw. ein spezielles semantisches Modell beschrieben. Wenn ein semantisches Modell sehr allgemein definiert wurde, kann es sinnvoll sein, erst im Zusammenhang mit bestimmten Konfigurationen bzw. Mustern, die typische Komponenten und Beziehungen ausprägen, von Architekturstilen zu sprechen. Ein Architekturstil wird in diesen Fällen erst durch ein(e) oder mehrere Konfigurationen bzw. Muster festgelegt. Das folgende Zitat aus [GAMMA et al. 1997] deutet am Beispiel von Mustern die Abhängigkeit der Ermittlung von Architekturstilen von bestimmten Sichtweisen oder Standpunkten an:

„Die Bestimmung dessen, was ein Muster ist und was nicht, hängt von der jeweiligen Perspektive ab. Was aus einer Perspektive als Muster erscheint, stellt aus einer anderen Perspektive betrachtet einen primitiven Baustein dar.“ ([GAMMA et al. 1997], S. 4)

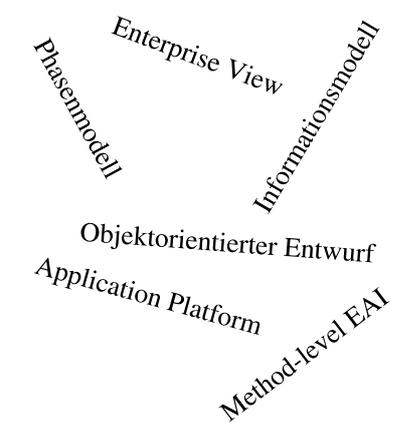
Konfigurationen bzw. Muster für einen Architekturstil können in einem Architektur-Referenzmodell definiert sein.

4 Standards für Informationssystemarchitekturen

4.1 Orientierung durch Rahmenwerke

Rahmenwerke für Informationssystemarchitekturen geben Orientierungshilfen im „Dschungel“

- der verschiedenen Betrachtungsweisen bei der Architekturbetrachtung,
- der Methoden und Werkzeuge für die Systemanalyse und den Systementwurf der
- der Architekturstile mit ihren unterschiedlichen Eignungen für bestimmte Anwendungen oder Problembereiche
- der Techniken zur Herstellung von Integrationen oder
- der Vorgehensweisen für die Durchführung von Projekten im Informationsmanagement.



Einige der hier als Rahmenwerke vorgestellten Werke sind Standards, die von Organisationen wie der Internationalen Organisation für Standardisierung (International Organization for Standardization, ISO) oder dem Europäischen Komitee für Normung (Comité Européen de Normalisation, CEN) publiziert wurden. Andere sind in Organisationen, die sich aus industriellen Interessengruppen entwickelten, entstanden. Zu letzteren Organisationen gehören beispielsweise die Object Management Group (OMG) oder The Open Group. Dabei steht die fehlende Aufnahme eines Standards durch „offizielle“ Organisationen wie der ISO oder dem CEN oft nicht in Zusammenhang mit der praktischen Relevanz des betreffenden Standards, weder positiv noch negativ.

Manche Rahmenwerke, z. B. The Open Group Architectural Framework (siehe Abschnitte 4.2.5 und 4.4.4), unterscheiden verschiedene Phasen (auch: „Stufen“) für eine Architekturentwicklung und stellen ein Vorgehens-Referenzmodell (siehe Kasten „Grundbegriffe (5)“) zur Verfügung.

Andere Rahmenwerke, z. B. das ISO Reference Model for Open Distributed Processing (siehe Abschnitt 4.2.2), unterscheiden verschiedene Sichtweisen auf die Architektur, in denen sich unterschiedliche Rollen bzw. Aufgaben von Personen widerspiegeln: Programmierer, die ein Softwareprodukt erstellen, betrachten Informationssysteme anders als Rechenzentrumsleiter, die die Programmentwicklung in Auftrag geben, als Ärzte und Pflegekräfte, die das Programm zur Erleichterung von Routinetätigkeiten nutzen wollen, oder als Unternehmensvorstände, die eine Prozessoptimierung anstreben.

Die Unterscheidung von Phasen und die Unterscheidung von Sichtweisen unterstützen das strukturierte Vorgehen beim Entwickeln oder Weiterentwickeln von Architekturen. Sie hilft den beteiligten Personen, die eingenommenen Rollen im Entwicklungsprozess zu ermitteln und die damit verbundenen Aufgaben abzugrenzen.

Auch Architektur-Referenzmodelle sind wesentlicher Bestandteil mancher Rahmenwerke, z. B. der Object Management Architecture (siehe Abschnitt 4.2.3). Sie erleichtern die Architekturentwicklung u. a. durch Vorgaben für die Aufteilung von Funktionalität auf Komponenten bzw. Komponentenkategorien.

Die meisten Rahmenwerke werden erst durch zusätzliche Standards oder andere Werke zu mächtigen Werkzeugen. Benötigt werden beispielsweise

- Erläuterungen oder Spezifikationen für einzelne Teile des Rahmenwerkes, z. B. die ausführlichen Spezifikationen für Dienste der in der Object Management Architecture beschriebenen Schnittstellenkategorien, und
- Modellierungsmethoden bzw. -sprachen für die Modellierung von Informationssystemen entsprechend verschiedener Sichten oder Entwicklungsphasen, z. B. die Methode der Ereignisgesteuerten Prozessketten (EPK) oder die Unified Modeling Language (UML).

Dieses Kapitel gibt einen Überblick über aktuelle Rahmenwerke für Informationssystemarchitekturen und mit den Rahmenwerken assoziierte Standards oder andere ergänzende Werke¹. Auf eine Beschreibung von verbreiteten Modellierungssprachen wie der UML wurde dabei verzichtet.

4.2 Technische Rahmenwerke für Informationssystemarchitekturen

4.2.1 Das OSI-Referenzmodell der ISO

Sehr bekannt und oft zitiert ist ein von der Internationalen Organisation für Standardisierung (International Organization for Standardization, ISO) herausgegebenes Referenzmodell: das Referenzmodell für die Kopplung offener Systeme, kurz OSI-Referenzmodell (ISO-Standards

¹ Der Überblick ist durch Auswahl und Zusammenfassung entstanden. Manche hier nicht vorgestellte (Rahmen-)Werke können eine ähnlich große Bedeutung haben wie die vorgestellten.

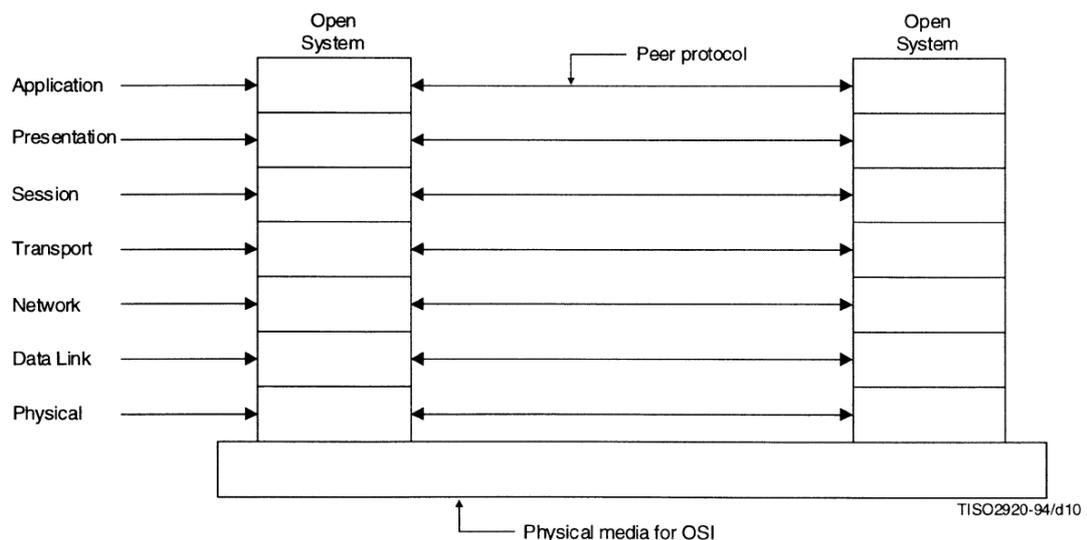


Abbildung 4.1: OSI-Referenzmodell (Quelle: [ISO/IEC JTC 1 1996c], S. 28)

ISO/IEC 7498-1 bis -4: Information technology — Open Systems Interconnection — Basic Reference Model). Das OSI-Referenzmodell ist ein Architektur-Referenzmodell für die Architektur von Kommunikationsbeziehungen zwischen Anwendungssystemen. Für die Architekturbetrachtung sieht es sieben Schichten vor, von denen jede die Dienste der darunterliegenden Schicht nutzt (Abbildung 4.1). Die Schichten und ihre Gegenstände sind (auszugsweise übersetzt aus [ISO/IEC JTC 1 1996c], S. 28 und 32-52, 7.X.2.1):

Die Anwendungssystemschiicht

Gegenstand: Die Anwendungssystemschiicht bildet den einzigen Weg für Anwendungsprozesse, auf die OSI-Umgebung zuzugreifen.

Die Präsentationsschiicht

Gegenstand: Die Präsentationsschiicht stellt die Repräsentation der Informationen bereit, die Anwendungssysteme entweder kommunizieren oder auf die sie sich in ihrer Kommunikation beziehen.

Die Sitzungsschiicht

Gegenstand: Zweck der Sitzungsschiicht ist es, das Werkzeug bereitzustellen, welches kooperierende Präsentationseinheiten benötigen, um ihren Dialog zu organisieren und zu synchronisieren und ihren Datenaustausch zu steuern.

Die Transportschiicht

Gegenstand: Der Transportdienst sorgt für transparente Datenübertragung zwischen Sitzungseinheiten und befreit letztere von allen Details der Art und Weise, auf welche zuverlässige und kostengünstige Datenübertragung ausgeführt wird.

Die Netzwerkschiicht

Gegenstand: Die Netzwerkschiicht stellt die funktionalen und verfahrenstechnischen Mittel für unverbundene und verbundene² Übermittlung zwischen Transporteinheiten bereit und sorgt daher für Unabhängigkeit der Transporteinheiten von Routing- und Weiterleitungsbelangen.

Die Datenübertragungsschiicht

Gegenstand: Die Datenübertragungsschiicht stellt die funktionalen und verfahrenstechnischen Mittel für Netzwerkeinheiten im unverbundenen Modus, für das Aufbauen, die Aufrechterhaltung und das Abbauen von Datenübertragungsverbindungen für Netzwerkeinheiten im verbundenen Modus und für die Übermittlung von datenübertragungsdienstbezogenen Datenelementen bereit.

Die physische Schicht

Gegenstand: Die physische Schicht stellt die mechanischen, elektrischen, funktionalen und verfahrenstechnischen Mittel für die Aktivierung, Aufrechterhaltung und Deaktivierung physischer Verbindungen zur Bitübertragung zwischen Datenübertragungseinheiten bereit.

Die im OSI-Referenzmodell definierten Schichten stehen nicht notwendigerweise für verschiedene Sichtweisen auf die Informationssystemarchitektur. Personen, die für technische Details der Kommunikation zwischen Anwendungssystemen verantwortlich sind, werden u. U. mit mehreren der Schichten in Berührung kommen. Die Schichten sind dann eine Hilfe zur Strukturierung des komplexen Kommunikationsthemas.

Der Standard *ISO/IEC 7498-1* enthält als Grundlage für die sieben OSI-Schichten, aber auch für beliebige andere schichtenorientierte Architekturen, eine ausführliche generische Spezifikation für die Aufteilung von Architekturen in Schichten. Ein zentrales Konzept der Spezifikation ist, dass jede Schicht N der über ihr liegenden Schicht $N + 1$ Dienste zur Verfügung stellt, deren detaillierte Abwicklung der Schicht $N + 1$ verborgen bleibt (Abbildung 4.2; [ISO/IEC JTC 1 1996c], Abschnitt 5).

Spezielle Standards auf der Basis des OSI-Referenzmodells

Auf der Basis des OSI-Referenzmodells wurden Standards bzw. Protokolle für physische Komponenten, z. B. Netzkabel oder Netzkarten, und für verbreitete Protokolle zum Da-

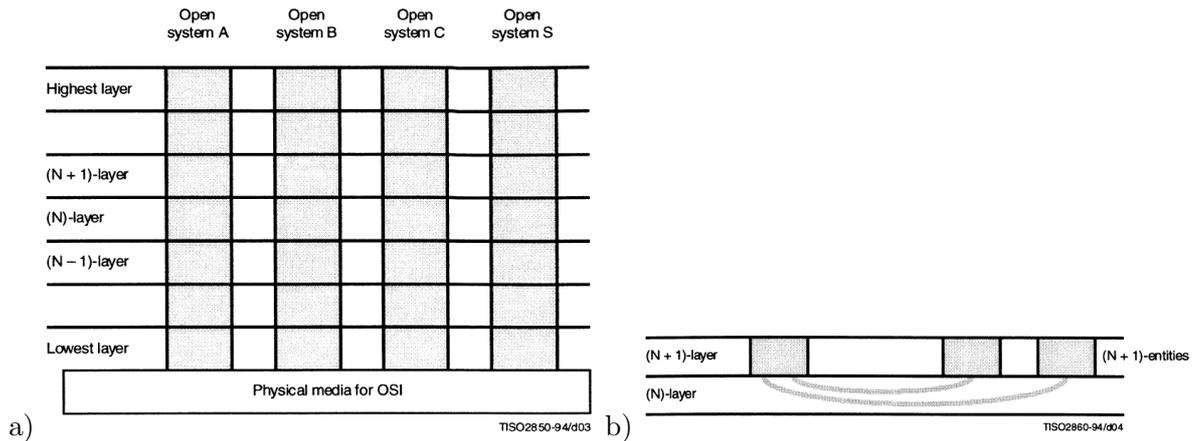


Abbildung 4.2: Übersichtsgrafik zum generischen Schichtenkonzept (a) und Kommunikation von Einheiten der Schicht $N + 1$ über die Schicht N (b) (Quelle [ISO/IEC JTC 1 1996c], S. 7-8, Figure 3 und Figure 4)

tenaustausch entwickelt, bzw. existierende Standards wurden den OSI-Schichten zugeordnet. *Hinweis:* Der Begriff *Protokoll* bezeichnet eine Vorschrift oder einen Standard für die Kommunikation zwischen Rechnersystemen.

Für die Datenübertragungsschicht wurden z. B. die Standards 802.X des Institute of Electrical and Electronics Engineers (IEEE) entwickelt ([WG 2001]). Der Standard 802.3 spezifiziert beispielsweise das sogenannte Ethernet ([IEEE 802.3 WG 2002]), die Standards 802.11x spezifizieren die als Wireless LAN bekannten Techniken ([IEEE 802.11 WG 2003]).

Zur Netzwerkschicht gehört das in RFC 791 definierte Internet Protocol (IP) ([INFORMATION SCIENCES INSTITUTE 1981a]).

Hinweis: Requests for Comments (RFCs) sind Spezifikationsdokumente für viele Techniken, von denen die meisten mit dem Internet verknüpft sind. Sie werden von der Internet Engineering Task Force (IETF) veröffentlicht.

Zur Transportschicht gehört das in RFC 793 und RFC 1122 definierte Transmission Control Protocol (TCP) ([INFORMATION SCIENCES INSTITUTE 1981b], [NETWORK WORKING GROUP 1989]).

Für die Präsentationsschicht existieren Standards für Text- und Bilddaten wie der American

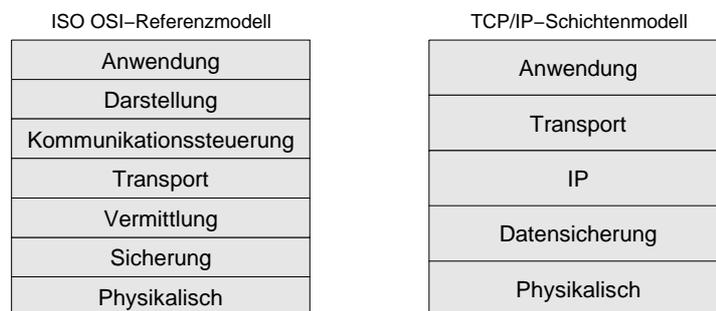


Abbildung 4.3: OSI- und TCP/IP-Schichtenmodelle (nach [WASHBURN and EVANS 1994], S. 9, 157)

Standard Code for Information Interchange (ASCII) und seine Erweiterungen (z. B. [ISO JTC 1 1998]), das Tagged Image File Format (TIFF) ([ISO TC 130 2004]) oder die Standards der Moving Picture Experts Group (MPEG) ([ISO JTC 1 1993]).

Der Anwendungsebene können anwendungsbereichsunabhängige Standards bzw. Protokolle, z. B. Telnet (RFC 854), das File Transfer Protocol (FTP) (RFC 959), das Hypertext Transfer Protocol (HTTP) ([NETWORK WORKING GROUP 1999]) oder der Kerberos Network Authentication Service (1510), aber auch anwendungsbereichsabhängige Kommunikationsstandards, z. B. Health Level Seven (HL7) ([HL7 1999]), zugeordnet werden.

Für die verbreitet genutzten Protokolle TCP und IP wird oft auch eine alternative Aufteilung in Schichten angegeben (Abbildung 4.3).

Integrationsanforderungen und das OSI-Referenzmodell

Die Standards zu den Schichten des OSI-Referenzmodells beschreiben Techniken für physische Integration, d. h. für physische Netzwerkkomponenten wie Netzwerkkarten oder Router, und grundlegende Techniken für Datenintegration. Letztere sind Voraussetzung für die Realisierung von Kommunikationsbeziehungen für spezielle Anwendungssysteme.

Der als Beispiel genannte Standard HL7 wurde unmittelbar für Datenintegration im Gesundheitswesen entwickelt.

4.2.2 Das Referenzmodell für offene verteilte Informationsverarbeitung der ISO

Die ISO-Standards *ISO/IEC 10746-1 bis -4: Information technology — Open Distributed Processing — Reference model* wurden für die Erstellung von Spezifikationen für Informationssysteme herausgegeben, die verteilte Informationsverarbeitung ermöglichen und dabei als offen charakterisiert werden. Sie enthalten umfassende Empfehlungen für die Analyse, den Entwurf und die Bewertung von verteilten Informationssystemen ([ISO/IEC JTC 1 1998b]). Die Standards werden auch als das Referenzmodell für offene verteilte Informationsverarbeitung (Reference Model for Open Distributed Processing, RM-ODP) bezeichnet (Kasten „Überblick über Standards zum Referenzmodell ...“).

Die RM-ODP-Standards werden ergänzt durch weitere ISO-Standards, wie z. B. *ISO/IEC 13235-X*, die viele der in den RM-ODP-Standards zusammenfassend beschriebenen Komponenten bzw. Eigenschaften von Komponenten ausführlich spezifizieren (Kasten „Ergänzende Standards zum RM-ODP ...“).

Grundlagen für die Beschreibung von offener verteilter Informationsverarbeitung

Der Standard *ISO/IEC 10746-2* ([ISO/IEC JTC 1 1996b]) definiert die in den übrigen ODP-Standards für die Architekturbeschreibung verwendeten Begriffe. Dazu gehört u. a. der Begriff *Objekt (object)*:

„**Object:** A model of an entity. An object is characterized by its behaviour (...) and, dually, by its state (...). (...) An object is encapsulated, i. e. any change in its state can only occur as a result of an internal action or as a result of an interaction (...) with its environment (...).“ ([ISO/IEC JTC 1 1996b], S. 3)

Das Zusammenwirken von Objekten geschieht über Interaktionen (interactions). Die Interaktionen, an denen ein Objekt teilnehmen kann, werden über Schnittstellen (interfaces) zusammengefasst:

Kasten 4.1: Überblick über ISO-Standards zum RM-ODP

ISO/IEC 10746-1: Overview gibt einen Überblick über die RM-ODP-Standards.

Auszug aus [ISO/IEC JTC 1 1998b], S. 8:

„6.2.1 Object modelling

Object modelling provides a formalization of well-established design practices of abstraction and encapsulation. (...)

The object modelling concepts cover:

- Basic modelling concepts – Providing rigorous definitions of a minimum set of concepts (action, object, interaction and interface) that form the basis for ODP system descriptions and are applicable in all viewpoints.
- Specification concepts – Addressing notions such as type and class that are necessary for reasoning about specifications and the relations between specifications, provide general tools for design, and establish requirements on specification languages.
- Structuring concepts – Building on the basic modelling concepts and the specification concepts to address recurrent structures in distributed systems, and cover such concerns as policy, naming, behaviour, dependability and communication.

(...)

ISO/IEC 10746-2: Foundations beschreibt allgemeine Begriffe zur Beschreibung verteilter Informationssysteme; dazu gehören u. a. der Begriff *object* sowie u. a. Begriffe zur Spezifikation von Objekten (z. B. *composition, refinement, type, class*), zur Organisation von Objekten (z. B. *configuration, domain*) oder zur Spezifikation von Eigenschaften von Systemen oder Objekten (z. B. *distribution transparency, contract*).

Auszüge aus [ISO/IEC JTC 1 1996b], S. 3 und 5:

„**8.1 Object:** A model of an entity. An object is characterized by its behaviour (...) and, dually, by its state (...). An object is distinct from any other object. An object is encapsulated, i.e. any change in its state can only occur as a result of an internal action or as a result of an interaction (...) with its environment (...).“

„9.1 Composition

a) (Of objects) – A combination of two or more objects yielding a new object, at a different level of abstraction. The characteristics of the new object are determined by the objects being combined and by the way they are combined. The behaviour of a composite object is the corresponding composition of the behaviour of the component objects.
(...)

„**Action:** Something which happens.

Every action of interest for modelling purposes is associated with at least one object.

The set of actions associated with an object is partitioned into **internal actions** and **interactions**.“
([ISO/IEC JTC 1 1996b], S. 4)

„**Interface:** An abstraction of the behaviour of an object that consists of a subset of the interactions of that object together with a set of constraints on when they may occur.“ ([ISO/IEC JTC 1 1996b], S. 4)

Der Standard definiert viele weitere Begriffe für die detaillierte Beschreibung von Objekten und ihren Interaktionen (siehe auch Kasten „Ergänzende Standards zum RM-ODP ...“).

Das Referenzmodell

Die wichtigsten Ansätze zur Informationssystemarchitekturbetrachtung in den RM-ODP-Standards enthält *ISO/IEC 10746-3* ([ISO/IEC JTC 1 1996a]). Das sind

- die Unterscheidung von verschiedenen Sichtweisen und

Kasten 4.2: Überblick über ISO-Standards zum RM-ODP (Fortsetzung)

ISO/IEC 10746-3: Architecture spezifiziert auf der Basis von ISO/IEC 10746-2 Charakteristika *offener* verteilter Informationssysteme. Zum Standard gehören u. a.

- die Unterscheidung verschiedener Sichtweisen (z. B. *enterprise viewpoint, information viewpoint*),
- den Sichtweisen zugehörige Sprachen,
- die Beschreibung von Funktionen, die in offenen verteilten Informationssystemen implementiert sein müssen (z. B. *thread management, recovery*), und
- die Beschreibung von Transparenzen, durch die bestimmte Systemeigenschaften verborgen werden (z. B. *location transparency, persistence transparency*).

Auszug aus [ISO/IEC JTC 1 1996a], S. 7:

„5 Enterprise language

The enterprise language comprises concepts, rules and structures for the specification of an ODP system from the enterprise viewpoint. (...)

5.1 Concepts

The enterprise language contains the concepts of ITU-T Rec. X.902 | ISO/IEC 10746-2 and those defined here, subject to the rules of 5.2.

5.1.1 Community: A configuration of objects formed to meet an objective. The objective is expressed as a contract which specifies how the objective can be met.

5.1.2 <X> federation: A community of <x> domains.“

ISO/IEC 10746-4: Architectural semantics formalisiert die Bedeutungen der in den Abschnitten 8 und 9 von ISO/IEC 10746-2 definierten Begriffe unter Nutzung verschiedener formaler Beschreibungssprachen (z. B. LOTOS, Z).

Auszug aus [ISO/IEC JTC 1 1998a], S. 5:

„4.1.2.1 Composition

- **of objects** A composite object is an object described through the application of one or more LOTOS combination operators. These include:

- *interleaving operator* (||);
- *parallel composition operators* (|| and |[gate-list|);
- *enabling operator* (>>);
- *disabling operator* ([>);
- *choice operator* (□)

(...)

- die Definition je einer Architektur-Beschreibungssprache für jede Sichtweise.

Die Sichten sind (übersetzt aus [ISO/IEC JTC 1 1996a], S. 4):

Die **Unternehmenssichtweise**: Eine Sichtweise auf das System und seine Umgebung, die den Zweck, den Anwendungsbereich und die Verfahrensweisen des Systems in den Mittelpunkt stellt.

Die **Informationssichtweise**: Eine Sichtweise auf das System und seine Umgebung, die die Semantik der Informationen und ihre Verarbeitung in den Mittelpunkt stellt.

Die **rechnerbezogene Sichtweise**: Eine Sichtweise auf das System und seine Umgebung, die eine Verteilung durch funktionale Zerlegung in Objekte, die über Schnittstellen interagieren, ermöglicht.

Die **Konstruktionssichtweise**: Eine Sichtweise auf das System und seine Umgebung, die

die Mechanismen und Funktionen für eine verteilte Interaktion zwischen Objekten im System in den Mittelpunkt stellt.

Die **technologische Sichtweise**: Eine Sichtweise auf das System und seine Umgebung, die die gewählte(n) Technologie(n) in den Mittelpunkt stellt.

Der Standard definiert außerdem

- Funktionen, die ein verteiltes Informationssystem als offen charakterisieren, und
- Transparenzen, durch die die Verteilung des Informationssystems vor seinen Nutzern und vor Entwicklern anwendungsspezifischer Komponenten verborgen wird.

Die Funktionen sind dabei in die vier Gruppen *Management (management)*, *Koordination (coordination)*, *Verzeichnis (repository)* und *Sicherheit (security)* aufgeteilt. Beispiele sind die Funktion *Objektmanagement (object management)* in der Gruppe *Management* und die Funktion *Handel (trading)* in der Gruppe *Verzeichnis*. Die Transparenzen sind *Zugriffstransparenz (access transparency)*, *Ausfalltransparenz (failure t.)*, *Ortstransparenz (location t.)*, *Migrations-
transparenz (migration t.)*, *Speichertransparenz (persistence t.)*, *Verlagerungstransparenz (re-
location t.)*, *Replikationstransparenz (replication t.)* und *Transaktionstranzparenz (transaction t.)*.

Spezielle Standards auf der Basis des RM-ODP

Im Standard *ISO/IEC 10746-3* selbst sind die Funktionen und Transparenzen NICHT in den Sprachen der verschiedenen Sichtweisen definiert, sondern hauptsächlich in kurzen Beschreibungstexten bezüglich wesentlicher Begriffe und Regeln dargestellt. Eine ausführliche sichtweisenbezogene Definition der Funktionen wird in zusätzlichen Standards gegeben, z. B. *ISO/IEC 13235-1 bis -3* für die in *ISO/IEC 10746-3* vorgesehene Trading Function (Kasten „Ergänzende Standards zum RM-ODP: ...“; [ISO/IEC JTC 1 1995]). Für die Transparenzen existieren keine ausführlichen sichtweisenbezogenen Standards.

Integrationsanforderungen und das RM-ODP

Das RM-ODP beschreibt Grundlagen für den Entwurf verteilter Informationssysteme. Auch wenn einige der im Standard beschriebenen Konzepte prinzipiell auch für konventionelle papierbasierte Informationsverarbeitung anwendbar sind, ist der Standard auf den Entwurf von Softwaresystemen ausgerichtet.

Die mit dem RM-ODP definierten Begriffe bilden eine generische Terminologie für die Beschreibung verteilter Informationssysteme. Sie können für viele rechnerbasierte Integrations-
techniken zur Erfüllung unterschiedlicher Integrationsanforderungen angewendet werden.

Die im RM-ODP beschriebenen speziellen Funktionen, die in einem verteilten System implementiert sein sollten, werden in den genannten ergänzenden Standards spezifiziert. Sie können als spezielle Techniken zur Verwaltung der Verteilung betrachtet werden.

Kasten 4.3: Ergänzende Standards zum RM-ODP: Spezifikation der in ISO/IEC 10746-3 vorgesehenen Trading Function in Standard ISO/IEC 13235-1 (Auszüge)

„6. Enterprise Specification of the Trading Function (...)“

6.1 Communities

The communities are:

6.1.1 trading community - a community of objects established for the purpose of trading and governed by a trading policy. (...)

6.2 Roles

Objects may play the following roles within a trading community:

6.2.1 trader - a role which registers service offers from exporter objects and returns service offers upon request to importer objects according to some criteria. (...)

6.3 Activities

The following activities are relevant to a trading community:

6.3.1 service export - a chain of actions by an exporter object and the trader object which establish and terminate a liaison in which the trader object is permitted to provide the exporter object's service offer to a group of importer objects. (...)

6.4 Policies

The activities of an ODP Trading system are governed by policy. (...)

6.4.1 Export policy

Export policy is a set of rules related to the service export activity (...).

The policy may include, amongst other things:

- an obligation for a service offer to be described in a specific way;
- a prohibition of specified service import activities from discovering the service offer;
- a permission for the service export to be propagated to an interworking trading community. (...)

6.5 Structuring Rules (...)

6.5.2 Transfer Rules

Exporter objects can export offers for services which they provide at their own interfaces, or may export offers for services provided by a distinct service provider object. (...)

7. Information Specification of the Trading Function (...)

7.2 Basic Concepts (...)

7.2.3 Service Description

A service description is at least an interface signature type. It can also include a set of service properties. Service properties contain information about computational aspects (...) as well as describing the technology, engineering, information, and enterprise aspects of the service.

The formal definition of service description is given in the Z schema ServiceDescription. (...)

```

ServiceDescription
signature : InterfaceSignatureType
service_properties : P Property

```

7.3 Invariant Schema The components of the trading system are:

- A set (offers) of service offers which are available for import;
- A set (nodes) of nodes into which these service offers are partitioned;
- A relationship (edges) between nodes to represent the edges of the trading graph, which governs the propagation of searches;
- Sets (edge_properties) of properties which are associated with the edges. (...)

```

TradingSystem
offers : P ServiceOffer
nodes : P Node
partition : ServiceOffer -> Node
edges : Node <-> Node
edge_properties : (Node x Node) -> P Property

dom partition = offers
ran partition <= nodes
dom edges <= ran edges <= nodes
dom edge_properties = edges
forall p, q : ServiceOffer • p.service_offer_identifier =
q.service_offer_identifier <-> p = q (... )

```

8. Computational Specification of the Trading Function (...)

8.2 Trading operation parameter types

8.2.1 Interface identifier type

```

enum InterfaceKindType { CORBA_IFCE, OTHER_IFCE };

union InterfaceIdentifierType switch (InterfaceKindType) {
case CORBA_IFCE:
Object corbalInterfaceId;
case OTHER_IFCE:
any otherInterfaceId;
};

typedef sequence<InterfaceIdentifierType> InterfaceIdListType;

```

The interface identifier type is a union with an CORBA Object reference choice arm, and an extension choice arm of type any. (...)

8.4 Standard Properties

8.4.1 Standard Service Offer Properties

8.4.1.1 Expiry date

“expiry_date”, a service offer property (...) “ ([ISO/IEC JTC 1 1995], S. 5 ff.)

4.2.3 Die Object Management Architecture

Innerhalb der zurück liegenden etwa zehn Jahre³ hat sich die Common Object Request Broker Architecture (CORBA) der Object Management Group (OMG) zu einem bekannten Standard entwickelt ([OMG 2004]). In Vorträgen und Publikationen, sogar in denen der OMG selbst, wird dabei selten auf das der CORBA zugrunde liegende Rahmenwerk, die Object Management Architecture (OMG OMA), hingewiesen (Kasten „Überblick über OMG-Standards . . .“; [OMG 1997]).

Den „Kern“ der OMA bilden

- das OMG Object Model, ein Metamodell für die Beschreibung von objektbasierten Informationssystemen, und
- das OMA Reference Model, ein Architektur-Referenzmodell für die Aufteilung von Komponenten eines Informationssystems in Schnittstellenkategorien und eine Vermittlungskomponente.

Das OMG Object Model und das OMG Component Model

Das OMG Object Model definiert als Metamodell Begriffe für die Beschreibung von objektbasierten Informationssystemen. Der zentrale Begriff ist *Objekt*. Ein Objekt ist nach dem OMG Object Model eine Komponente eines Informationssystems, die anderen Komponenten Dienste, d. h. eine bestimmte Funktionalität, zur Verfügung stellt:

„An *object* is an identifiable, encapsulated entity that provides one or more services that can be requested by a client.“ ([OMG 1997], S. 3-2)

Es werden weitere Begriffe zur genaueren Spezifikation von Objekten definiert. Dazu gehören u. a. *Anfrage (request)*, *Typ (type)*, *Schnittstelle (interface)* und *Operation (operation)*. Eine Schnittstelle ist beispielsweise nach dem OMG Object Model eine Beschreibung einer Menge von Operationen, die bei einem Objekt von anderen Komponenten angefordert werden können, d. h. die anderen Komponenten zur Verfügung gestellt werden:

„An *interface* is a description of a set of possible operations that a client may request of an object. An object *satisfies* an interface if it can be specified as the target object in each potential request described by the interface.“ ([OMG 1997], S. 3-5)

Eine Operation ist eine abgrenzbare Einheit, die einen bestimmten Dienst, d. h. eine bestimmte Funktionalität, bezeichnet:

„An *operation* is an identifiable entity that denotes a service that can be requested.“ ([OMG 1997], S. 3-5)

Der Begriff *Operation* kann dabei als Präzisierung des im RM-ODP verwendeten allgemeinen Interaktionsbegriffes verstanden werden (vgl. Abschnitt 4.2.2).

Das OMG Object Model definiert außerdem Begriffe zur Beschreibung der Implementierung von Objekten. Dazu gehört u. a. der Begriff *Methode*:

„Code that is executed to perform a service is called a *method*. A method is an immutable description of a computation that can be interpreted by an execution engine.“ ([OMG 1997], S. 3-8)

Allein mit den Begriffen des OMG Object Models kann das Zusammenwirken von Komponenten eines Informationssystems nicht oder nur grob beschrieben werden. Hauptsächlich deshalb wurde das OMG Object Model nachträglich durch das OMG Component Model erweitert

³ Die Anfänge der CORBA-Spezifikation liegen möglicherweise noch länger zurück.

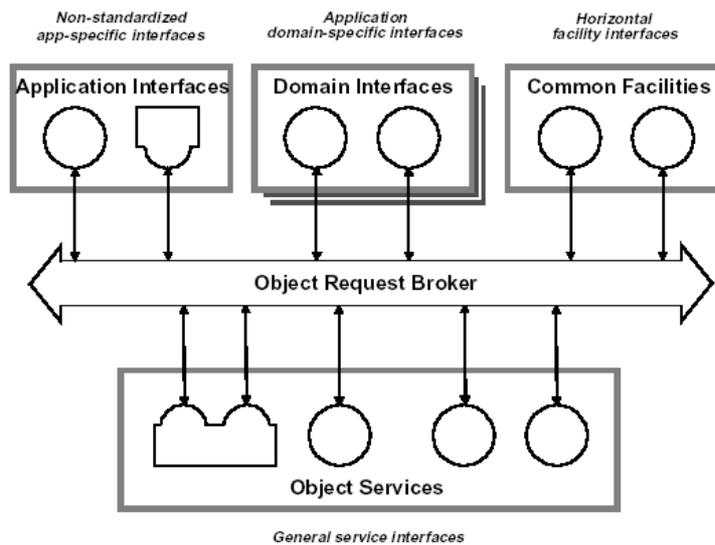


Abbildung 4.4: Das Object Management Architecture Reference Model (Quelle: [OMG 1997], Abb. 4-1)

([OMG 2002a]). Darin wird der Begriff *Komponente* als Spezialisierung und gleichzeitige Erweiterung von *Objekt* eingeführt. Das OMG Component Model führt mit dieser Erweiterung zusätzliche Begriffe ein, mit denen hauptsächlich das Zusammenwirken von Komponenten spezifiziert werden kann. Beispiele sind die Begriffe *Verbindung (connection)*, *Verbindungspunkte (receptacles)*, *Ereignistyp (event type)*, *Ereignisquelle (event source)*, *Ereignissenke (event sink)*, *Komponentenheimat (component home)* oder *Komponentenkonfiguration (component configuration)*. Zu den meisten Begriffen des OMG Component Models werden Schnittstellen spezifiziert, z. B. für das Erzeugen von Objektinstanzen, das Empfangen von Ereignissen usw. Damit gehört das OMG Component Model zu den Spezifikationen von gemeinsamen Diensten im Sinne des nachfolgend vorgestellten OMA Reference Model.

Für die formale Spezifikation von Objekten bzw. Komponenten auf der Basis der im OMG Object Model und im OMG Component Model definierten Begriffe wurde die Interface Definition Language (OMG IDL) entwickelt. Sie ist in Kapitel 3 der CORBA-Spezifikation definiert ([OMG 2004], S. 3-1 - 3-74).

Das Object Management Architecture Reference Model

Am Anfang dieses Abschnitts wurde neben dem OMG Object Model als Metamodell auch das Object Management Architecture Reference Model (OMA Reference Model) als Architektur-Referenzmodell eingeführt (Abbildung 4.4). Sie beschreibt

- eine zentrale Komponente, den **Object Request Broker (ORB)**, der die Kommunikation von Objekten und ihren „Auftraggebern“ ermöglicht und
- vier Kategorien, in die Objekte bzw. ihre Schnittstellen auf der Basis der Dienste, die durch die von ihnen bereitgestellten Operationen repräsentiert werden, eingeteilt werden können.

Die Kategorien werden folgendermaßen abgegrenzt (übersetzt aus [OMG 1997], Abschnitt 4.1.2, S. 4-1 - 4-2, siehe auch die ausführlicheren Beschreibungen in [OMG 1997], Abschnit-

te 4.1.5-4.1.7, S. 4-3 - 4-4):

Objektdienste (object services) sind Schnittstellen für allgemeine Dienste, die in voraussichtlich jedem Programm, das auf verteilten Objekten basiert, genutzt werden.

Gemeinsame Dienste (common facilities) sind Schnittstellen für horizontale benutzerorientierte Dienste, die in den meisten Anwendungsbereichen benötigt werden.

Anwendungsbereichsschnittstellen (domain interfaces) sind spezifische Schnittstellen für einen bestimmten Anwendungsbereich.

Anwendungssystemschnittstellen (application interfaces) sind nicht-standardisierte, anwendungssystemspezifische Schnittstellen.

Spezielle Standards auf der Basis der OMA

Die zentrale Komponente der OMA, der ORB, wird in einem umfangreichen Standard, der Common Object Request Broker Architecture (CORBA), spezifiziert ([OMG 2004]). Im CORBA-Standard wird u. a. auch die Interface Definition Language (IDL) für die Spezifikation der Schnittstellen des ORB aber auch für weitere Schnittstellenspezifikationen in ergänzenden Standards der OMG definiert.

Im Gegensatz zu den übrigen Dienst- bzw. Schnittstellenkategorien wird für die Objektdienste bereits mit dem OMA Reference Model nicht nur eine allgemeine Abgrenzung vorgenommen, sondern genauer beschrieben, welche Funktionalität über welche Objektdienste zur Verfügung gestellt wird ([OMG 1997], S. 4-7 - 4-12). Eine ausführliche Spezifikation der Objektdienste wird jedoch in weiteren Dokumenten der OMG vorgenommen. Beispiele für Objektdienste sind

- der Trading Object Service für das Anbieten und Finden von Diensten ([OMG 2000f]),
- der Persistent State Service für das Speichern von Objekten ([OMG 2002c]) oder
- der Security Service für das Schützen von Informationen ([OMG 2002d]).

Auch für die Kategorie der gemeinsamen Dienste und insbesondere für die Kategorie der Anwendungsbereichsschnittstellen wurden von der OMG Spezifikationsdokumente für einzelne Dienste herausgegeben. Beispiele für gemeinsame Dienste sind

- die Meta-Object Facility ([OMG 2000d]; siehe auch Kasten „Überblick über OMG-Standards zur ...“),
- die Internationalization, Time Operations, and Related Facilities ([OMG 2000c]) und
- die Mobile Agent Facility ([OMG 2000e]).

Beispiele für Anwendungsbereichsschnittstellen sind

- für den Anwendungsbereich *Transportwesen*
 - die Air Traffic Control Specification ([OMG 2000a]),
- für den Anwendungsbereich *Finanzen*
 - die Currency Specification ([OMG 2000b]),
- für den Anwendungsbereich *Biowissenschaften*
 - die Biomolecular Sequence Analysis Specification ([OMG 2001a]),
 - die Gene Expression Specification ([OMG 2003a]),
 - die Genomic Maps Specification ([OMG 2002b]) und
- für den Anwendungsbereich *Gesundheitswesen*

MDA-Sichtweisen	RM-ODP-Sichtweisen
rechnerunabhängige Sichtweise	Unternehmenssichtweise Informationssichtweise
plattformunabhängige Sichtweise	Unternehmenssichtweise Informationssichtweise rechnerbezogene Sichtweise
plattformspezifische Sichtweise	(Konstruktionssichtweise)* (technologische Sichtweise)*

* Die Zuordnung der Konstruktionssichtweise und der technologischen Sichtweise des RM-ODP zur plattformspezifischen Sichtweise des MDA Guide ist im MDA Guide nicht dokumentiert. Sie wurde vom Autor dieser Arbeit ergänzt.

Tabelle 4.1: Zuordnung von RM-ODP-Sichtweisen zu MDA-Sichtweisen (Quelle: [OMG 2003b], S. 3-1 - 3-2)

- die Clinical Observations Access Service Specification ([OMG 2001b]),
- die Person Identification Service Specification ([OMG 2001c]) und
- die Resource Access Decision Facility Specification ([OMG 2001d]).

Für Anwendungsschnittstellen wurden von der OMG keine Standards herausgegeben. Zu dieser Kategorie gehören die Schnittstellen, die nicht in einem Standard spezifiziert werden können, sondern die auf bestimmte konkrete Anwendungsszenarios zugeschnitten sind.

Model Driven Architecture

Die bisher in diesem Abschnitt beschriebenen Standards sind unmittelbar auf die OMA bezogen. Sie werden ergänzt durch den Model Driven Architecture Guide (MDA Guide), der eine Anleitung, d. h. ein Vorgehens-Referenzmodell, für die Verwendung von Modellen bei der Softwareentwicklung enthält.

Der MDA Guide definiert zuerst grundlegende Begriffe im Zusammenhang mit der Modellierung, z. B. *Modell (model)*, *Architektur (architecture)*, *Sichtweise (viewpoint)*, *Sicht (view)* oder *Plattform (platform)*. Die Definition für *Sichtweise* ist beispielsweise:

„A *viewpoint* on a system is a technique for abstraction using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within that system. Here ‘abstraction’ is used to mean the process of suppressing selected detail to establish a simplified model.“ ([OMG 2003b], S. 2-3)

Hier sei der Begriff *Plattform* hervorgehoben, der wesentliches sprachliches Hilfsmittel für die Abgrenzung von Abstraktionsschichten bei der Aufteilung von Informationssystemen in Komponenten ist:

„A *platform* is a set of subsystems and technologies that provide a coherent set of functionality through interfaces and specified usage patterns, which any application supported by that platform can use without concern for the details of how the functionality provided by the platform is implemented.“ ([OMG 2003b], S. 2-3)

Der Plattformbegriff wird u. a. auch im TOGAF Technical Reference Model (siehe Abschnitt 4.2.5) verwendet.

Unter Nutzung des Plattformbegriffes beschreibt der MDA Guide drei Sichtweisen auf Systeme (übersetzt aus [OMG 2003b], S. 2-5):

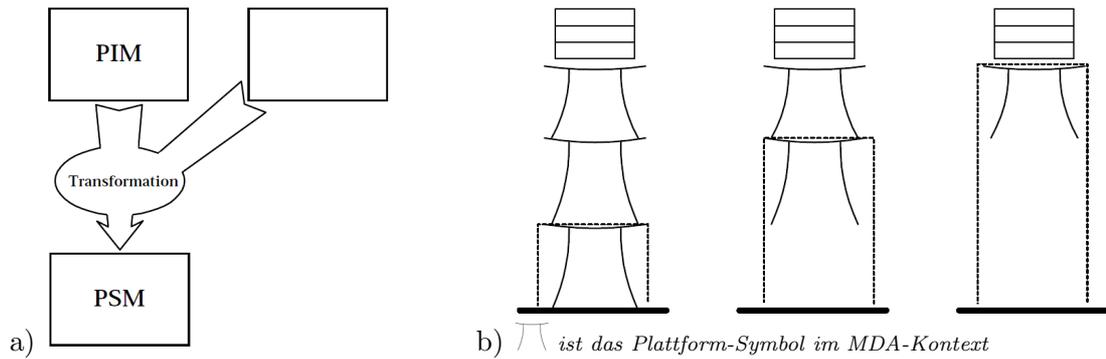


Abbildung 4.5: Das Model Driven Architecture Pattern (a) (Quelle: [OMG 2003b], Abb. 2-2) und generische Unterscheidung von Sichtweisen auf Plattformen (b) (Quelle: [OMG 2003b], Abb. 6-6)

Die **rechnerunabhängige Sichtweise (computation independent viewpoint)** ist auf die Umgebung des betrachteten Systems und auf die Anforderungen an das System gerichtet; die Details der Struktur und der Prozesse des Systems sind verborgen oder sogar unbestimmt.

Die **plattformunabhängige Sichtweise (platform independent viewpoint)** ist auf die Arbeitsweise des betrachteten Systems gerichtet, wobei die Details, die bestimmte Plattformen betreffen, verborgen bleiben. Eine plattformunabhängige Sicht zeigt den Teil der vollständigen Spezifikation, der sich zwischen verschiedenen Plattformen nicht ändert.

Die **plattformspezifische Sichtweise (platform specific viewpoint)** auf das betrachtete System kombiniert die plattformunabhängige Sichtweise mit der zusätzlichen Berücksichtigung der Details der Verwendung einer bestimmten Plattform durch das System.

Modelle werden als Sichten auf Systeme aufgefasst. Entsprechend den Sichtweisen führt der MDA Guide die Kategorien

- rechnerunabhängige Modelle (computation independent models, CIM),
- plattformunabhängige Modelle (platform independent models, PIM) und
- plattformspezifische Modelle (platform specific models, PSM)

ein, ergänzt um eine zusätzliche Kategorie für

- Plattformmodelle (platform models).

Die Vorgehens-Beschreibung im MDA Guide ordnet den MDA-Sichtweisen Sichtweisen aus dem RM-ODP zu (Tabelle 4.1, vgl. Abschnitt 4.2.2; [OMG 2003b], S. 3-1 - 3-2). Den größten Teil des MDA Guide nehmen Beschreibungen von Transformationen von Modellen und zugrunde liegenden Metamodellen ein. Der Schwerpunkt liegt dabei auf der Transformation von plattformunabhängigen Modellen in plattformabhängige Modelle. In diesem Zusammenhang wird das MDA-Muster (MDA pattern) eingeführt (Abbildung 4.5a).

An dieser Stelle sei auf die Ähnlichkeit der generischen Plattformbetrachtung im MDA Guide ([OMG 2003b], S. 6-1 - 6-6) und des generischen Schichtenkonzeptes, das dem OSI-Referenzmodell zugrunde liegt (vgl. Abschnitt 4.2.1), hingewiesen. Dazu wird im MDA Guide das Prinzip beschrieben, Plattformen aus verschiedenen „übereinander liegenden“ Sichtweisen zu betrachten, wobei die Details bzgl. der unteren Sichtweisen jeweils den darüber liegenden Sichtweisen verborgen bleiben (Abbildung 4.5b).

Kasten 4.4: Überblick über OMG-Standards zur Architekturentwicklung

Die **Object Management Architecture (OMA)** ist der grundlegende Rahmen für die Entwicklung verteilter objektbasierter Informationssysteme ([OMG 1997]). Sie enthält

- das **OMG Object Model**, ein Metamodell für die Beschreibung von objektbasierten Informationssystemen, und
- das **Object Management Architecture Reference Model (OMA Reference Model)**, ein Architektur-Referenzmodell mit
 - der zentralen Vermittlungskomponente, dem **Object Request Broker (ORB)**, und
 - den vier Schnittstellenkategorien *Objektdienste*, *gemeinsame Dienste*, *Anwendungsbereichsschnittstellen* und *Anwendungssystemschnittstellen*.

Die **Common Object Request Broker Architecture (CORBA)** spezifiziert die zentrale Komponente der OMA, den ORB ([OMG 2004]). Im CORBA-Standard wird auch die **Interface Definition Language (IDL)** für die Spezifikation von Schnittstellen definiert. Die IDL wird für die Spezifikation des ORB sowie in allen weiteren OMG-Dokumenten, die Schnittstellen spezifizieren, verwendet, z. B. für die Objektdienste und die gemeinsamen Dienste.

Für die Schnittstellenkategorien *Objektdienste*, *gemeinsame Dienste* und *Anwendungsbereichsschnittstellen* der OMA wurden verschiedene Spezifikationsdokumente für einzelne Dienste herausgegeben. Beispiele sind:

- der **Trading Object Service** für das Anbieten und finden von Diensten ([OMG 2000f]; Kategorie *Objektdienste*).
- die **Internationalization, Time Operations, and Related Facilities** für das Anbieten und finden von Diensten ([OMG 2000c]; Kategorie *gemeinsame Dienste*).
- die **Clinical Observations Access Service Specification** für den Zugriff auf Befunde und andere Resultate medizinischer Untersuchungen ([OMG 2001b]; Kategorie *Anwendungsbereichsschnittstellen*, Anwendungsbereich *Gesundheitswesen*).

Der **Model Driven Architecture Guide (MDA Guide)** ist, unabhängig von der OMA, eine Anleitung für die Verwendung von Modellen bei der Softwareentwicklung ([OMG 2003b]). Darin werden grundlegende Begriffe für die Modellierung, wie z. B. *Modell (model)*, *Architektur (architecture)*, *Sichtweise (viewpoint)*, *Sicht (view)* oder *Plattform (platform)*, definiert.

Der MDA Guide beschreibt außerdem drei Sichtweisen auf Systeme:

- die rechnerunabhängige Sichtweise (*computation independent viewpoint*),
- die plattformunabhängige Sichtweise (*platform independent viewpoint*) und
- die plattformspezifische Sichtweise (*platform specific viewpoint*).

Der größte Teil des MDA Guide beschreibt Modelltransformationen, hauptsächlich für Transformationen plattformunabhängiger Modelle in plattformspezifische Modelle.

OMG-Standards zur Umsetzung des MDA Guide sind ([OMG 2003b], S. 7-1 - 7-2)

- die **Unified Modeling Language (UML)**, die sich auch außerhalb der OMG-Standards als universelle Modellierungssprache für Informationssystemarchitekturen, Softwareentwürfe und viele weitere andere Aspekte der Informationsverarbeitung durchgesetzt hat ([OMG 2003c]),
- die umfangreiche **Meta-Object Facility (MOF) Technology** zur Erstellung und Verwaltung von Modellen und Metamodellen und zur Wahrung der Konsistenz zwischen Modellen und Metamodellen ([OMG 2000d]),
- die **Profile** für die Spezifikation neuer Modellierungssprachen auf der Basis der UML ([OMG 2003c], Abschnitt 2.6 „Extension Mechanisms“) und
- Spezifikationen verschiedener **Plattformen**, z. B. Realtime CORBA, CORBA Components oder die anwendungsbereichsspezifischen Dienste für das Gesundheitswesen.

Die MOF ist gleichzeitig ein Beispiel für Common Facilities entsprechend der OMA.

Hinweis: Die Reihenfolge der Standards entspricht nicht in jedem Fall der zeitlichen Reihenfolge ihrer Veröffentlichung.

Integrationsanforderungen und die OMA

Die OMA wurde, wie das RM-ODP, für den Entwurf verteilter Informationssysteme entwickelt. Sie ist jedoch spezieller auf die Bereitstellung von Diensten durch Komponenten ausgerichtet. Mit der CORBA wird eine Komponente für die Vermittlung von Dienstanforderungen an Komponenten und die Bereitstellung der Ergebnisse der Dienstanforderungen spezifiziert.

Die OMA und insbesondere die CORBA spezifizieren folglich eine Technik für die funktionale Integration. Wegen der in Abschnitt 2.2.3 beschriebenen Dualität von Datenintegration und funktionaler Integration kann auf der Basis der OMA auch Datenintegration hergestellt werden. Zusätzlich können unter Nutzung von Implementierungen ergänzender Standards weitere Integrationsanforderungen erfüllt werden. Ein Beispiel ist die Resource Access Decision Facility, die für die Realisierung von Zugriffsintegration angewendet werden kann.

4.2.4 Das Healthcare Information Framework

Ein Rahmenwerk, das für den Anwendungsbereich der Informationsverarbeitung im Gesundheitswesen entwickelt wurde, ist das Healthcare Information Framework (HIF, Standard *EN 12443: Healthcare Information Framework*) des Europäischen Komitee für Normung (Comité Européen de Normalisation, CEN).

Das HIF ist ein Architektur-Referenzmodell. Ähnlich dem TOGAF TRM (siehe Abschnitt 4.2.5) sieht das HIF eine Unterscheidung dreier Ebenen bei der Betrachtung von Informationssystemarchitekturen vor (Abbildung 4.6). Die Ebenen werden folgendermaßen voneinander abgegrenzt (übersetzt aus [CEN TC251 1997], S. 10):

- (Die) **Gesundheitswesen-Anwendungssystemebene** modelliert die Datenflüsse und Funktionalitäten, die benötigt werden, um Prozesse im Gesundheitswesen zu unterstützen.
- (Die) **Gesundheitswesen-Middlewareebene** modelliert gemeinsam genutzte Dienste, die benötigt werden, um die Anwendungssystemebene zu unterstützen.
- (Die) **Gesundheitswesen-Datenübermittlungsebene** modelliert die technologische Infrastruktur, die der Middlewareebene Dienste zur Verfügung stellt.

Spezielle Standards auf der Basis des HIF

Das CEN sieht für jede der Ebenen des HIF spezielle Standards vor. Bisher existiert mit der Healthcare Information Systems Architecture (HISA), Teil 1 (CEN-Standard *EN 12967-1: Healthcare Information Systems Architecture Part 1: Healthcare Middleware Layer*) nur für die Middlewareebene eine ausführliche Spezifikation ([CEN TC251 1997]).

In der HISA werden zwei Dienstklassen unterschieden:

- gemeinsame Dienste des Gesundheitswesens (Healthcare Common Services) und
- allgemeine gemeinsame Dienste (Generic Common Services).

Die gemeinsamen Dienste des Gesundheitswesens können mit den anwendungsbereichsspezifischen Diensten der OMA für den Anwendungsbereich des Gesundheitswesens verglichen werden, die allgemeinen gemeinsamen Dienste mit den gemeinsamen Diensten der OMA.

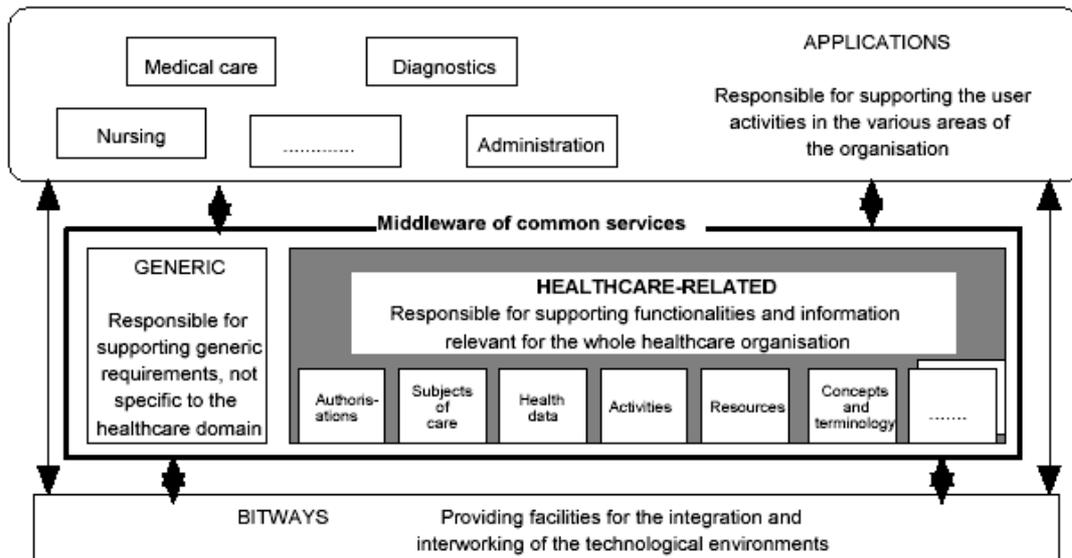


Abbildung 4.6: Die Schichten der Architektur von Informationssystemen im Gesundheitswesen (Quelle: [CEN TC251 1997], Abbildung 1)

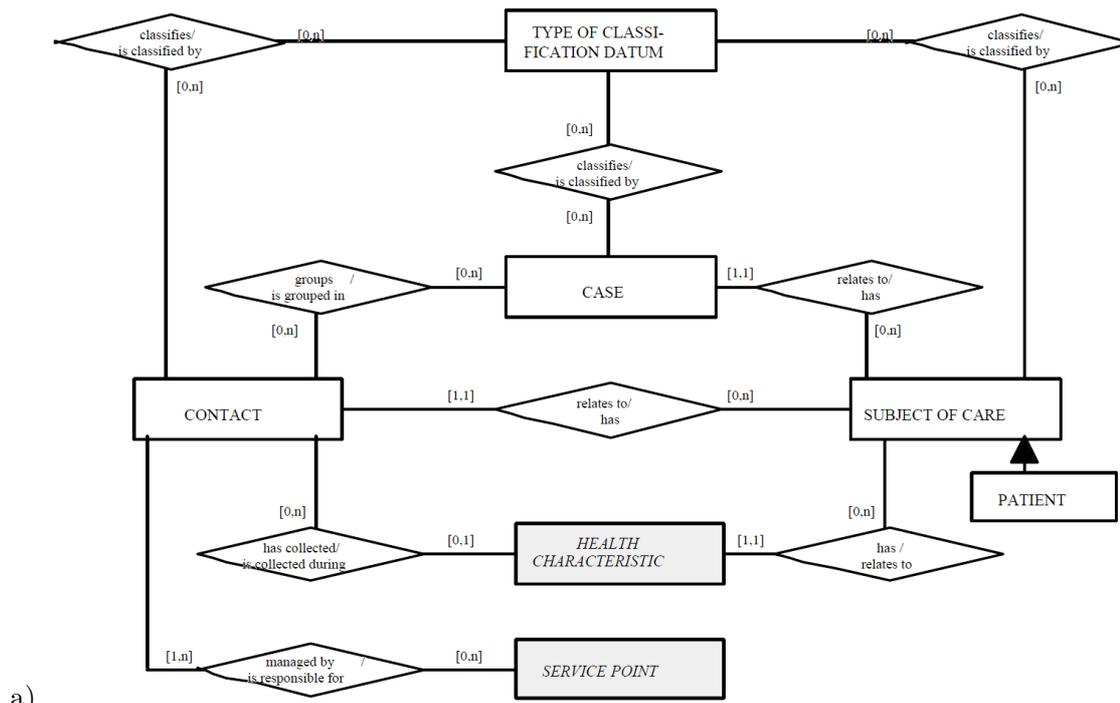
Zentraler Inhalt des Standards sind informationale und funktionale Spezifikationen⁴ von sechs Dienstkategorien der gemeinsamen Dienste des Gesundheitswesens ([CEN TC251 1997], S. 13):

- Dienste zum Gegenstand der Gesundheitsversorgung (Subject of Care Healthcare Common Services),
- Dienste zu Gesundheits-Charakteristika (Health Characteristic Healthcare Common Services),
- Dienste zu Aktivitäten (Activity Healthcare Common Services),
- Dienste zu Ressourcen (Resource Healthcare Common Services),
- Dienste zur Autorisierung (Authorisation Healthcare Common Services) und
- Dienste zu Begriffen (Concept Healthcare Common Services).

Einige Bezeichnungen der Dienstkategorien lassen nicht unmittelbar erkennen, für die Verwaltung welcher Informationen sie spezifiziert wurden: Gegenstand der Versorgung sind Patienten; Gesundheits-Charakteristika sind Informationen zum Gesundheitszustand, z. B. Vitalwerte und Befunde; Aktivitäten sind hauptsächlich Untersuchungen und Behandlungen; Ressourcen sind z. B. Materialien oder Medikamente. Dienste zur Autorisierung verwalten Benutzerkennungen und Zugriffsrechte; Dienste zu Begriffen stellen beispielsweise als Terminologieserver Begriffssysteme zur Verfügung.

Für jede Dienstgruppe enthält die HISA ein Informationsmodell der verwalteten Informationen. Sie enthält weiterhin eine für alle Dienstkategorien gleiche funktionale Spezifikation (Abbildung 4.7). Die Informationsmodelle können als Informations-Referenzmodelle verwendet werden. Die funktionale Spezifikation kann als funktionale Referenz-Spezifikation für Dienste in Informationssystemen im Gesundheitswesen verwendet werden.

⁴ Die informationale Spezifikation ist die Spezifikation der zu verarbeitenden Informationen; die funktionale Spezifikation ist die Spezifikation der bereitzustellenden Funktionalität.



For each entity defined in the structural view of the various HCS, the healthcare middleware layer shall provide the rest of the healthcare information system with a set of services allowing, at least, the following functionalities:

- retrieval of a list of instances of the entity, specified on the basis of different selection criteria defined through conditions on the attributes of the entity or on the relationships the entity has with other entities;
 - maintenance of one instance of the entity, i.e.
 - retrieval of the full set of information of one instance;
 - entering of a new instance;
 - modification of the attributes of one already existing instance;
 - deletion of one instance;
 - retrieval of the list of the instances related to one instance of the entity; for each relationship connected to the entity;
- b)
- maintenance of one already existing instance for each relationship connected to the entity.

Abbildung 4.7: Informationsmodell für die Subject of Care Healthcare Common Services (a) und funktionale Spezifikation (b) (Quelle: [CEN TC251 1997], S. 15)

Integrationsanforderungen und das HIF

Der einzige bisher zum HIF veröffentlichte spezielle Standard, die HISA, verdeutlicht die Fokussierung der HIF-Entwicklung auf dienstbasierte Architekturen. Damit legen das HIF und die HISA, wie auch die OMA und die zugehörigen Dienstspezifikationen, Grundlagen für funktionale Integration.

Im Gegensatz zur OMA existieren zum HIF keine Spezifikationen zur Vermittlung von Dienstanforderungen. Es wird also keine Integrationstechnik im engeren Sinne beschrieben. Die

in der HISA spezifizierten Dienstgruppen beschreiben Informationen und zugehörige Funktionalität für die Anwendung im Gesundheitswesen. Implementierungen der Dienste zur Automatisierung können zur Herstellung von Zugriffsintegration beitragen, Implementierungen der Dienste zu Begriffen können zur Herstellung von semantischer Integration beitragen.

4.2.5 The Open Group Architectural Framework, Version 7

The Open Group Architectural Framework (TOGAF) wurde von der Open Group⁵ auf der Basis des Technical Architecture Framework for Information Management (TAFIM) des US-amerikanischen Verteidigungsministeriums erstellt (Abbildung 4.8; [DoD DISA 1996], [THE OPEN GROUP 2001]). Hier wird zunächst die Version 7 vorgestellt, die technische Aspekte in den Vordergrund stellt. In Abschnitt 4.4.4 wird die Version 8 (genauer: 8.1) vorgestellt, die zusätzlich technikenunabhängige unternehmensbezogene Aspekte berücksichtigt⁶. Seine zentralen Inhalte sind

- die TOGAF Architecture Development Method (TOGAF ADM) und
- die TOGAF Foundation Architecture.

⁵ Die (The) Open Group ist eine internationale Organisation, die die Open Software Foundation (OSF) und die X/Open-Organisation zusammenfasst.

⁶ Beide TOGAF-Versionen wurden zur Zeit der Erstellung dieser Arbeit parallel von der Open Group publiziert. Sofern in bestimmten Projekten unternehmensbezogene Aspekte ausgeblendet werden können, kann die Anwendung der Version 7 anstelle der Version 8 effizienter sein.

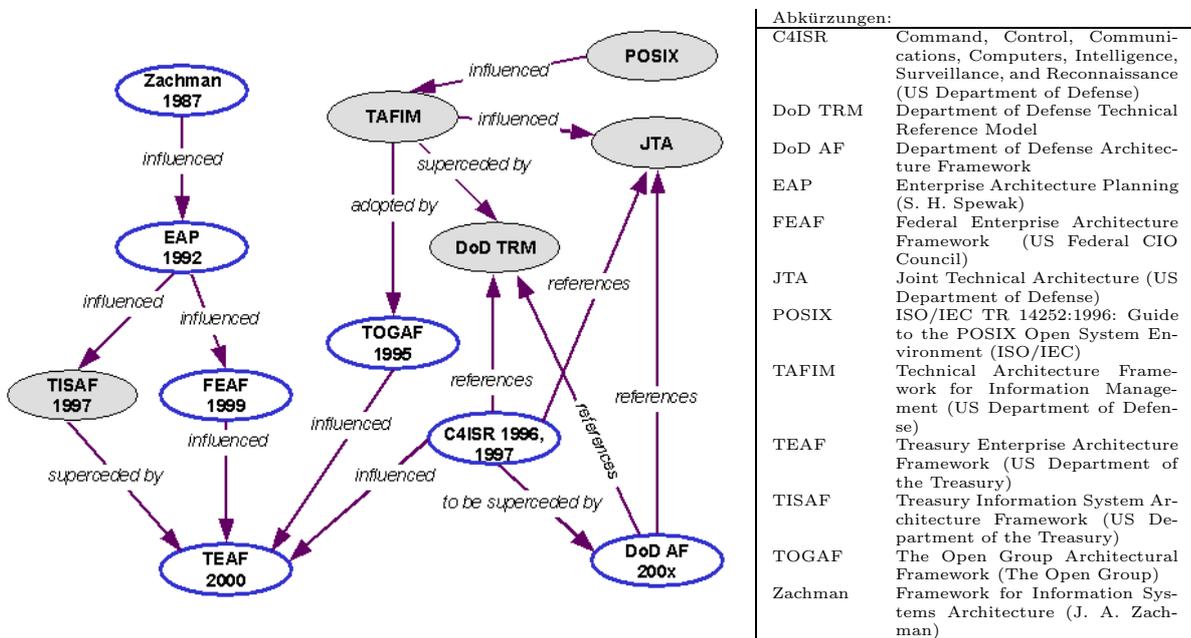


Abbildung 4.8: TOGAF und seine Beziehungen zu anderen US-amerikanischen Rahmenwerken (Quelle: unbekannt)

Die TOGAF Architecture Development Method

Die TOGAF Architecture Development Method (TOGAF ADM) ist ein Vorgehens-Referenzmodell (siehe Kästen „Grundbegriffe (4)“ und „Grundbegriffe (5)“). Sie beschreibt Phasen für das Vorgehen zur Entwicklung und Weiterentwicklung von Informationssystemarchitekturen. Die Phasen und ihre Ziele sind (übersetzt aus [THE OPEN GROUP 2001], Part II):

Phase A: Einleitung und Rahmenbedingungen

Ziel: Ziel dieser Phase ist es, basierend auf den Grundsätzen, Unternehmenszielen und strategischen Faktoren des Unternehmens die relevanten Unternehmensanforderungen, die für diese Architekturentwicklung gelten, und eine Vision der Architektur, die eine Antwort auf die Anforderungen zeigt, zu definieren.

Phase B: Basisbeschreibung

Ziel: Ziel dieser Phase ist es, auf hoher Ebene eine Beschreibung der Charakteristika des vorhandenen Systems zu erstellen. Das ist notwendig, da die Beschreibung den Ausgangspunkt der Architekturentwicklung dokumentiert und Interoperabilitätsprobleme aufzählt, die die endgültige Architektur berücksichtigen muss.

Phase C: Zielarchitektur

Ziel: Ziel dieser Phase ist es, die Zielarchitektur zu ermitteln, die die Grundlage der folgenden Implementierungsarbeiten bildet.

Phase D: Möglichkeiten und Lösungen

Ziel: Diese Phase ermittelt die Kenngrößen der vorzunehmenden Änderungen, die Hauptphasen des Änderungsprozesses und die Projekte auf höchster Ebene, die durchgeführt werden müssen, um vom Ist- zum Zielzustand zu gelangen. Sie bildet die Basis des Implementierungsplans, der für den Wechsel zur Zielarchitektur notwendig ist.

Phase E: Migrationsplanung

Ziel: Ziel dieser Phase ist es, die verschiedenen Implementierungsprojekte nach ihrer Priorität zu ordnen. Zu den Aktivitäten gehören die Bewertung der Abhängigkeiten, Kosten und Nutzen der verschiedenen Migrationsprojekte. Die nach Priorität geordnete Projektliste ist weitere Grundlage für den Implementierungsplan.

Phase F: Implementierung

Ziel: Ziel dieser Phase ist es, Empfehlungen für jedes Implementierungsprojekt zu formulieren und eine Architekturvereinbarung zu erstellen, die die Systemimplementierung und -bereitstellung regelt. Dann wird in dieser Phase das System implementiert und bereitgestellt.

Phase G: Architekturwartung

Ziel: Ziel dieser Phase ist es, ein Wartungsverfahren für die neue Basis, die mit Beendigung der Implementierung erreicht wird, festzulegen. Dieses Verfahren wird üblicherweise für eine kontinuierliche Überwachung von – unter anderem – neuen technologischen Entwicklungen und Änderungen im Unternehmensumfeld sowie für die Bestimmung, ob ein neuer Architekturentwicklungszyklus zu initiieren ist, sorgen.

Jede Phase der TOGAF ADM ist unterteilt in Arbeitsschritte. Phase C besteht beispielsweise aus acht Schritten (übersetzt aus [THE OPEN GROUP 2001], Part II, Phase C, siehe Abbildung 4.9):

1. Repräsentation der Basisbeschreibung unter Anwendung der TOGAF Foundation Architecture, um einen gemeinsamen Ausgangspunkt anzugeben,
2. Prüfung verschiedener Standpunkte bezüglich der Architektur, um sicherzustellen, dass die Anforderungen aller Interessengruppen an das geforderte System berücksichtigt werden,
3. Auswahl eines Architekturmodells auf hoher Ebene,
4. Festlegung einer Menge von Kriterien für die Dienstausswahl,
5. Auswahl der benötigten Dienste,
6. Prüfung, ob die Unternehmensziele eingehalten werden,
7. detaillierte Definition der Architektur und
8. Durchführung einer Diskrepanzanalyse, um jede fehlende Funktionalität zu ermitteln.

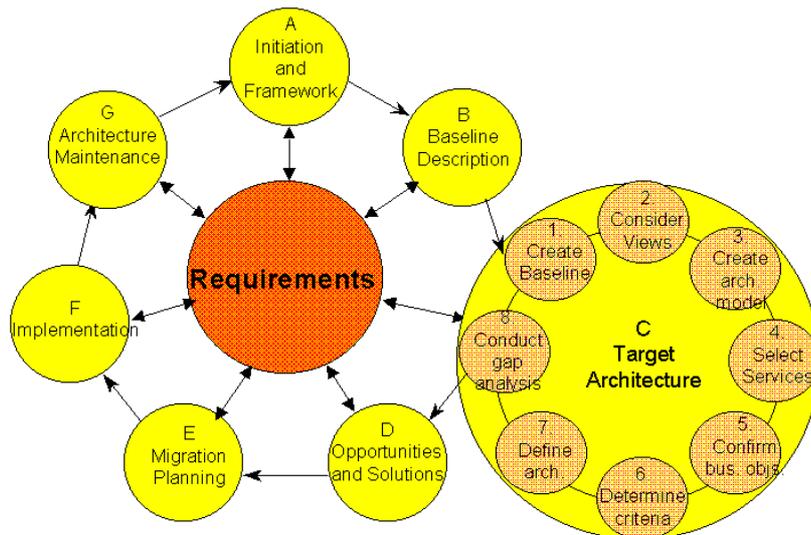


Abbildung 4.9: Die Phasen der TOGAF Architecture Development Method und die Schritte der Phase C (Quelle: [THE OPEN GROUP 2001], Part II, Introduction, Figure 2)

Das TOGAF Technical Reference Model

Die bereits vorgestellte TOGAF ADM ist, wie oben beschrieben, ein Vorgehens-Referenzmodell. TOGAF enthält weiterhin ein Architektur-Referenzmodell,

- das TOGAF Technical Reference Model (TOGAF TRM).

Es wird ergänzt um Listen von Standards für die im TOGAF TRM angegebenen Dienstkategorien:

- die TOGAF Standards Information Base (TOGAF SIB).

TOGAF TRM und TOGAF SIB werden unter der Überschrift *TOGAF Foundation Architecture* zusammengefasst ([THE OPEN GROUP 2001], Part III). Die oben beschriebene TOGAF ADM kann als Vorgehens-Referenzmodell für die Überführung des TOGAF TRM in ein spezifisches Architekturmodell betrachtet werden.

Das TOGAF TRM sieht für die Informationssystembeschreibung eine Unterteilung in drei Einheiten vor (Abbildung 4.10):

- die Anwendungssysteme, unterteilt in geschäftsspezifische und infrastrukturelle Anwendungssysteme,
- die Anwendungsplattform⁷ mit verschiedenen Dienstkategorien und
- die Kommunikationsinfrastruktur.

Zu den drei Einheiten, hauptsächlich zu den Dienstkategorien der Anwendungsplattform, werden Erklärungstexte, jedoch keine formalen Beschreibungen, angegeben.

Die Unterteilung in die drei Einheiten ähnelt den Ebenen des Healthcare Information Framework; die Dienstkategorien ähneln den Dienstkategorien der Healthcare Information Systems Architecture (siehe Abschnitt 4.2.4).

⁷ Vgl. Plattformbegriff in Abschnitt 4.2.3.

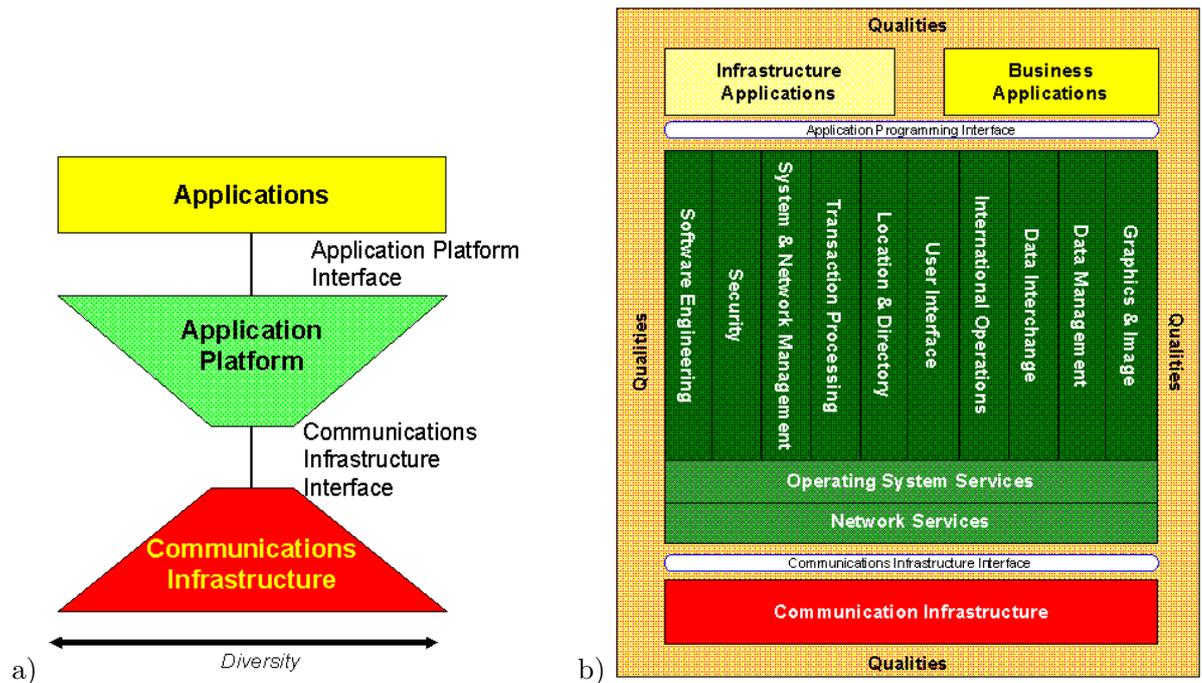


Abbildung 4.10: Einfache (a) und detailliertere (b) Grafik zum TOGAF TRM (Quellen: [THE OPEN GROUP 2001], Part III, High-level Breakdown, Figure 1 und Part III, TRM in Detail, Figure 1)

Die TOGAF Standards Information Base – Standards für die Anwendung des TOGAF TRM

Im Gegensatz zu den bisher vorgestellten Rahmenwerken ist TOGAF selbst nicht Grundlage für die Entwicklung spezieller Standards für Integrationstechniken.

Von der Open Group wird für die Anwendung des TOGAF TRM eine sehr umfassende Sammlung von Standards, die TOGAF Standards Information Base. Sie enthält Listen von und Verweise auf Standards für Dienste der einzelnen im TOGAF TRM genannten Dienstkategorien. Dabei werden nicht nur Dienstspezifikationen im Sinne der OMA-basierten Dienste angegeben, sondern auch Protokolle für den Datenaustausch, Dateiformatspezifikationen und viele weitere Standards. Die Standarddokumente selbst sind jedoch nicht unmittelbar in der SIB enthalten. Beispiele für angegebene Standards sind

- HTML 4.0 Reference Specification,
- ISO/IEC 10918: Information technology – Digital compression and coding of continuous-tone still images (JPEG-Spezifikation) oder
- X/Open C525: Protocols for X/Open Interworking: XNFS - Version 3.

Die ersten beiden gehören zur Kategorie der Datenaustauschdienste (data interchange services), X/Open C525 gehört zur Kategorie der Netzwerkdienste (network services).

Die TOGAF SIB enthält u. a. auch die in Abschnitt 4.2.1 genannten Standards bzw. Protokolle IEEE 802.X, IP, TCP, ASCII, TIFF, MPEG und FTP sowie die in Abschnitt 4.2.3 beschriebene CORBA.

Integrationsanforderungen und TOGAF

TOGAF wurde entwickelt, um Projekte zur Entwicklung bzw. Weiterentwicklung von Architekturen zu unterstützen. Damit kann TOGAF indirekt über die Unterstützung von Integrationsprojekten, die immer auch die Architektur betreffen, zur Erfüllung von Integrationsanforderungen beitragen.

Die in der TOGAF SIB enthaltenen Standards können prinzipiell zur Herstellung aller Integrationsanforderungen beitragen.

4.3 Weitere Standards für Integrationstechniken

Neben den im vorhergehenden Abschnitt vorgestellten Standards für Integrationstechniken und zugrunde liegenden Rahmenwerken existieren viele weitere, die nicht unmittelbar mit den vorgestellten Rahmenwerken in Beziehung stehen. Sie können jedoch für die Erfüllung mancher der in Kapitel 2 beschriebenen Integrationsanforderungen genutzt werden. Einige Beispiele werden im Folgenden kurz vorgestellt.

Das Distributed Component Object Model

Das Distributed Component Object Model (DCOM) ist ein von der Firma Microsoft entwickelter Ansatz für die Beschreibung und Implementierung verteilter Komponenten, die Dienste bereitstellen. Er basiert auf dem Component Object Model (COM). Das COM und seine Erweiterung DCOM sind vergleichbar mit der CORBA ([EMMERICH 2003], S. 133-148). Sie sind, ähnlich der CORBA, für die Realisierung von funktionaler Integration geeignet.

Terminologieserver

Terminologieserver stellen Begriffsverzeichnisse zur Verfügung. Neben der einfachen Bereitstellung der Verzeichnisse bieten sie i. d. R. umfangreiche Funktionen zur Suche von Begriffen oder zur Abbildung von Begriffssystemen aufeinander an. Oft enthalten sie Verweise auf weiterführende Literatur zu den Begriffen ([INGENERF and DIEDRICH 1997]). Terminologieserver sind für die Realisierung von semantischer Integration geeignet.

HL7 Context Management Standard

Die Clinical Context Object Workgroup (CCOW), die zur HL7-Organisation gehört, hat eine Standard-Reihe für Kontextintegration veröffentlicht.

Der „Kern“-Standard „Health Level Seven Standard Context Management Specification, Technology- and Subject-Independent Component Architecture“ sieht vor, dass Anwendungssysteme, die hinsichtlich eines bestimmten Kontextes synchronisiert werden müssen, über eine zentrale Komponente, den Kontextmanager, koordiniert werden. Der Standard beschreibt die notwendigen Schnittstellen des Kontextmanagers und der beteiligten Anwendungssysteme ([CCOW 2000c]).

Speziell für den Patientenkontext und den Benutzerkontext wurden ergänzende Standards veröffentlicht, die die Spezifikationen der speziellen Kontextdaten beinhalten ([CCOW 2000a],

[CCOW 2000b]). Zur Standard-Reihe gehören außerdem Standards zur Umsetzung der Kontextsynchronisierung auf der Basis bestimmter Techniken.

Standards für Präsentationsintegration

Für die Herstellung von Präsentationsintegration wurden mehrere Standards zur Gestaltung von Benutzungsschnittstellen entworfen. Sie enthalten Vorschriften und Richtlinien für die Gestaltung von Bildelementen wie Menüs, Tabellen oder Steuerelemente für Dialogfenster.

Beispiele sind Common User Application (CUA) von IBM ([BERRY and REEVES 1992]), Motif ([X/OPEN 1995]), die Apple Human Interface Guidelines ([APPLE COMPUTER INC. 1987]) und die Grundsätze ergonomischer Dialoggestaltung für Bildschirmarbeitsplätze (Standard DIN 66234, Teil 8; vgl. auch Standard ISO 9241-10) des Deutschen Institutes für Normung (DIN) ([DIN 1988]).

4.4 Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen

Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen enthalten, wie die im vorhergehenden Abschnitt vorgestellten technischen Rahmenwerke, Empfehlungen oder Anleitungen für ein systematisches Management von Informationssystemen. Sie schließen unternehmensorientierte Managementmethoden und -werkzeuge, z. B. für die Geschäftsprozessmodellierung, in die Architekturbetrachtung ein. Manche von ihnen sind aus technischen Rahmenwerken für Informationssystemarchitekturen entstanden und deshalb durch diese geprägt (z. B. TOGAF, siehe Abschnitte 4.2.5 und 4.4.4).

Auch unternehmensbezogene Rahmenwerke stellen Vorgehens-Referenzmodelle zur Verfügung oder geben durch Unterscheidung verschiedener Sichten Unterstützung bei der Entwicklung oder Weiterentwicklung von Informationssystemarchitekturen (vgl. Abschnitt 4.2).

Spezielle Standards für Integrationstechniken und unternehmensbezogene Rahmenwerke

Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen bilden i. d. R. nicht die Grundlage für die Entwicklung spezieller Standards für bestimmte Integrationstechniken. Sie wurden entwickelt, um die Durchführung architekturbezogener Projekte unter Berücksichtigung der Ziele, der Geschäftsprozesse und weiterer Spezifika des betrachteten Unternehmens zu unterstützen. In den folgenden Abschnitten sind daher, im Gegensatz zu den vorhergehenden, keine Unterabschnitte zu speziellen Standards für Integrationstechniken enthalten.

Eine Ausnahme ist der in Abschnitt 4.4.3 beschriebene Ansatz *Enterprise Application Integration (EAI)*. Geleitet durch die EAI-Bestrebungen wurde die Entwicklung verschiedener Integrationstechniken vorangetrieben und die Anwendung existierender Techniken gefördert. Der Abschnitt zu EAI enthält also kurze Unterabschnitte zur EAI-Technologie und zu Architektur-Referenzmodellen für EAI.

Integrationsanforderungen und unternehmensbezogene Rahmenwerke

Mit der genannten Unabhängigkeit unternehmensbezogener Rahmenwerke von speziellen Integrationstechniken ist auch eine Unabhängigkeit von einzelnen Integrationsanforderungen oder

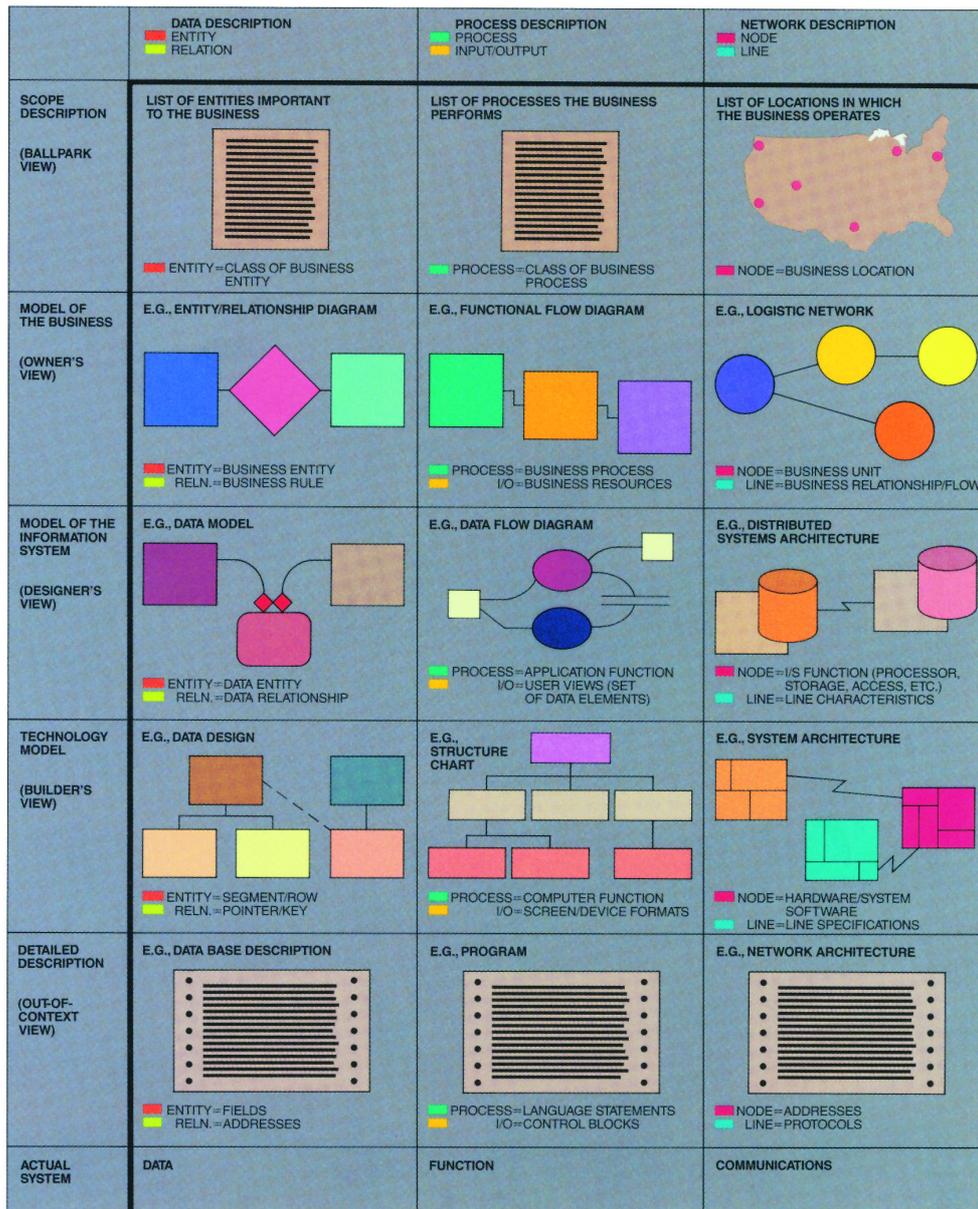


Abbildung 4.11: Das Zachman-Rahmenwerk für Informationssystemarchitektur (Quelle: [ZACHMAN 1999])

Anforderungskategorien verbunden. Alle Projekte, die die Integration und damit auch die Architektur betreffen, sollten durch unternehmensbezogene Rahmenwerke unterstützt werden können. In den folgenden Abschnitten sind daher, im Gegensatz zu den vorhergehenden, keine Unterabschnitte zu Integrationsanforderungen enthalten.

4.4.1 Das Zachman-Rahmenwerk

Das Framework for Information Systems Architecture von J. A. Zachman (Zachman-Rahmenwerk) ist eines der ersten unternehmensbezogenen Rahmenwerke für Informationssystemarchi-

tekturen. Es ist wesentliche Grundlage für die Entwicklung einer Reihe von US-amerikanischen Rahmenwerken (Abbildung 4.8).

Im Gegensatz zum TOGAF (siehe Abschnitte 4.2.5 und 4.4.4) steht im Zachman-Rahmenwerk nicht ein Vorgehens-Referenzmodell, sondern die Unterscheidung verschiedener Sichtweisen auf Informationssysteme bzw. ihre Architektur im Mittelpunkt. Im Zachman-Rahmenwerk werden dazu fünf Sichtweisen unterschieden (übersetzt aus [ZACHMAN 1987]):

- die Anwendungsbereichsbeschreibung (Rahmensichtweise⁸),
- das Geschäftsmodell / Unternehmensmodell (Sichtweise des Eigentümers),
- das Informationssystemmodell (Konstrukteurssichtweise),
- das technische Modell (Sichtweise des Erbauers) und
- die detaillierte (technische) Beschreibung (kontextlose Sichtweise).

Diese Sichtweisen werden jeweils in

- eine Datenbeschreibung,
- eine Prozessbeschreibung und
- eine Netzwerkbeschreibung

aufgeteilt (Abbildung 4.11). Damit ergibt sich eine Matrix, deren Zellen für unterschiedliche Architekturbetrachtungen stehen.

Jeder dieser Architekturbetrachtungen kann ein Vorrat an Methoden und Werkzeugen, insbesondere für die Modellierung, zugeordnet werden. Beispielsweise sind für die Prozessbeschreibung innerhalb des Unternehmensmodells Methoden und Werkzeuge der Geschäftsprozessmodellierung geeignet. Für die Datenbeschreibung innerhalb des technischen Modells können z. B. Datenbankschemabeschreibungen verwendet werden. Diese Zuordnung wird im Zachman-Rahmenwerk jedoch nicht ausdrücklich vorgenommen.

4.4.2 Die Architektur integrierter Informationssysteme

Die Architektur integrierter Informationssysteme (ARIS) ist der vermutlich bekannteste deutsche Ansatz zu Informationssystemarchitekturen ([SCHEER 1998b], [SCHEER 1998a]). Sie ist in der universitären Forschung entstanden und hat durch erfolgreichen Transfer in die Wirtschaft große Verbreitung gefunden. Ihre Anwendung in mehreren großen Unternehmen hat die Verbreitung gefördert.

Im ARIS-Ansatz werden fünf Sichten⁹ auf Informationssysteme unterschieden (Abbildung 4.12):

- die Organisationssicht,
- die Steuerungssicht,
- die Datensicht,
- die Funktionssicht und
- die Leistungssicht.

Jeder der Sichten werden drei Ebenen zugeordnet:

⁸ In [ZACHMAN 1987] wird die Bezeichnung *view* verwendet. Hier wird davon ausgegangen, dass damit der Begriff *Sichtweise*, wie er bisher in diesem Kapitel verwendet wurde, gemeint ist.

⁹ Die in der ARIS verwendete Bezeichnung *Sicht* entspricht dem Begriff *Sichtweise*, wie er bisher in diesem Kapitel verwendet wurde.

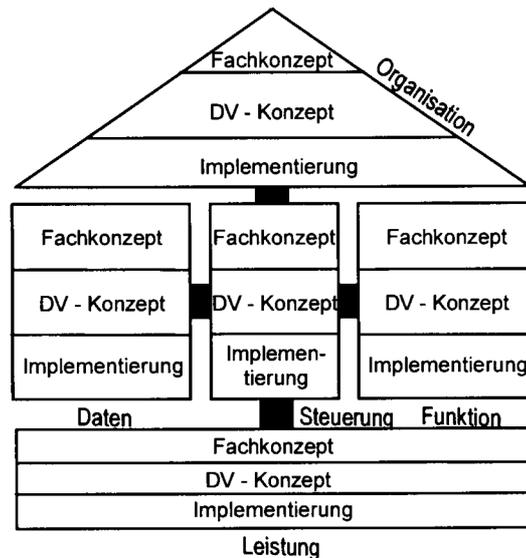


Abbildung 4.12: Das ARIS-Haus (Quelle: [SCHEER 1998b], S. 46, Abbildung 20)

- das Fachkonzept,
- das DV-Konzept und
- die Implementierung.

Jedem Sicht-Ebene-Paar können Methoden und Werkzeuge für seine Bearbeitung zugeordnet werden. [SCHEER 1998b] und [SCHEER 1998a] enthalten entsprechende Vorgaben und Empfehlungen (siehe Kasten „Beispiele für Methoden zur Datensicht und zur Steuerungssicht der ARIS“).

Die in den Fachkonzepten und DV-Konzepten zulässigen Begriffe und deren Beziehungen

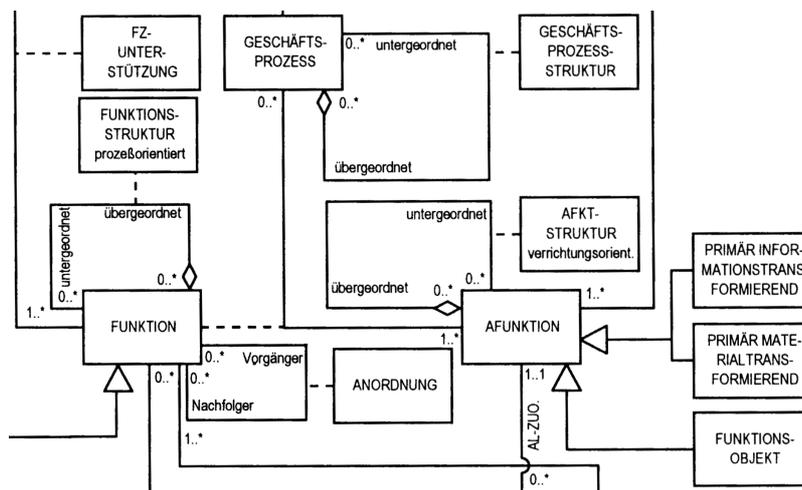


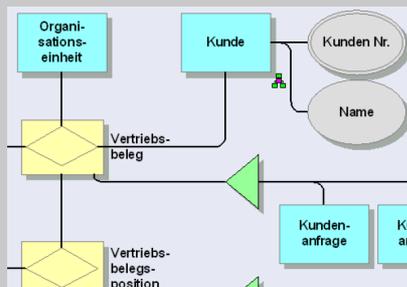
Abbildung 4.13: Metamodell der Fachkonzeptebene der Funktionssicht (Ausschnitt, Quelle: [SCHEER 1998a], S. 38, Abbildung 33)

Kasten 4.5: Beispiele für Methoden zur Datensicht und zur Steuerungssicht der ARIS

Hinweis: Die Beispiele können die Komplexität des ARIS-Ansatzes nicht vollständig wiedergeben.

Datensicht

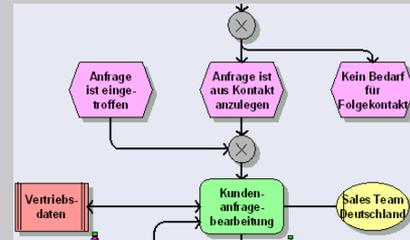
Bei der Erstellung des Fachkonzeptes werden Informationsmodelle (auch: semantische Datenmodelle) erstellt, z. B. auf der Basis des Entity-Relationship-Modells:



(Quelle: [IDS SCHEER AG 2004])

Steuerungssicht

Bei der Erstellung des Fachkonzeptes werden u. a. ereignisgesteuerte Prozessketten (EPK) verwendet:



(Quelle: [IDS SCHEER AG 2004])

Weiterhin werden u. a.

- Funktions-Organisationszuordnungsdiagramme,
- Klassendiagramme, basierend auf der objektorientierten Analyse (OOA), und
- Anwendungsfalldiagramme (Use Case Diagrams)

verwendet. Die Verwendung der verschiedenen Methoden für die Steuerungssicht ergibt sich aus den verschiedenen Verknüpfungen zwischen den Sichten.

Bei der Erstellung des DV-Konzeptes werden implementierungsabhängige logische Datenmodelle, z. B. relationale Datenmodelle, verwendet.

Kunde (Kunden Nr., Name, Adresse, ...)
Organisationseinheit (OE Nr., Name, ...)

Dabei ist, sofern zutreffend, auch die Einhaltung von Normalformen zu beachten.

Die logischen Datenmodelle werden in die Datenbeschreibungssprache (Data Description Language (DDL), auch: Datendefinitionssprache (Data Definition Language)) der verwendeten Datenbanksysteme überführt, z. B. in einen SQL-Dialekt.

```
CREATE TABLE kunden (
  kundennr DECIMAL(10) PRIMARY KEY,
  ...
);
```

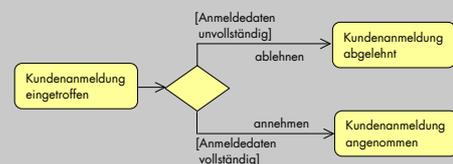
Zum DV-Konzept der Datensicht gehört auch das Definieren von Datenbanktriggern, die die Datenintegrität sichern.

Bei der Implementierung des DV-Konzeptes werden interne Datenbankschemata angelegt, bei denen, im Unterschied zu den Datenbankschemata des DV-Konzeptes, u. a. Optimierungen für schnelleren Zugriff vorgenommen werden. Dabei wird die zum Datenbanksystem gehörende Datenspeicherbeschreibungssprache (Data Storage Description Language DSDL) verwendet.

(vgl. [SCHEER 1998a])

Im DV-Konzept werden die im Fachkonzept angegebenen Steuerungsvorgaben implementierungsabhängig definiert. Dazu gehören u. a.

- Verfeinerungen des objektorientierten Entwurfs des Fachkonzeptes und Ergänzung um Zustands-, Sequenz- und Aktivitätendiagramme,



- Zuordnungen von Datenbankschemata aus der Datensicht zu Modulen der Funktionssicht und
- Definitionen von Datenbanktransaktionen und Datenbanktriggern.

Bei der Implementierung wird das DV-Konzept mit Hilfe von Programmiersprachen und den von den Datenbankmanagementsystemen bereitgestellten Funktionen in ausführbare Anwendungssysteme überführt.

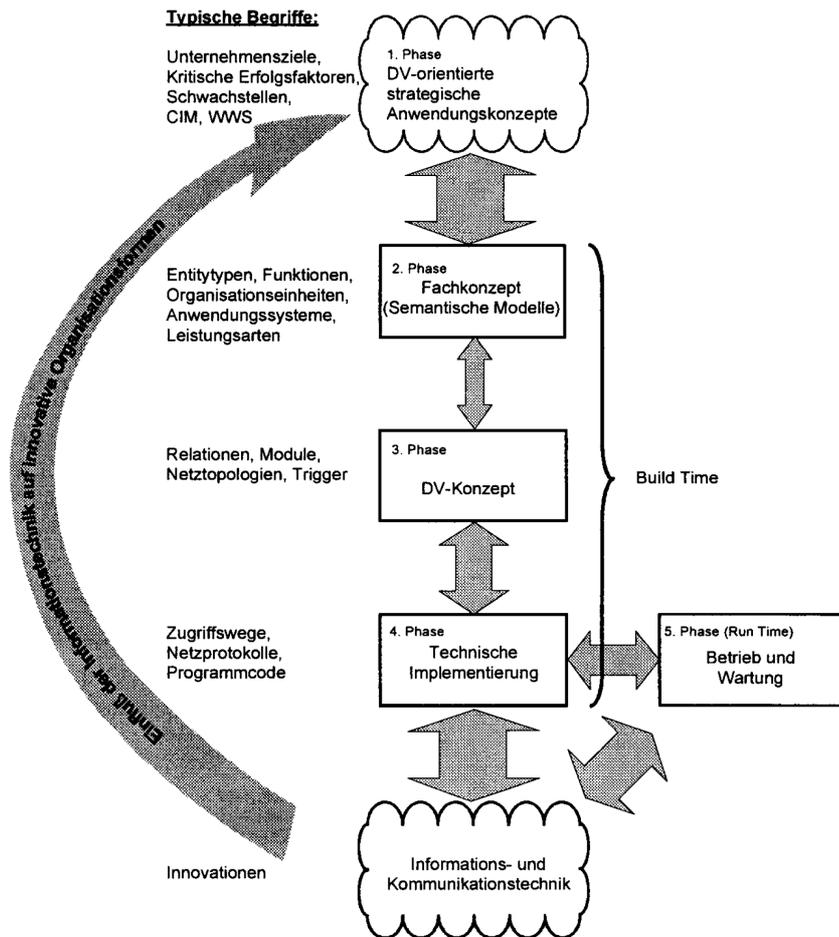


Abbildung 4.14: Das ARIS-Phasenmodell (Quelle: [SCHEER 1998b], S. 39, Abbildung 16)

werden durch miteinander verknüpfte Metamodelle vorgeschrieben. Dadurch werden die Arbeiten zum Architekturentwurf, die zu den einzelnen Sicht-Ebene-Paaren gehören, aufeinander abgestimmt. Die Metamodelle wurden als Klassendiagramme mit der Unified Modeling Language (UML) erstellt. Das durch Verknüpfung der einzelnen Metamodelle entstehende ARIS-Metamodell wird auch ARIS-Informationsmodell genannt (Abbildung 4.13).

Die Sichten und Ebenen des ARIS-Ansatzes sind eingebettet in einen Geschäftsprozessrahmen. Die Optimierung von Geschäftsprozessen ist danach Hauptziel jeder Entwicklung bzw. Weiterentwicklung von Informationssystemen (vgl. Abschnitt 5.2). Prozessmodelle sind wichtiger Ausgangspunkt für den Architekturentwurf. Gleichzeitig beeinflusst jede Änderung der Architektur die Geschäftsprozesse. Die Zusammenhänge zwischen

- Prozessgestaltung,
- Prozessplanung und -steuerung,
- Workflowsteuerung und
- Anwendungssystemen

sind in einem Phasenmodell festgehalten (Abbildung 4.14).

4.4.3 Enterprise Application Integration

Unter dem Titel *Enterprise Application Integration (EAI)* wurde in den letzten Jahren ein umfassender Integrationsansatz propagiert und diskutiert ([HASSELBRING 2000], [LINTHICUM 2000b]). Es existiert kein offizieller Standard oder allgemein anerkanntes Rahmenwerk, das EAI und die zugehörigen Methoden definiert und gegenüber anderen Architektur- oder Integrationsansätzen abgrenzt.

Ausgangspunkt der EAI-Bestrebungen ist die in vielen Unternehmen vorhandene Vielfalt an Werkzeugen zur Informationsverarbeitung: Unterschiedliche Organisationseinheiten eines Unternehmens, möglicherweise auch einzelne Mitarbeiter innerhalb einer Organisationseinheit, nutzen unterschiedliche Werkzeuge bei der Erfüllung unterschiedlicher oder eventuell sogar derselben Aufgaben. Dieser Sachverhalt wird hier nach [HASSELBRING 2000] als vertikale Fragmentierung eines Informationssystems bezeichnet.

Die Gewährleistung einer angemessenen Zusammenarbeit der Mitarbeiter bzw. der Einrichtungen, die für das Erreichen der Unternehmensziele notwendig ist, stellt oft eine große Herausforderung für das Unternehmen dar. Im Zentrum der EAI-Bestrebungen steht die Integration vorhandener Anwendungssysteme, die auf einen bestimmten Anwendungsbereich zugeschnitten sind, jedoch nicht oder kaum auf die Integration mit anderen Anwendungssystemen vorbereitet sind. Diese Anwendungssysteme werden auch als Altsysteme (legacy systems) bezeichnet [HASSELBRING 2000].

EAI-Ebenen

In der EAI-Literatur spiegelt sich u. a. die Idee der Überwindung vertikaler Fragmentierung durch horizontale Integration auf verschiedenen Ebenen wider. In [HASSELBRING 2000] wird die Unterscheidung von verschiedenen Integrationsebenen zunächst zur Abgrenzung von EAI gegenüber anderen Integrationstypen genutzt (Abbildung 4.15). In der EAI-Literatur wird dieser Ansatz durch Unterscheidung verschiedener EAI-Typen bzw. -Ebenen weitergeführt bzw. ergänzt.

Unterscheidungen mehrerer Ebenen, auf denen Integration hergestellt werden kann, können als Rahmenwerke betrachtet werden, die Orientierung in Integrationsprojekten geben. Abbildung 4.16a zeigt die Einordnung von EAI als eine von drei Integrationsebenen, die Abbildungen 4.16b und c zeigen die in [LINTHICUM 2000b] und [LONGO 2001] beschriebenen

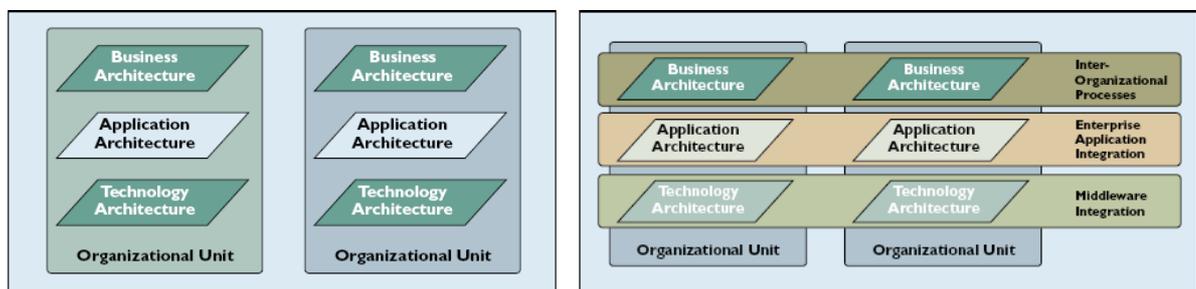


Abbildung 4.15: Vertikale Fragmentierung von Informationssystemen und horizontale Integration (Quelle: [HASSELBRING 2000], S. 34-35, Figure 1 und Figure 2)

4.4 Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen

a)	b)	c)
Inter-Organizational Processes	User Interface Level EAI	Business-to-Customer Integration
Enterprise Application Integration	Method Level EAI	Business-to-Business Integration
Middleware Integration	Application Interface Level EAI	Application-to-Application Integration
	Data Level EAI	Component-Based Applications Development

Abbildung 4.16: Unterscheidung von Integrationsebenen in [HASSELBRING 2000] (a) sowie von EAI-Ebenen in [LINTHICUM 2000b] (b) und in [LONGO 2001] (c)

EAI-Ebenen¹⁰. In [LINTHICUM 2000b] werden die EAI-Ebenen folgendermaßen beschrieben (auszugsweise übersetzt aus [LINTHICUM 2000b], S. 18-20¹¹):

Daten-EAI ist der Prozess, inkl. der Techniken und Technologie, des Bewegens von Daten zwischen Datenspeichern. Das kann als das Extrahieren von Informationen aus einer Datenbank, eventuell das Verarbeiten der Informationen soweit erforderlich und das Aktualisieren der Informationen in einer anderen Datenbank beschrieben werden.

Anwendungssystemschnittstellen-EAI bezieht sich auf die Nutzung von Schnittstellen, die von kundenspezifischen oder handelsüblichen Anwendungssystemen bereitgestellt werden. Entwickler nutzen diese Schnittstellen, um sowohl auf Geschäftsprozesse als auch auf einfache Informationen zuzugreifen.

Methoden-EAI ist die gemeinsame Nutzung der Geschäftslogik, die innerhalb eines Unternehmens vorhanden ist. Beispielsweise könnte die Methode¹² zur Aktualisierung einer Kundenakte von mehreren Anwendungssystemen aufgerufen werden, und Anwendungssysteme könnten ihre Methoden gegenseitig aufrufen, ohne dass jede Methode in jedem einzelnen Anwendungssystem neu geschrieben werden muss.

Benutzungsschnittstellen-EAI ist ein eher primitiver, aber trotzdem wichtiger Ansatz. Bei Anwendung dieses Szenarios können Architekten und Entwickler Anwendungssysteme durch Nutzung ihrer Benutzungsschnittstellen als gemeinsamen Integrationsbezugspunkt bündeln.

EAI-Technologie

Integrationstechnologien und zugehörige Integrationstechniken füllen einen großen Anteil der EAI-Literatur aus. Sehr oft beschriebene Technologien sind

- die Technologie der nachrichtenorientierten Middleware (message-oriented middleware, MOM), wobei insbesondere Kommunikationsserver (message broker) eine bedeutende Rolle einnehmen, ([LINTHICUM 2000b], S. 291-317, [LUBLINSKY 2002], [BLOCH and BRUCE 2004], [NIEMANN et al. 2002]),
- die Technologie des Internets mit ihren vielen unterschiedlichen, sich oft ergänzenden Techniken, z. B. dem Hypertext Transfer Protocol (HTTP) oder den Java Server Pages (JSP), ([JURIC 2002], [LUBLINSKY and FARREL JR. 2001], [BLOCH and BRUCE 2004]) oder, eng damit verknüpft,
- die Technologie der verteilten Bereitstellung von Anwendungsfunktionalität über Komponenten auf Anwendungsservern (application server). Bekannte Industriestandards für

¹⁰ Ergänzend siehe z. B. auch die Unterscheidung von B2B-Integrationstypen in [LUBLINSKY 2002].

¹¹ Die Abkürzung *EAI* wird hier nicht übersetzt.

diese Komponenten sind Enterprise Java Beans (EJB) oder das Component Object Model (COM/COM+). Für die Kommunikation mit bzw. zwischen verteilten Objekten sind Standards wie die CORBA (vgl. Abschnitt 4.2.3) oder das Distributed Component Object Model (DCOM) (vgl. Abschnitt 4.3) von Bedeutung ([LINTHICUM 2000a]).

Im Zusammenhang mit den verschiedenen EAI-Ebenen aus [LINTHICUM 2000b] werden im selben Werk auch dafür geeignete Integrationstechniken genannt. Zur Realisierung von Daten-EAI sind beispielsweise Techniken zum Datenbankzugriff geeignet. Dabei müssen u. a.

- die Funktionsweise der von Datenbanksystemen bereitgestellten Schnittstellen, z. B. Open DataBase Connectivity (ODBC) oder Java Database Connectivity (JDBC),
- die den Datenbanken zugrunde liegenden Datenmodellierungsansätze, z. B. relationaler oder objektorientierter Ansatz, und
- die auf den Datenmodellierungsansätzen basierenden Datenbankschemata

berücksichtigt werden ([LINTHICUM 2000b], S. 23-36, siehe auch [LUBLINSKY 2002]). Zur Realisierung von Methoden-EAI sind beispielsweise Techniken zum verteilten Aufruf von Methoden geeignet. Dabei müssen u. a.

- die in den beteiligten Anwendungssystemen bereitgestellte Funktionalität und
- die zur Verfügung stehenden Vermittlungstechniken, z. B. Object Request Broker (ORB) oder Transaction Processing Monitor (TP monitor),

berücksichtigt werden ([LINTHICUM 2000b], S. 61-77).

EAI-Architektur-Referenzmodelle

In [SCHMIDT 2002] wird ein einfaches Architektur-Referenzmodell mit fünf Ebenen für die Unterstützung von EAI beschrieben. Es ordnet jeder der Architekturebenen einige zur betreffenden Ebene gehörende Modelle und Dienste zu (Abbildung 4.17).

Im Beitrag [LUTZ 2000] werden fünf Architekturmuster für EAI vorgestellt, die zum Verständnis der vielen unterschiedlichen EAI-Ansätze beitragen können (Abbildung 4.18). Die ersten vier Muster haben eine zentrale Komponente, die auch den Namen des Musters bestimmt (übersetzt aus [LUTZ 2000], S. 66):

1. Der **Integrationsadapter (integration adapter)** wandelt eine vorhandene Anwendungsschnittstelle in eine benötigte Schnittstelle um.
2. Der **Integrationsbote (integration messenger)** minimiert die Kommunikationsabhängigkeiten zwischen Anwendungssystemen hinsichtlich.
3. Die **Integrationsfassade (integration façade)**¹³ stellt eine vereinfachte Schnittstelle zu Server-Komponenten (back-end applications) zur Verfügung, um die Abhängigkeiten zwischen Client- und Server-Komponenten zu minimieren.
4. Der **Integrationsmediator (integration mediator)** kapselt die Interaktionslogik von Anwendungssystemen, um die Abhängigkeiten zwischen verschiedenen Anwendungssystemen zu minimieren.
5. Das **Prozessautomatisierungsmuster (process automator pattern)** beschreibt einen Architekturansatz für die Minimierung der Abhängigkeiten zwischen Prozessautomatisierungslogik und Anwendungssystemen.

¹³ In [LUTZ 2000] wird der Buchstabe ç im Text, jedoch nicht in den Abbildungen verwendet.

4.4 Unternehmensbezogene Rahmenwerke für Informationssystemarchitekturen

Architectural Layer	Services or Functions Performed	Organizational Responsibility
Business Process	<ul style="list-style-type: none"> • Business Process Modeling • Process Registry • Enterprise Workflow Mgmt. • Process Control and Reporting 	Business Process Owners
Process Mediation	<ul style="list-style-type: none"> • Business Event Model • Event Registry • Message Delivery • Message Management 	EAI Management Group
Information Services	<ul style="list-style-type: none"> • Shared Data Model • Domain Object Models • Object Repository • Inference Engines • Common Information Services 	EAI Management Group
Application Mediation	<ul style="list-style-type: none"> • Business Event Model • Event Registry • Message Delivery • Message Management 	Application Owners
Application Services	<ul style="list-style-type: none"> • User Interface • Application Workflow • Data Steward and Repository 	Application Owners

Abbildung 4.17: Einfaches Architektur-Referenzmodell für EAI (Quelle: [SCHMIDT 2002])

Beispiel für Integrationsadapter sind die sogenannten Hüllen (Wrapper), die spezielle Anwendungsschnittstellen auf von anderen Komponenten besser nutzbare Schnittstellen abbilden. Sie können wichtiges Hilfsmittel sein, um die anderen EAI-Architekturmuster umzusetzen.

Beispiele für Integrationsboten sind Nachrichtenpuffersysteme oder auch Kommunikationsserver, mit i. d. R. mächtiger Funktionalität für die Filterung, die Übersetzung und die Pufferung von Nachrichten.

Integrationsfassaden kommen u. a. dann zum Einsatz, wenn die von mehreren Servern bereitgestellten Schnittstellen für den Zugriff durch Clients vereinheitlicht und zusammengefasst werden sollen. Beispiel ist eine Komponente, über die die Verfügbarkeit von unterschiedlichen Server-Komponenten abgefragt werden kann.

Integrationsmediatoren kommen u. a. dann zum Einsatz, wenn Informationen oder Funktionalitäten, die wichtig für die gemeinsame Nutzung von Anwendungssystemen sind, aus den speziellen Anwendungssystemen herausgelöst werden. Beispiel ist eine Komponente, die die Abbildung von Identifikationsnummern aus dem Nummernraum eines Anwendungssystems in den Nummernraum anderer Anwendungssysteme vornimmt. Aus dem Gesundheitswesen kann in diesem Zusammenhang das Konzept des Master Patient Index genannt werden.

Das Prozessautomatisierungsmuster nutzt eine spezielle Form der Integrationsfassaden, die das Initiieren von Aktivitäten aus einem Prozessablauf in Aufrufe der von speziellen Anwendungssystemen bereitgestellten Dienste umsetzt. Damit wird die Komponente zur Prozesssteuerung unabhängiger von anwendungssystemspezifischen Details.

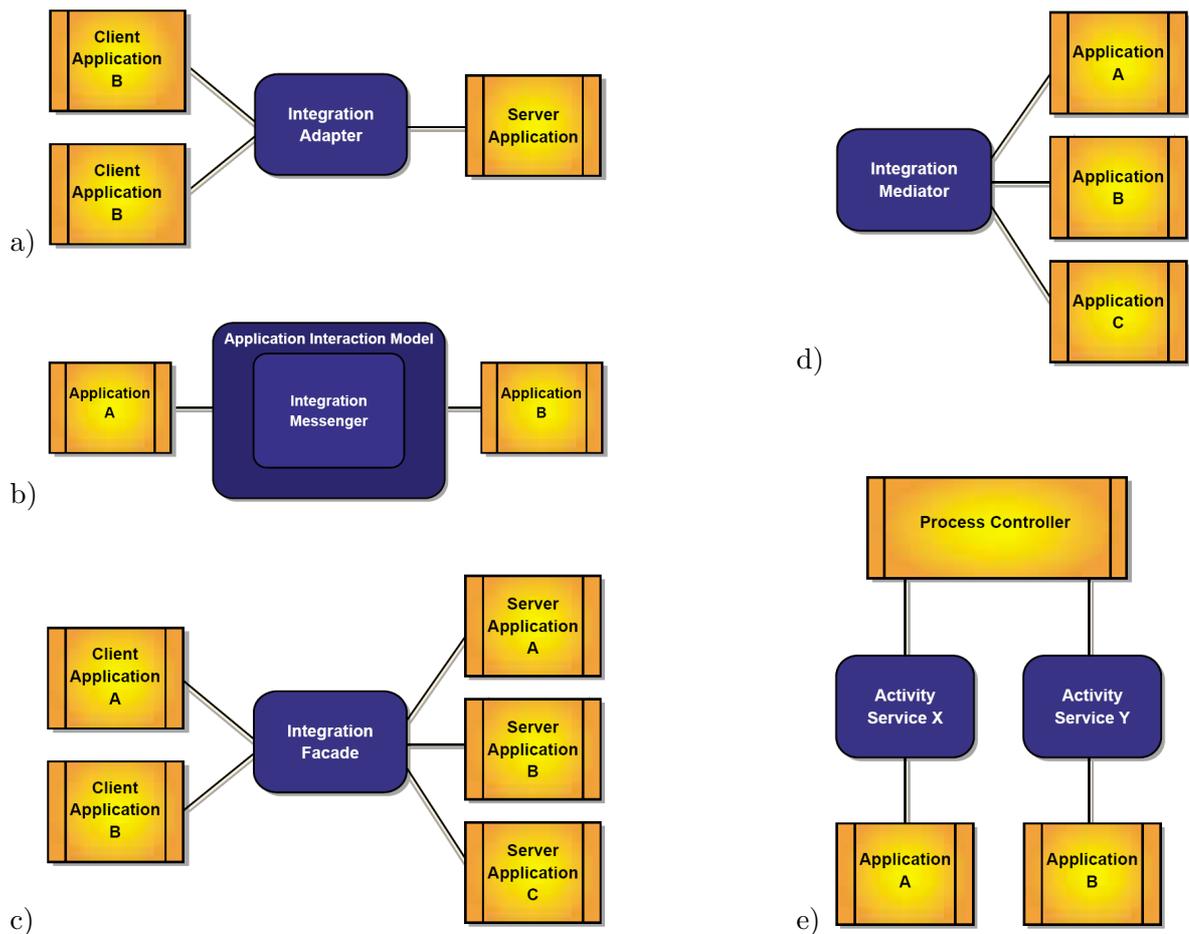


Abbildung 4.18: EAI-Architekturmuster (Quelle: [LUTZ 2000])

Ein Vorgehens-Referenzmodell für EAI

Im bereits mehrfach referenzierten [LINTHICUM 2000b] wird auch ein Vorgehens-Referenzmodell zur Herstellung von EAI beschrieben. Es besteht aus zwölf Schritten (übersetzt aus [LINTHICUM 2000b], S. 92, ergänzend siehe auch [ALLEN 2001]):

1. Verstehen des Unternehmens und des Problembereiches
2. Zusammenhänge der Daten verstehen
3. Zusammenhänge der Prozesse verstehen
4. Ermitteln aller Anwendungsschnittstellen
5. Ermitteln der Geschäftsereignisse
6. Ermitteln der Datentransformationsszenarios
7. Zuordnung der Informationsbewegungen
8. Anwenden der Technologie
9. Testen, Testen, Testen
10. Berücksichtigen der Leistungsfähigkeit
11. Definieren des Nutzens
12. Festlegung von Wartungsmaßnahmen

4.4.4 The Open Group Architectural Framework, Version 8

The Open Group Architectural Framework, Version 8, ist eine im Vergleich zur Version 7 (vgl. Abschnitt 4.2.5) wesentlich erweiterte und überarbeitete TOGAF-Version (im Folgenden abgekürzt mit TOGAF8). Darin werden u. a. die Bezüge der Informationssystemarchitektur zur Unternehmensarchitektur berücksichtigt.

Die zentralen Inhalte von TOGAF8 sind

- die TOGAF Architecture Development Method (TOGAF ADM),
- das TOGAF Enterprise Continuum (TOGAF EC) und
- die TOGAF Resource Base (TOGAF RB).

Die TOGAF Architecture Development Method

Die TOGAF ADM in der Version 8 ist ein Vorgehens-Referenzmodell zur Entwicklung und Weiterentwicklung von Unternehmens- und Informationssystemarchitekturen. Sie enthält mehrere Phasen, von denen, im Unterschied zur Version 7, nicht nur eine, sondern mehrere Phasen der Architekturentwicklung gewidmet sind. Die Phasen sind (übersetzt aus [THE OPEN GROUP 2003], Part II):

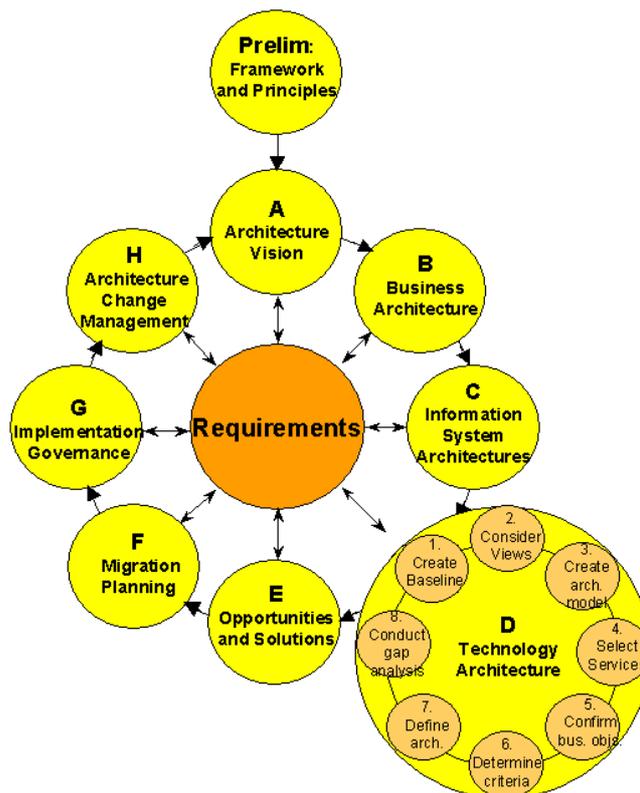


Abbildung 4.19: Die Phasen der TOGAF Architecture Development Method und die Schritte der Phase D (Quelle: [THE OPEN GROUP 2003], Part II, Introduction, Figure 2)

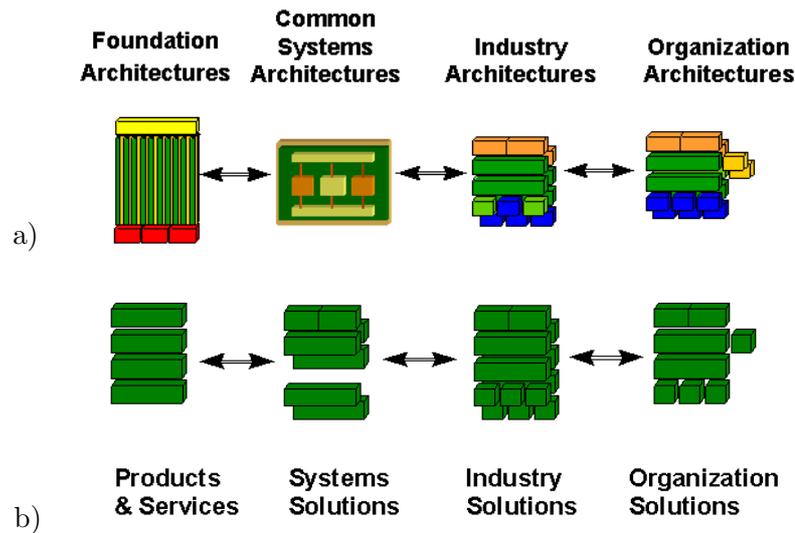


Abbildung 4.20: Architekturkontinuum (a) und Lösungskontinuum (b) in TOGAF, Version 8 (Quelle: [THE OPEN GROUP 2003], Part III, Enterprise Continuum in Detail, Architecture Continuum, Figure 1 und Solutions Continuum, Figure 3)

- Vorbereitende Phase: Rahmenbedingungen und Grundsätze
- Phase A: Architekturvision,
- Phase B: Unternehmensarchitektur,
- Phase C: Informationssystemarchitekturen,
- Phase D: Technologiearchitektur,
- Phase E: Möglichkeiten und Lösungen,
- Phase F: Migrationsplanung,
- Phase G: Implementierungsüberwachung und
- Phase H: Architekturveränderungsmanagement.

Jede Phase der TOGAF ADM ist unterteilt in Arbeitsschritte (Abbildung 4.19). Phase D besteht beispielsweise aus acht Schritten (übersetzt aus [THE OPEN GROUP 2003], Part II, Phase D):

1. Erstellung einer Basisbeschreibung im TOGAF-Format,
2. Berücksichtigung verschiedener Architektur-Referenzmodelle, Sichten und Werkzeuge,
3. Erstellung eines Architekturmodells aus Bausteinen,
4. Auswahl der von den Bausteinen benötigten Dienste,
5. Überprüfung der Einhaltung von Unternehmenszielen,
6. Bestimmung der Kriterien für die Spezifikationsauswahl,
7. Vervollständigung der Architekturdefinition und
8. Durchführung einer Diskrepanzanalyse.

Das TOGAF Enterprise Continuum

Das TOGAF Enterprise Continuum (TOGAF EC) ergänzt die TOGAF ADM um Informationen bzw. Anleitungen zur Einordnung und Wiederverwendung von Architektur-Referenzmodellen und anderen Materialien zur Architekturentwicklung ([THE OPEN GROUP 2003], Part III). Diese Anleitungen sind aufgeteilt auf

- ein Architekturkontinuum (Architecture Continuum) und
- ein Lösungskontinuum (Solutions Continuum).

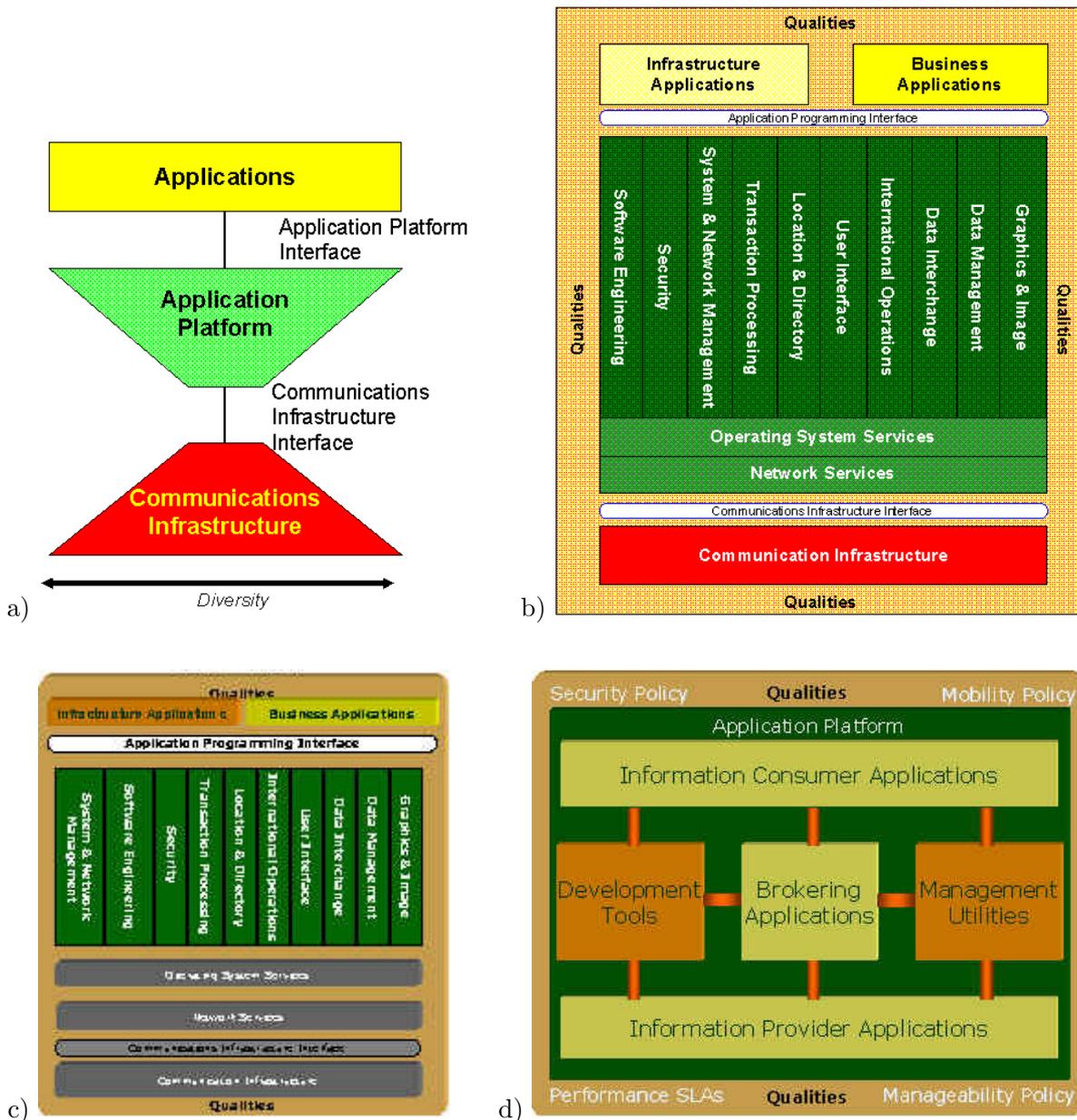


Abbildung 4.21: TOGAF TRM in TOGAF, Version 8, (a und b) und Ableitung der Elemente des TOGAF III-RM aus dem TOGAF TRM (c und d) (Quellen: [THE OPEN GROUP 2003], Part III, Foundation Architecture: Technical Reference Model, High-level Breakdown, Figure 1, und TRM in Detail, Figure 1, [THE OPEN GROUP 2003], Part III, Integrated Information Infrastructure Reference Model, High-Level View, Figure 2b und Figure 3)

Im Architekturkontinuum werden vier Architekturtypen in einer Folge zueinander in Beziehung gesetzt. Die Folge verdeutlicht ein Entwicklungsprinzip, nach dem unternehmensspezifische Architekturen (enterprise architectures, organization architectures) aus anwendungsbereichsspezifischen Architekturen (industry architectures), diese aus anwendungsbereichsübergreifenden Architekturen mit Bezug zu bestimmten Diensten (common system architectures) und diese aus grundlegenden Architekturen (foundation architectures) entwickelt werden (Ab-

bildung 4.20a). Im TOGAF selbst werden auch eine grundlegende Architektur, die TOGAF Foundation Architecture, bestehend aus

- dem TOGAF Technical Reference Model (TOGAF TRM) und
- der TOGAF Standards Information Base (TOGAF SIB),

sowie eine anwendungsbereichsübergreifende Architektur mit Bezug zum Konzept des „grenzenlosen Informationsflusses“ („boundaryless information flow“),

- das TOGAF Integrated Information Infrastructure Reference Model (TOGAF III-RM),

beschrieben.

Im Lösungskontinuum werden die zu den Architekturtypen des Architekturkontinuums zugehörigen Lösungstypen zueinander in Beziehung gesetzt: unternehmensspezifische Lösungen (organization solutions) werden aus anwendungsbereichsspezifischen Lösungen (industry solutions), diese aus anwendungsbereichsübergreifenden Systemlösungen (systems solutions), diese aus Produkten und Diensten (products and services) entwickelt (Abbildung 4.20b).

Das TOGAF Technical Reference Model und die TOGAF Standards Information Base

Das bereits in Abschnitt 4.2.5 vorgestellte TOGAF TRM ist eine grundlegende Architektur zum entsprechenden Architekturtyp im Architekturkontinuum. Es ist ein Architektur-Referenzmodell, das keinen Bezug zu bestimmten Anwendungsbereichen oder bestimmten dienstbezogenen anwendungsbereichsübergreifenden Ansätzen hat (Abbildung 4.21a und b, vgl. Abschnitt 4.2.5). Die TOGAF SIB gibt Standards für die Dienstkategorien aus dem TOGAF TRM an (vgl. Abschnitt 4.2.5).

Das TOGAF Integrated Information Infrastructure Reference Model

Das TOGAF III-RM ist eine anwendungsbereichsübergreifende Architektur zum entsprechenden Architekturtypen im Architekturkontinuum. Es ist ein Architektur-Referenzmodell, das das Konzept des „grenzenlosen Informationsflusses“ (boundaryless information flow) bzw. eine zugehörige Architektur, die Integrated Information Infrastructure (III), in den Mittelpunkt stellt. Entsprechend dem Architekturkontinuum enthält und erweitert das TOGAF III-RM Elemente des TOGAF TRM (Abbildung 4.21). Für eine ausführlichere Vorstellung des TOGAF III-RM sei hier auf [THE OPEN GROUP 2003], Part III, Integrated Information Infrastructure Reference Model verwiesen.

4.4.5 Enterprise Application Planning

Das letzte hier vorgestellte Rahmenwerk ist der Ansatz Enterprise Application Planning (EAP) von S. H. Spewak ([SPEWAK and HILL 1992]). Er beschreibt ein Vorgehens-Referenzmodell für den Entwurf und die Entwicklungsplanung von Informationssystemarchitekturen.

*„Enterprise Architecture Planning is the Process of **defining architectures** for the use of information in support of the business and the **plan** for implementing those architectures.“ ([SPEWAK and HILL 1992], S. 1)*

EAP ist ausdrücklich auf die ersten beiden Sichten im Zachman-Rahmenwerk bezogen.

Das Vorgehensmodell besteht aus sieben Phasen, aufgeteilt auf vier Ebenen (Abbildung 4.22). Die Phasen sind (übersetzt aus [SPEWAK and HILL 1992], S. 13-17):

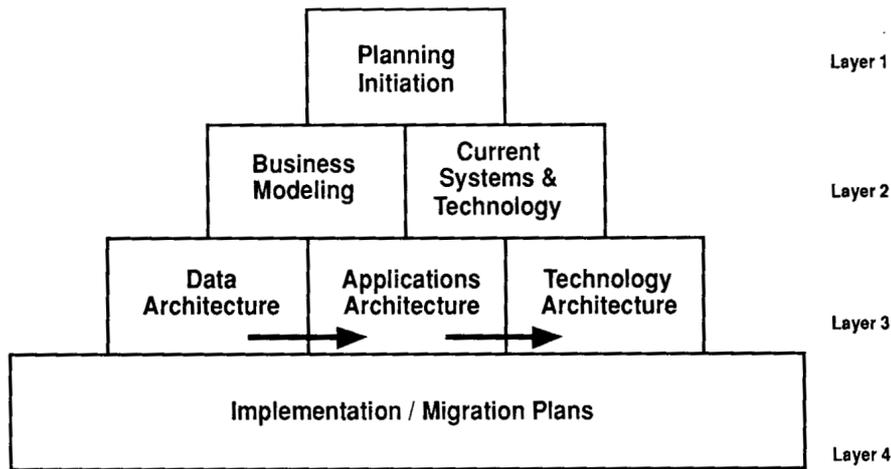


Abbildung 4.22: Phasen des Enterprise Architecture Planning (Quelle: [SPEWAK and HILL 1992], S. 13, Figure 1.7)

Ebene 1–Wo wir starten

Phase 1: Planungsbeginn. Starten des EAP auf dem richtigen Weg, einschließlich der zu nutzenden Methodologie, der beteiligten Personen und der zu nutzenden Werkzeuge. Das führt zur Erstellung eines Arbeitsplans für EAP und zur Sicherung der Zustimmung des Managements zum Durchlaufen der folgenden sechs Phasen.

Ebene 2–Wo wir sind

Phase 2: Unternehmensmodellierung. Stellt eine Wissensbasis über das Unternehmen und die im Unternehmensbetrieb verwendeten Informationen zusammen.

Phase 3: Aktuelle Systeme & Technologie. Bestimmt, welche Anwendungssysteme und unterstützenden Technologieplattformen vorhanden sind. Das ist eine zusammenfassende Bestandsaufnahme der Anwendungssysteme, Daten und Technologieplattformen, um eine Basis für langfristige Migrationspläne bereitzustellen.

Ebene 3–Wo wir zukünftig sein wollen

Phase 4: Datenarchitektur. Bestimmt die wichtigsten Arten von Daten, die im Unternehmen benötigt werden.

Phase 5: Anwendungssystemarchitektur. Bestimmt die wichtigsten Arten von Anwendungssystemen, die zur Verwaltung der Daten zur Erfüllung der Unternehmensaufgaben benötigt werden.

Phase 6: Technologiearchitektur. Bestimmt die Technologieplattformen, die benötigt werden, um eine Umgebung für die Anwendungssysteme, die die Daten verwalten und die Unternehmensaufgaben unterstützen, bereitzustellen.

Ebene 4–Wie wir dahin kommen

Phase 7: Implementierungs-/Migrationspläne. Bestimmt die Reihenfolge der Einführung der Anwendungssysteme, erstellt einen Terminplan für die Einführung, eine Kosten-Nutzen-Analyse und einen klaren Pfad für den Übergang vom aktuellen Stand zum gewünschten Stand.

Jede der sieben Phasen ist in Schritte unterteilt, zu denen jeweils Aufgaben und Richtlinien zu deren Erfüllung beschrieben werden. Einige Aufgaben sind unmittelbar mit bestimmten Methoden, z. B. der Entity-Relationship-Modellierung, verknüpft (Siehe Kästen „Schritte der EAP-Phasen ...“).

Kasten 4.6: Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien

Phase 1: Planungsbeginn

- Schritt 1: Ermittlung des Anwendungsbereiches und der Ziele für die Enterprise Application Planning
- Schritt 2: Entwicklung einer Zielvorstellung
- Schritt 3: Adaptierung einer Planungsmethodik
- Schritt 4: Bereitstellung von Rechnerressourcen für die Planung
- Schritt 5: Zusammenstellung des Planungsteams
- Schritt 6: Vorbereitung des EAP-Arbeitsplans
- Schritt 7: Einholung der Zustimmung der Unternehmensleitung

Phase 2A: Vorbereitendes Unternehmensmodell

- Schritt 1: Dokumentation der Organisationsstruktur

Gegenstand

Gegenstand dieses Schrittes ist es, die Organisationsstruktur zu dokumentieren und Personen und Standorte zu ermitteln, die Unternehmensaufgaben erfüllen. Es gibt zwei wichtige Verwendungen für diese Informationen in EAP: (1) die Ermittlung von Personen, die während der Unternehmensbefragung befragt werden müssen, und (2) die Bestimmung des Ausmaßes der gemeinsamen Nutzung von Daten und Anwendungssystemen. (...)

Erwartete Ergebnisse

1. aktuelle Organisationsdiagramme
2. Liste der Positionen und Titel inkl. der Standorte, wo sie auftreten, und die Anzahlen der Personen in den betreffenden Positionen
3. Dokumentation der Unternehmensziele und der strategischen Geschäftspläne
4. Eingabe der genannten Informationen in das Planungswerkzeug und Berichterstellung

Aufgaben und Richtlinien

- Aufgabe 1: *Sammeln Sie aktuelle Organisationsdiagramme und geben Sie die Informationen in das Planungswerkzeug ein.* Falls keine Organisationsdiagramme vorhanden oder die vorhandenen nicht aktuell sind, zeichnen Sie sie und aktualisieren Sie sie. Die Diagramme sollten so viele der folgenden Informationen wie möglich beinhalten:
 1. Abteilungen (Name und Ort)
 2. Titel/Positionen
 3. Personen (Name und Telefon)
 4. Berichtswege (direkt und indirekt)
 5. Anzahl der Personen in den Positionen oder Abteilungen.(...)
 - Aufgabe 2: *Ermitteln Sie Unternehmensstandorte und ordnen Sie sie Organisationseinheiten zu.* (...)
Die folgenden Richtlinien können verwendet werden, um zu bestimmen, ob *Unternehmensstandort* ein Attribut der Organisationseinheit oder eine eigene Struktur in der EAP-Datenbank sein sollte. *Unternehmensstandort* sollte ein Attribut sein, wenn
 - die meisten Unternehmensaufgaben an einem Standort oder in einer Stadt erfüllt werden,
 - die Organisationsstruktur an jedem Ort im Wesentlichen gleich ist oder
 - jede Region oder jeder Ort eine unabhängige Organisationsstruktur hat, bei der nur auf höchster Ebene an eine zentrale oder gemeinsame Organisationseinheit berichtet wird.*Unternehmensstandort* sollte in einer eigenen Datenstruktur gespeichert werden, wenn (...)
 - Aufgabe 3: *Dokumentieren Sie Unternehmensziele (optional).* (...)
 - Aufgabe 4: *Erstellen Sie Berichte über die Organisationseinheiten, die Berichtsstruktur, die Standorte und (optional) die Unternehmensziele.* (...)
- Schritt 2: Bestimmung der Aufgaben (...)
Schritt 3: Bekanntgabe des vorbereitenden Unternehmensmodells (...)

Phase 2B: Unternehmensbefragung

- Schritt 1: Terminplanung für Interviews
- Schritt 2: Vorbereitung der Interviews
- Schritt 3: Durchführung der Interviews
- Schritt 4: Eingabe der Daten in das Planungswerkzeug
- Schritt 5: Bekanntgabe des vollständigen Unternehmensmodells

(zusammengestellt und übersetzt aus [SPEWAK and HILL 1992])

Kasten 4.7: Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien (Fortsetzung 1)

Phase 3: Aktuelle Systeme und Technologiearchitektur

- Schritt 1: Bestimmung des Anwendungsbereiches, der Ziele und des Arbeitsplanes für einen Ressourcenkatalog
- Schritt 2: Vorbereitung der Datenerfassung
- Schritt 3: Datenerfassung
- Schritt 4: Datenregistrierung
- Schritt 5: Validierung der Ressourcenkatalogdaten und Erstellung einer vorläufigen Version des Ressourcenkataloges
- Schritt 6: Erstellung von Grafiken
- Schritt 7: Bekanntgabe des Ressourcenkataloges
- Schritt 8: Verwaltung und Wartung des Ressourcenkataloges

Phase 4: Datenarchitektur

- Schritt 1: Auflistung von möglichen Dateneinheiten für die nachfolgende Festlegung

Gegenstand

Gegenstand dieses Schrittes ist es, alle potentiellen Dateneinheiten zu ermitteln, die zur Unterstützung des Unternehmens benötigt werden. (...)

Erwartetes Ergebnis

1. Liste der Namen der möglichen Dateneinheiten. Die Liste kann vorläufige Definitionen, Synonymkennzeichnungen, Hinweise auf Aufgaben, die die Daten verwenden, und Teammitglieder, die die betreffenden Namen vorgeschlagen haben, beinhalten.

Aufgaben und Richtlinien

- Aufgabe 1: *Teilen Sie das Unternehmensmodell auf Teammitglieder auf.* Um eine Liste von Einheiten zu formulieren, sollte die folgende Dokumentation vollständig und für das EAP-Team verfügbar sein: Aufgabendefinitionen aus dem Unternehmensmodell, Formulare zu Informationsquellen, Beispiele für Informationen aus den Informationsquellen, Befragungsnotizen, (...).

Teilen Sie dieses Material, unterteilt nach Aufgabenzugehörigkeit, unter den EAP-Teammitgliedern auf; das heißt, ein Mitglied kann die Produktfertigungsaufgabe bearbeiten, ein anderes die Personalverwaltungsaufgabe usw.

- Aufgabe 2: *Jedes Teammitglied erstellt eine Liste von Einheiten (Personen, Orte, Gegenstände, Begriffe, Ereignisse) für die nachfolgende Festlegung.* (...)

- Aufgabe 3: *Fügen Sie die einzelnen Listen zusammen.* (...)

- Schritt 2: Festlegung der Dateneinheiten, Attribute und Beziehungen

- Schritt 3: Zuordnung der Dateneinheiten zu Unternehmensaufgaben

- Schritt 4: Bekanntgabe der Datenarchitektur

Phase 5: Anwendungssystemarchitektur

- Schritt 1: Auflistung möglicher Anwendungssysteme
- Schritt 2: Festlegung der Anwendungssysteme
- Schritt 3: Zuordnung der Anwendungssysteme zu Unternehmensaufgaben
- Schritt 4: Analyse der Auswirkungen auf vorhandene Anwendungssysteme
- Schritt 5: Bekanntgabe der Anwendungssystemarchitektur

Phase 6: Technologiearchitektur

- Schritt 1: Ermittlung der Technologieplattformen und -prinzipien
- Schritt 2: Festlegung der Technologieplattformen und der Verteilung von Daten und Anwendungssystemen
- Schritt 3: Zuordnung der Technologieplattformen zu Anwendungssystemen und Unternehmensaufgaben
- Schritt 4: Bekanntgabe der Technologiearchitektur

Phase 7A: Einführungsplan

- Schritt 1: Festlegung der Einführungsreihenfolge
- Schritt 2: Schätzung des Aufwandes und der benötigten Ressourcen sowie Erstellung eines Terminplanes
- Schritt 3: Schätzung von Kosten und Nutzen des Einführungsplanes
- Schritt 4: Bestimmung von Erfolgsfaktoren und Angabe von Empfehlungen

Phase 7B: Planungsabschluss

- Schritt 1: Vorbereitung des endgültigen Berichtes
- Schritt 2: Abschließende Präsentation für die Unternehmensleitung

(zusammengestellt und übersetzt aus [SPEWAK and HILL 1992])

Kasten 4.8: Schritte der EAP-Phasen und Beispiele für zugehörige Gegenstände, erwartete Ergebnisse, Aufgaben und Richtlinien (Fortsetzung 2)

Phase 7C: Überleitung zur Einführung

- Schritt 1: Planung der Überleitung
- Schritt 2: Adaptierung eines Systementwicklungsansatzes
- Schritt 3: Bereitstellung von Rechnerressourcen für die Einführung
- Schritt 4: Überarbeitung der Architekturen
- Schritt 5: Einleitung der organisatorischen Änderungen
- Schritt 6: Einstellung von Personal
- Schritt 7: Anbieten von Schulungen
- Schritt 8: Festlegung von Programmierstandards
- Schritt 9: Festlegung von Verfahrensstandards
- Schritt 10: Entwicklung eines detaillierten Terminplanes für die ersten Anwendungssysteme
- Schritt 11: Bekanntgabe der Beendigung der Überleitung

(zusammengestellt und übersetzt aus [SPEWAK and HILL 1992])

4.5 Rahmenwerke und Architekturstile

Einige der vorgestellten Rahmenwerke und der zugehörigen Standards enthalten Elemente, die für die Definition eines Architekturstiles geeignet sind (vgl. Abschnitt 3.3).

Der Standard *ISO/IEC 10746-2* definiert Begriffe für die Beschreibung offener verteilter Informationssysteme (vgl. Abschnitt 4.2.2). Das OMG Object Model und das OMG Component Model definieren die meisten der im OMA Reference Model, in der CORBA und in den ergänzenden Standards für einzelne Dienstkategorien verwendeten Begriffe (vgl. 4.2.3). Das ARIS-Metamodell ist ein sehr umfassendes Metamodell, in dem Begriffe für sehr viele Aspekte von Informationssystemen zusammengestellt sind.

Das OMA Reference Model ist ein Architektur-Referenzmodell für Architekturen mit Komponenten, die Dienste zur Verfügung stellen (vgl. Abschnitt 4.2.3). Auch die Ebenen des HIF, ergänzt um die Dienstgruppen der HISA, können als Architektur-Referenzmodell betrachtet werden (vgl. Abschnitt 4.2.4). Gleiches gilt für das TOGAF TRM (vgl. Abschnitt 4.2.5).

4.6 Vergleichbarkeit von Architekturstandards

Die Begriffsdefinitionen des RM-ODP, des OMG Object Models, des OMG Component Models und das ARIS-Metamodell haben viele Gemeinsamkeiten. Begriffe wie *Objekt*, *Schnittstelle* oder *Ereignis* werden überall nahezu gleich definiert. Viele Begriffe aus dem Standard *ISO/IEC 10746-2* für die Spezifikation von Komponenten und die Beschreibung ihrer Interaktionen sind u. a. in der von der OMG definierten UML formalisiert enthalten (vgl. Kasten „Überblick über OMG-Standards ...“ in Abschnitt 4.2.3).

Eine formale Überführung der Begriffsdefinitionen ist wegen vieler unterschiedlicher Details und unterschiedlicher Berücksichtigung verschiedener Aspekte von Informationssystemen nicht oder nur in sehr vereinfachter Weise möglich. Hier wird keine entsprechende Überführungsvorschrift angegeben.

Für die genannten Architektur-Referenzmodelle gilt gleiches wie für die Begriffsdefinitionen/Metamodelle: Ähnliche Unterscheidungen von Ebenen und Dienstkategorien sowie ähnliche Spezifikationen von Diensten weisen auf grundsätzliche Austauschbarkeit hin. Die Dienst-

gruppen der HISA können beispielsweise prinzipiell durch die von der OMG spezifizierten Dienste für den Anwendungsbereich des Gesundheitswesens ersetzt werden. Unterschiedliche Details machen jedoch eine formale Überführung sehr schwer.

Für einen formalen Vergleich von Integrationstechniken werden im Teil III verschiedene Ansätze vorgestellt. Sie können genutzt werden, um konkrete Anwendungen von Integrations-techniken in bestimmten Szenarios zu vergleichen. Einen Vergleich unabhängig von einem konkreten Informationssystem kann auf der Basis eines Architektur-Referenzmodells erfolgen. Ein für den Vergleich geeignetes Architektur-Referenzmodell muss die in einem interessierenden Anwendungsbereich typischerweise genutzten Komponenten enthalten. Die für diese Komponenten existierenden Integrationsanforderungen müssen ebenfalls vorliegen. Durch Anpassung des Referenzmodells hinsichtlich der Nutzung der zu vergleichenden Integrationstechniken können Modelle abgeleitet werden, auf deren Basis der Vergleich durchgeführt wird.

Übersicht über Rahmenwerke und einige zugehörige Standards

In den Tabellen 4.2 und 4.3 sind die vorgestellten Rahmenwerke hinsichtlich der verwendeten Begriffsdefinitionen bzw. Metamodelle, der Unterscheidung von Sichtweisen auf die Informationssystemarchitektur und der Bereitstellung von Referenzmodellen gegenübergestellt.

	OSI-Referenzmodell	Referenzmodell für offene verteilte Informationsverarbeitung (RM-ODP)	Object Management Architecture (OMA)	The Open Group Architectural Framework (TOGAF) 7	Healthcare Information Framework (HIF)
Terminologie / Metamodell		Begriffsdefinitionen in ISO/IEC 10746-2 mit Begriffen wie z. B. <ul style="list-style-type: none"> – Objekt – Komposition – Aktion – Schnittstelle 	OMG Object Model mit Begriffen wie z. B. <ul style="list-style-type: none"> – Objekt – Schnittstelle – Operation OMG Component Model mit Begriffen wie z. B. <ul style="list-style-type: none"> – Komponente – Ereignistyp – Ereignisquelle / -senke 		
Sichtweisen		Unterscheidung von sieben Sichtweisen in ISO/IEC 10746-3: <ul style="list-style-type: none"> – Unternehmenssichtweise – Informationssichtweise – rechnerbezogene Sichtweise – Konstruktionsichtweise – technologische Sichtweise 	Unterscheidung von drei Sichtweisen im Model Drive Architecture Guide (MDA Guide): <ul style="list-style-type: none"> – rechnerunabhängige Sicht – plattformunabhängige Sicht – plattformabhängige Sicht 		
Architektur-Referenzmodell	Unterscheidung von 7 Schichten, von denen jede die Funktionalität der darunterliegenden nutzt: <ul style="list-style-type: none"> – Anwendungssystemschicht – Präsentationsschicht – Sitzungsschicht – Transportschicht – Netzwerkschicht – Datenübertragungsschicht – physische Schicht 		OMA Reference Model mit der zentralen Komponente <ul style="list-style-type: none"> – Object Request Broker (ORB) und vier Schnittstellencategorien: <ul style="list-style-type: none"> – Objektdienste – gemeinsame Dienste – Anwendungsbereichsschnittstellen – Anwendungssystemschnittstellen 	TOGAF Technical Reference Model (TOGAF TRM) mit Unterscheidung von drei Ebenen: <ul style="list-style-type: none"> – Anwendungssystemebene – Dienstebene, aufgeteilt auf verschiedene Dienstkategorien – Kommunikationsinfrastrukturebene 	Unterscheidung von 3 Ebenen: <ul style="list-style-type: none"> – Anwendungssystemebene – Middlewareebene – Datenübertmittlungsebene Für die Middlewareebene werden 6 Dienstkategorien spezifiziert: <ul style="list-style-type: none"> – Dienste zum Gegenstand der Gesundheitsversorgung – Dienste zu Gesundheitscharakteristika – Dienste zu Aktivitäten – Dienste zu Ressourcen – Dienste zur Autorisierung – Dienste zu Begriffen
Informations-Referenzmodell			Spezifikationen der Datentypen für die Parameter der Operationen der verschiedenen Schnittstellen; kein Informationsmodell i. e. S.		informationale Spezifikation der Dienstkategorien
Vorgehens-Referenzmodell			Einführung des MDA Patterns und Beschreibung von Modelltransformationen im MDA Guide unter Berücksichtigung der ebenfalls im MDA Guide beschriebenen Sichtweisen	TOGAF Architecture Development Method (TOGAF ADM) mit 7 Phasen: A: Einleitung und Rahmenbedingungen B: Basisbeschreibung C: Zielarchitektur D: Möglichkeiten und Lösungen E: Migrationsplanung F: Implementierung G: Architekturwartung	

Tabelle 4.2: Übersicht über Rahmenwerke zu Informationssystemarchitekturen (Teil 1)

Terminologie / Metamodell	Zachman-Rahmenwerk	Architektur-integrierter Informationssysteme (ARIS)	Enterprise Application Integration (EAI)	The Open Group Architectural Framework (TOGAF) 8	Enterprise Architecture Planning (EAP)
<p>Sichtweisen</p> <ul style="list-style-type: none"> - Organisationsicht - Steuerungssicht - Datensicht - Funktionsicht - Leistungssicht 	<p>Architektur-integrierter Informationssysteme (ARIS)</p> <p>ARIS-Metamodell (= ARIS-Informationssystemmodell) mit Begriffen wie z. B.</p> <ul style="list-style-type: none"> - Organisationseinheit - Funktion - Modul - Informationsobjekt - Relation - Klasse - Ereignis - Nachricht 	<p>Unterscheidung der 5 Sichtweisen</p> <ul style="list-style-type: none"> - Anwendungsbereichsbeschreibung - Unternehmensmodell - Informationssystemmodell - technisches Modell - detaillierte Beschreibung, jeweils unterteilt in - Datenbeschreibung - Prozessbeschreibung - Netzwerkbeschreibung 	<p>Die von verschiedenen Autoren definierten EAI-Ebenen implizieren i. d. R. eine Unterscheidung verschiedener Komponententypen, also jeweils ein Architektur-Referenzmodell. Ebenen nach [LINTHUCUM 2000b]:</p> <ul style="list-style-type: none"> - Daten-EAI (Komponententyp: Datenbank) - Anwendungssystemschnittstellen-EAI (Komponententyp: Anwendungssystem) - Methoden-EAI (Komponententyp: Anwendungssystem) - Benutzungsschnittstellen-EAI (Komponententyp: Benutzungsschnittstelle) 	<p>TOGAF Technical Reference Model (TOGAF TRM) mit Unterscheidung von drei Ebenen:</p> <ul style="list-style-type: none"> - Anwendungssystemebene - Dienstebene, aufgeteilt auf verschiedene Dienstkategorien - Kommunikationsinfrastrukturebene 	
<p>Architektur-Referenzmodell</p>					
<p>Informations-Vorgehens-Referenzmodell</p>	<p>ARIS-Phasenmodell mit 4 sich gegenseitig rückkoppelnden Phasen:</p> <ul style="list-style-type: none"> - Prozessgestaltung - Prozessplanung und -steuerung - Workflowsteuerung - Anwendungssystem(-entwurf) 	<p>Vorgehens-Referenzmodell nach [LINTHUCUM 2000b] mit 12 Schritten:</p> <ul style="list-style-type: none"> - Verstehen des Problembereiches - Zusammenhänge der Daten verstehen - Zusammenhänge der Prozesse verstehen - Ermitteln aller Anwendungssystemschnittstellen - Ermitteln aller Geschäftsereignisse - Ermitteln der Datentransformationsszenarios - Zuordnung der Informationsbewegungen - Anwenden der Technologie - Testen, Testen, Testen - Berücksichtigen der Leistungsfähigkeit - Definieren des Nutzens - Festlegung von Wartungsmaßnahmen 	<p>TOGAF Architecture Development Method (TOGAF ADM) mit 1+8 Phasen:</p> <ul style="list-style-type: none"> - Vorbereitung: Rahmenbedingungen und Grundsätze - A: Architekturvision - B: Unternehmensarchitektur - C: Informationssystemarchitekturen - D: Technologietechnik - E: Möglichkeiten und Lösungen - F: Migrationsplanung - G: Implementierungssteuerung - H: Architekturveränderungsmanagement 	<p>Vorgehens-Referenzmodell nach [SPEWAK and HILL 1992] mit 7 Phasen auf 3 Ebenen:</p> <ul style="list-style-type: none"> • Ebene 1 (Start): Planungsbeginn • Ebene 2 (IST-Zustand): Unternehmensmodellierung - Aktuelle Systeme & Technologie • Ebene 3 (SOLL-Zustand): Datenarchitektur - Anwendungssystemarchitektur - Technologietechnik • Ebene 4 (Migrationsplan): Implementierungs-/Migrationspläne 	

Tabelle 4.3: Übersicht über Rahmenwerke zu Informationssystemarchitekturen (Teil 2)

5 Geschäftsprozessmodellierung — Eine Einführung

5.1 Vorbemerkungen

Dieses Kapitel stellt kurz drei Ansätze zur Geschäftsprozessmodellierung vor: die Methode der Ereignisgesteuerten Prozessketten (EPK), die Business Process Modeling Language (BPML) und das Semantische Objektmodell (SOM). Mit der bisherigen unmittelbaren Fokussierung der Arbeit auf die Informationssystemarchitektur mag dieses „eingeschobene“ Kapitel zunächst überraschen; der Zusammenhang kann jedoch leicht hergestellt werden.

Die in Abschnitt 4.4 vorgestellten unternehmensbezogenen Rahmenwerke beschreiben Beziehungen zwischen Unternehmensaufgaben bzw. Geschäftsprozessen und der Informationssystemarchitektur im engeren Sinn (vgl. Definition zu Architekturmodell im Kasten „Grundbegriffe (5)“ in Abschnitt 3.4). Die Beziehungen können beispielsweise durch Unterscheidung von unternehmensbezogenen und technischen Sichtweisen, z. B. in der Architektur integrierter Informationssysteme (ARIS) und im Zachman-Rahmenwerk, oder durch Unterscheidung von unternehmensbezogenen und informationssystembezogenen Entwicklungsphasen, z. B. im The Open Group Architectural Framework (TOGAF), Version 8, und im Enterprise Application Planning (EAP), ausgedrückt werden. In den unterschiedlichen Werken ist dabei die Verknüpfung der unternehmensbezogenen und der technischen Aspekte unterschiedlich ausführlich beschrieben.

Auch das in dieser Arbeit im Vordergrund stehende Metamodell 3LGM² unterscheidet eine unternehmensbezogene fachliche Ebene von zwei technikbezogenen Werkzeugebenen. Über verschiedene Assoziationsbeziehungen wurden die Beziehungen zwischen den Ebenen formalisiert (Kapitel 6). Bei der Überarbeitung des 3LGM² in Kapitel 7 wird u. a. der Begriff *Ereignistyp* berücksichtigt¹, der sowohl in der Geschäftsprozessmodellierung als auch in Standards für Informationssystemarchitekturen verwendet wird.

5.2 Ereignisgesteuerte Prozessketten

Die Methode der Ereignisgesteuerten Prozessketten (EPK) ist nicht nur, aber auch, durch die enge Verknüpfung mit der ARIS zu einem bekannten Ansatz für die Geschäftsprozessmodellierung geworden ([KELLER et al. 1992], [SCHEER et al. 1995], [LANGNER et al. 1997], [SCHEER 1998b], vgl. Abschnitt 4.4.2).

Grundlage für die Methode der EPK sind die Begriffe

- *Funktion* und
- *Ereignis*.

Das der Methode zugrunde liegende Verständnis der beiden Begriffe sei hier durch die folgenden kurzen Zitate beschrieben:

¹ Der Begriff Ereignistyp ist bereits in der zugrunde gelegten Version 2 des 3LGM² enthalten. Seine Beziehungen zu den anderen Elementen des 3LGM² werden jedoch überarbeitet.

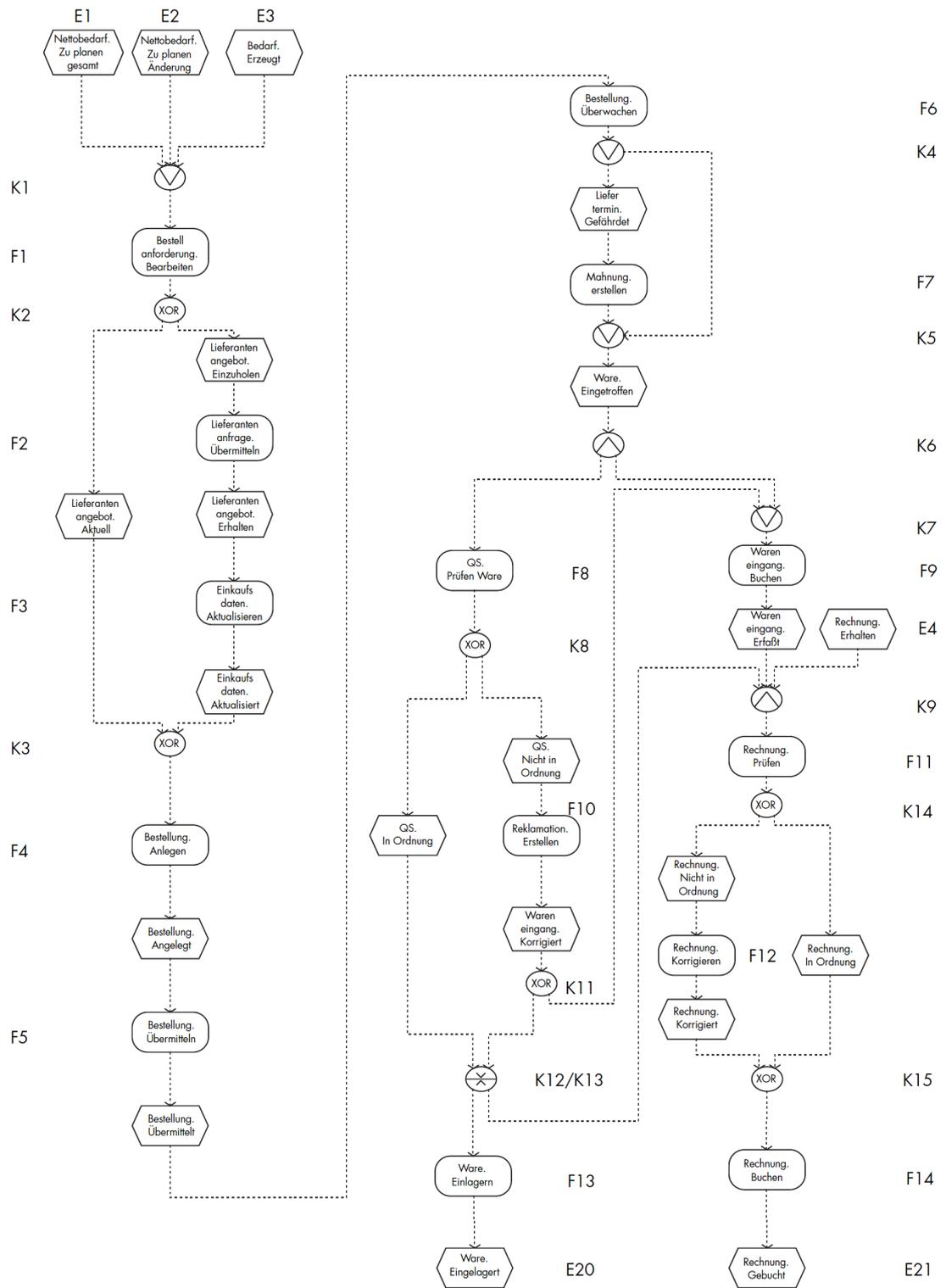


Abbildung 5.1: Beispiel für eine EPK (Quelle: [LANGNER et al. 1997], S. 481, Abb. 1)

„(...) wird der Funktionsbegriff im Sinne der Aufgabe verwendet, d. h. es stellt eine durch physische oder geistige Aktivitäten zu verwirklichende Soll-Leistung dar.“ ([KELLER et al. 1992], S. 8)

„Ein Ereignis ist in Anlehnung an die DIN 69900 das Eintreten eines definierten Zustandes, der eine Folge von Aktivitäten bewirkt.“ ([KELLER et al. 1992], S. 10)

Der Zusammenhang zwischen Funktionen und Ereignissen kann folgendermaßen zusammengefasst werden: Das Beenden einer Funktionsausführung entspricht dem Eintreten eines Ereignisses, welches wiederum Bedingung für das Ausführen nachfolgender Funktionen sein kann. Weitere wichtige Elemente von EPK sind boolesche Konnektoren (Abbildung 5.1).

Die Funktionen und Ereignisse von EPK haben Typcharacter. Die Elemente einer EPK sind also nicht einzelne, bei konkreten Prozessausführungen (Prozessinstanzen) durchgeführte Aktivitäten und eintretende Ereignisse, sondern Typen, die in konkreten Prozessausführungen ausgeprägt werden. Die Ausprägungen können dabei abhängig von den durch einen Prozess verarbeiteten Informationen variieren: Beispielsweise könnten die unterschiedlichen Ereignisse „500 Mullbinden angefordert“ und „1000 Mullbinden angefordert“ Ausprägungen desselben Typs sein. In [KELLER et al. 1992], S. 11 wird dazu ausdrücklich auf den Zusammenhang zwischen den Informationsobjekten in einem Datenmodell für den betrachteten Anwendungsbereich und den Ereignistypen hingewiesen.

5.3 Die Business Process Modeling Language

Die Business Process Modeling Language (BPML) wurde von der Business Process Management Initiative (BPMI) entwickelt. Zur BPML gehören verschiedene bekannte Unternehmen, darunter Adobe Systems, IDS Scheer, Popkin Software, SAP oder SeeBeyond².

Die BPML definiert Begriffe für die Beschreibung von Prozessen und gibt auf der Basis einer XML-ähnlichen Grammatik eine Syntax für die Nutzung der Begriffe an ([BPML WG 2002]). Dokumente zur Prozessbeschreibung werden also in einer XML-ähnlichen Sprache verfasst. Wesentliche Begriffe der BPML sind u. a.

- *Aktivität (activity)*,
- *Aktivitätstyp (activity type)*,
- *Prozess (process)*,
- *Kontext (context)* und
- *Signal (signal)*.

Für Aktivitäten werden verschiedene Typen unterschieden, z. B. *Aktion (action)*, *Zuweisung (assign)* oder *Aufruf (call)*. Die Syntax für Aktivitäten wird beispielsweise folgendermaßen definiert ([BPML WG 2002], S. 13):

```
<{activity type}
  name = NCName
  {other attributes}>
  Content: (documentation?, {other element}*)
</{activity type}>
```

Abbildung 5.2 zeigt ein Beispiel für eine Prozessbeschreibung mit der BPML.

² Der einleitende Text basiert auf den Angaben der BPML, die im Februar 2005 unter der Internet-Adresse www.bpmi.org verfügbar waren.

```

<wsdl:message name="requestMessage">
  <wsdl:part name="orderID" element="type:orderID"/>
  <wsdl:part name="sender" element="type:senderService"/>
  <wsdl:part name="details" element="type:orderDetails"/>
</wsdl:message>
. . .

<wsdl:portType name="exampleServiceType">
  <wsdl:operation name="request">
    <wsdl:input message="srv:requestMessage"/>
  </wsdl:operation>
  <wsdl:operation name="cancel">
    <wsdl:input message="srv:cancelMessage"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:message name="acceptMessage">
  <wsdl:part name="orderID" element="type:orderID"/>
</wsdl:message>
. . .

<bpml:process name="example">
  <bpml:event activity="receiveRequest"/>
  <bpml:context>
    <bpml:property name="orderID" type="type:identifier"/>
    <bpml:property name="customerService" type="inst:service"/>
    <bpml:property name="orderDetails" element="type:orderDetails"/>
    <bpml:property name="invoiceDetails" element="type:invoiceDetails"/>
  </bpml:context>

  <bpml:action name="receiveRequest"
    portType="srv:exampleServiceType" operation="request">
    <bpml:input element="type:orderID"
      property="bp:orderID"/>
    <bpml:input element="type:senderService"
      property="bp:customerService"/>
    <bpml:input element="type:orderDetails"
      property="bp:orderDetails"/>
  </bpml:action>

  <bpml:sequence>
    . . .
    <bpml:action name="sendAcceptance"
      portType="srv:customerServiceType" operation="accept"
      locate="bp:customerService">
      <bpml:output element="type:orderID">
        <bpml:source property="bp:orderID"/>
      </bpml:output>
    </bpml:action>
    . . .
  </bpml:sequence>

  <bpml:action name="sendInvoice" . . .>
    . . .
  </bpml:action>
</bpml:process>

```

Abbildung 5.2: Beispiel für eine Prozessbeschreibung mit der BPML; die mit „wsdl“ beginnenden Definitionen sind WSDL-Definitionen für Dienste (Quelle: [BPMI BPML WG 2002], S. 67-70, Beispiel 5)

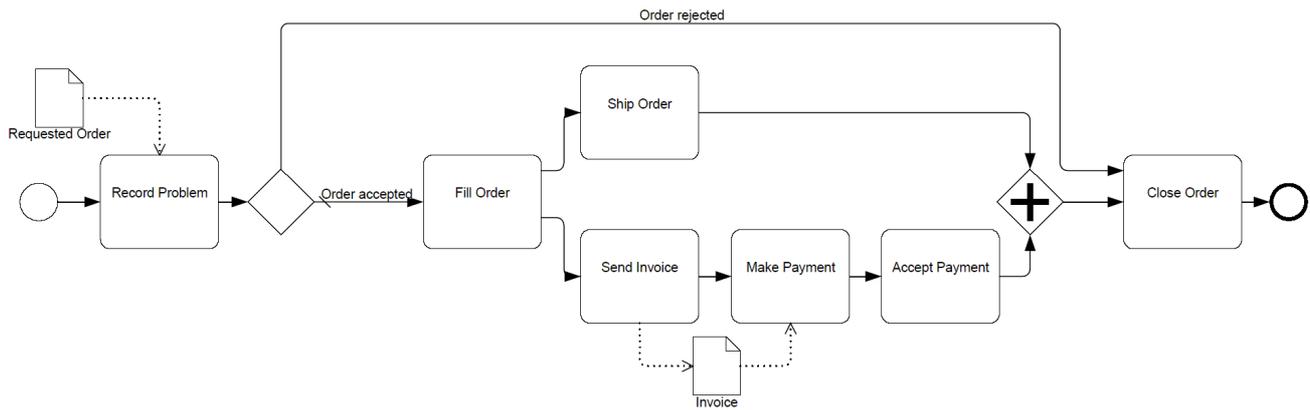


Abbildung 5.3: Beispiel für eine mit der BPMN erstellte Prozessbeschreibung (Quelle: [BPMI NOTATION WG 2004], S. 139, Abb. 87)

Auch die Elemente von BPML-Dokumenten haben wie die Elemente von EPK Typcharakter, auch wenn sie nicht ausdrücklich als Typen bezeichnet werden (vgl. Abschnitt 5.2). In der BPML werden u. a. die Begriffe *Ereignis (event)* und *Signal (signal)* definiert. Der Begriff *Signal* hat dabei eine ähnliche Bedeutung wie der Begriff *Ereignis* in der Methode der Ereignisgesteuerten Prozessketten (vgl. Abschnitt 5.2):

„Signals are used to coordinate the execution of activities executing in the same context. For example, to synchronize the start of one activity with the completion of another activity. Signals are also used to reflect conditions that arise from the execution of activities, and to allow other activities executing in that context to detect and react to these conditions.“ ([BPMI BPML WG 2002], S. 37)

Im Gegensatz zu EPK, bei denen Funktionstypen formal nur über „zwischengeschaltete“ Ereignistypen zu einer Kette zusammengesetzt werden können, sieht die BPML für einfache Zusammenhänge von Funktionen, z. B. Sequenzen, spezielle Sprachelemente vor.

Ereignisse in der BPML beschreiben das Erzeugen von Prozessinstanzen. Das Ereignisattribut eines Prozesses gibt an, ob Instanzen dieses Prozesses durch eine Aktivität, durch eine Nachricht oder durch ein Signal erzeugt werden. Bei Instanziierung durch Nachrichten oder Signale werden mit dem Attribut zusätzliche Aktivitäten angegeben, die die betreffenden Nachrichten bzw. Signale senden bzw. anzeigen ([BPMI BPML WG 2002], S. 20-23).

Die BPML-Spezifikation sieht ausdrücklich vor, dass Dienstdefinitionen, die in der Web Services Description Language (WSDL) vorliegen, in den Prozessbeschreibungen genutzt werden können (Abbildung 5.2, oberer Teil; [BPMI BPML WG 2002], S. 7, [W3C 2004]).

Ergänzend zur BPML wurde von der BPMI bisher die Business Process Modeling Notation (BPMN), eine grafische Sprache für die Erstellung von Geschäftsprozessmodellen, herausgegeben (Abbildung 5.3). Weitere angekündigte Spezifikationen, wie die Business Process Query Language (BPQL), waren zum Zeitpunkt der Erstellung dieser Arbeit noch nicht verfügbar.

5.4 Das Semantische Objektmodell

Das Semantische Objektmodell (SOM) ist ein Ansatz zur prozessorientierten Betrachtung von Unternehmensarchitekturen. Er stellt die Prozessmodellierung zu verschiedenen Ebenen der

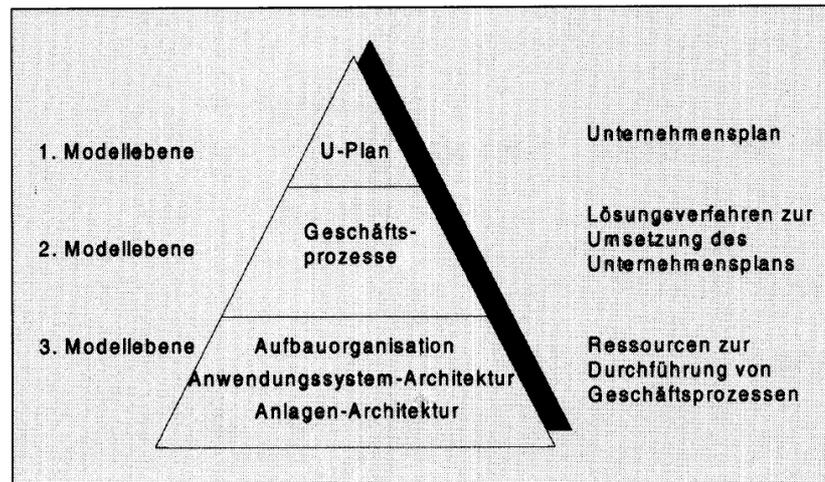


Abbildung 5.4: Ebenen der Unternehmensarchitektur im SOM-Ansatz (Quelle: [FERSTL and SINZ 1995], S. 212, Abb. 2)

Architekturbetrachtung in Beziehung und ist insofern mit der ARIS vergleichbar (vgl. Abschnitte 4.4.2 und 5.2). Die Ausführungen dieses Abschnittes basieren auf [FERSTL and SINZ 1995]³.

Hinsichtlich der Unternehmensarchitektur werden im SOM-Ansatz drei Modellebenen unterschieden (Abbildung 5.4⁴; [FERSTL and SINZ 1995], S. 212):

- der Unternehmensplan, der u. a. das betrachtete betriebliche System von seiner Umwelt abgrenzt, Unternehmensziele beschreibt und Strategien, z. B. die Marktstrategie, bestimmt,
- die Geschäftsprozessmodelle, die Lösungsverfahren für die Erfüllung von Unternehmenszielen beschreiben, und
- die Anwendungssysteme, die als Teil der Ressourcen für die Prozessabwicklung hinsichtlich der Übernahme bestimmter (automatisierbarer) Anteile der Geschäftsprozesse spezifiziert werden.

Die sehr ausführlichen Definitionen und Erläuterungen zur Geschäftsprozessmodellierung in [FERSTL and SINZ 1995] sollen hier nicht wiedergegeben werden. Im Sinne der Abschnitte 5.2 und 5.3 werden die zentralen Begriffe zur Prozessbeschreibung genannt, die in einem Metamodell formal zueinander in Beziehung gesetzt werden (Abbildung 5.5):

- *Aufgabe*,
- *Objekt*,
- *Leistung*,
- *Transaktion* und
- *Ereignis*

Die Begriffe spiegeln die in [FERSTL and SINZ 1995], S. 214 beschriebenen Sichten auf Ge-

³ Eine Einführung in den SOM-Ansatz wird u. a. auch in [FERSTL and SINZ 2001], S. 180-213 gegeben.

⁴ In der Abbildung zur Unternehmensarchitektur werden auf der dritten Ebene neben der Anwendungssystem-Architektur auch die Aufbauorganisation und die Anlagen-Architektur genannt. In den Ausführungen in [FERSTL and SINZ 1995] werden die beiden letzteren nicht näher betrachtet.

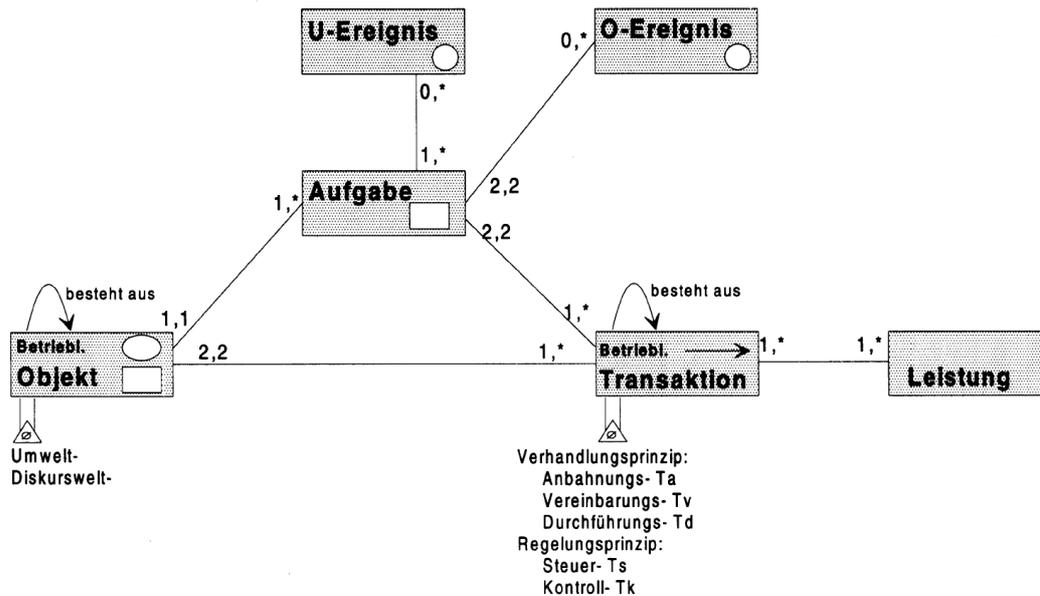


Abbildung 5.5: Metamodell für die Geschäftsprozessmodellierung im SOM-Ansatz (Quelle: [FERSTL and SINZ 1995], S. 216, Abb. 7)

schäftsprozesse wider:

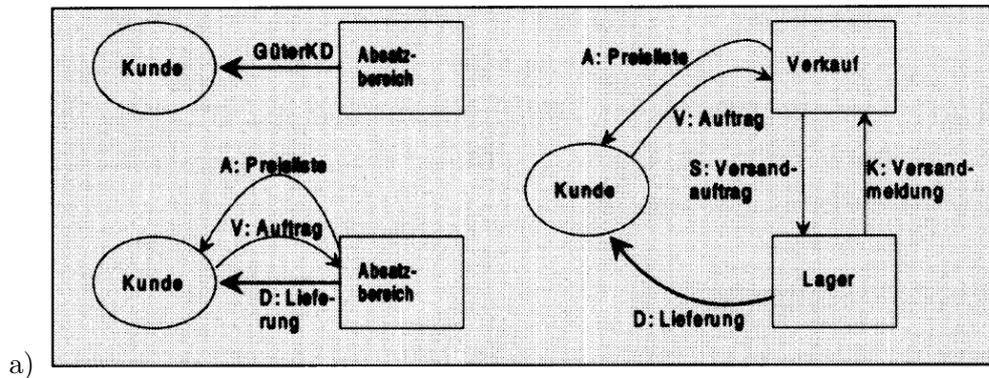
- die Leistungssicht mit dem Schwerpunkt der Beauftragung und Lieferung von Leistungen durch Geschäftsprozesse,
- die Lenkungssicht mit dem Schwerpunkt der Koordinierung von betrieblichen Objekten hinsichtlich der Erstellung und Übergabe von Leistungen und
- die Ablaufsicht mit dem Schwerpunkt der Ereignissteuerung der Aufgaben, die den betrieblichen Objekten zugeordnet sind.

Die ersten beiden Sichten werden dabei als strukturorientiert, die dritte als verhaltensorientiert beschrieben.

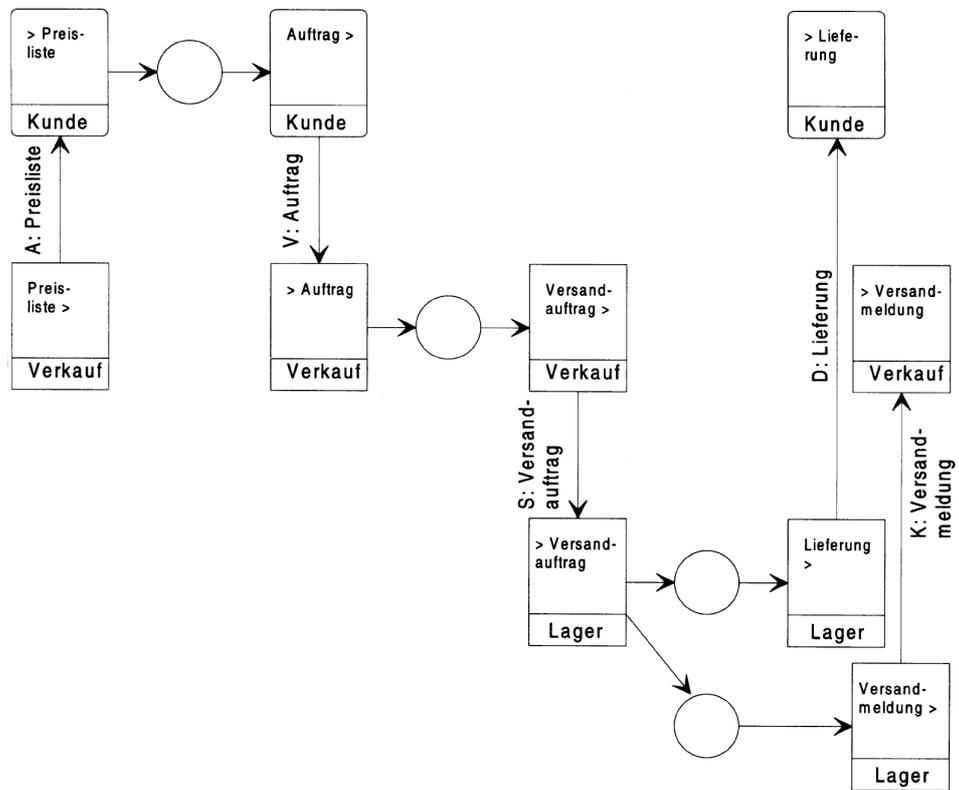
Die Prozessmodellierung auf der Basis des SOM hat, wie die beiden anderen in diesem Kapitel vorgestellten Ansätze, Typcharakter: Es werden nicht einzelne Prozessausführungen betrachtet, sondern die bei allen konkreten Abläufen immer wieder beteiligten Aufgaben, Ereignisse (im Sinne von Ereignistypen) usw.

Der SOM-Ansatz sieht für die strukturorientierte Prozessmodellierung Interaktionsmodelle, für die verhaltensorientierte Modellierung Vorgangs-Ereignis-Modelle vor⁵. In den Interaktionsmodellen werden die oben genannten Begriffe *Objekt* und *Transaktion* verwendet. In den Vorgangs-Ereignis-Modellen werden neben den Begriffen *Objekt* und *Transaktion* insbesondere die Begriffe *Aufgabe* und *Ereignis* verwendet (Abbildung 5.6).

⁵ Ein Modell kann mehrere Schemata, z. B. mehrere Interaktionsschemata, enthalten.



a)



b)

Abbildung 5.6: Beispiel für ein Interaktionsmodell (a) und ein Vorgangs-Ereignis-Modell (b) (Quelle: [FERSTL and SINZ 1995], S. 218-219, Abb. 10 u. 12)

Teil II

Architekturmodellierung

6 Einführung in das 3LGM²

6.1 Vorbemerkungen

Das 3LGM² ist ein Metamodell für die Modellierung von Informationssystemen (vgl. Kasten „Grundbegriffe (4)“ im Kapitel 3). Es definiert Begriffe zur Beschreibung von Informationssystemen und Beziehungen zwischen den Begriffen. Das 3LGM² wurde für die Unterstützung des Managements von Informationssystemen entwickelt, nicht für die Spezifikation von Software ([WINTER et al. 2001], [WINTER et al. 2003], [WENDT et al. 2004]).

Im Vergleich zu anderen Werken, die Begriffe zur Informationssystembeschreibung definieren, gehören zum 3LGM² nur ca. 35 Begriffe. Im Standard *ISO/IEC 10746-2* (vgl. Abschnitt 4.2.2), der die Begriffe für das Referenzmodell für offene verteilte Informationsverarbeitung (RM-ODP) definiert, sind es dagegen ca. 100 Begriffe, im ARIS-Metamodell (vgl. Abschnitt 4.4.2) sind es weit über 100 Begriffe. Durch die im 3LGM² relativ geringe Anzahl der für die Informationssystembeschreibung zulässigen Begriffe ist die Ausdrucksfähigkeit hinsichtlich bestimmter Details von Architekturen beschränkt. Diese möglicherweise zunächst als Nachteil aufgefasste Tatsache kann gleichzeitig als Vorteil hinsichtlich der Erlernbarkeit und der praktischen Anwendbarkeit angesehen werden. Beispiele für die Anwendung des 3LGM² werden in [WENDT et al. 2004] und [BRIGL et al. 2004] beschrieben.

In dieser Arbeit wird das 3LGM² in der Version 2 (3LGM² V2) angewendet und überarbeitet (siehe Kapitel 7 und Abschnitt 9.3.3). Diese Version enthält im Wesentlichen Vereinfachungen im Vergleich zu der in [WINTER et al. 2003] publizierten Version 1. Das 3LGM² in der Version 1 ist durch umfassende Überarbeitung des Metamodells 3LGM, das bereits 1995 publiziert wurde, entstanden ([WINTER and HAUX 1995]).

6.2 Die Ebenen des 3LGM² und ihre Hauptklassen

Die Begriffe des 3LGM² sind auf 3 Ebenen aufgeteilt: Das 3LGM² definiert

- eine **fachliche Ebene**,
- eine **logische Werkzeugebene** und
- eine **physische Werkzeugebene**

mit jeweils unterschiedlichen Klassen. Die Unterscheidung der drei Ebenen reflektiert die verbreitete Unterscheidung von Sichten auf die Informationssystemarchitektur (siehe Kapitel 4). Dabei werden Beziehungen zwischen den Ebenen explizit durch Assoziationsbeziehungen beschrieben. Von den in Kapitel 4 beschriebenen Rahmenwerken enthält nur die ARIS im zugehörigen ARIS-Metamodell vergleichbare Beziehungen.

Dieser Abschnitt (6.2) stellt zunächst die Hauptklassen vor, die folgenden Abschnitte 6.3.1 und 6.3.2 stellen einige weitere Klassen vor, die in dieser Arbeit benötigt werden. Praktisch nützliche Modelle werden selten ausschließlich aus Instanzen der Hauptklassen bestehen. Im Gegenteil: Erst mit der Verwendung der weiteren Klassen werden die in den meisten Anwendungsfällen interessierenden Details modelliert werden können. Die Hauptklassen werden hier

hervorgehoben, da alle weiteren Klassen ohne sie nicht sinnvoll verwendet werden können.

6.2.1 Fachliche Ebene

Auf der fachlichen Ebene werden **Aufgaben**¹ der Informationsverarbeitung sowie **Objekttypen**, die bei der Erfüllung der **Aufgaben** benötigt bzw. erzeugt werden, modelliert (Abbildung 6.1a). **Objekttypen** können als Informationstypen verstanden werden.

Beispiele für **Aufgaben** der Informationsverarbeitung im Krankenhaus sind die Befundung von Röntgenbildern, die Therapieplanung für die Chemotherapie, die administrative Patientenaufnahme oder die Patientenabrechnung. Typische **Objekttypen** sind Patient, Fall, Befund oder Diagnose (siehe auch Abbildung 6.1b).

6.2.2 Logische Werkzeugebene

Auf der logischen Werkzeugebene werden **Anwendungsbausteine** modelliert. Als **Anwendungsbausteine** werden Komponenten eines Informationssystems aufgefasst, die durch die Installation eines **Softwareproduktes** bzw. die (organisatorische) Installation eines **Standardorganisationsplanes** entstehen (Abbildung 6.2a).

Anwendungsbausteine, die durch Installation eines **Softwareproduktes** entstehen, werden nach dem 3LGM² als **rechnerbasierte Anwendungsbausteine** bezeichnet; **Anwendungsbausteine**, die durch Installation eines **Standardorganisationsplanes** entstehen, werden als **papierbasierte Anwendungsbausteine** bezeichnet. Das Wort „papierbasiert“ hat sich dabei in der praktischen Anwendung als die am leichtesten vermittelbare Umschreibung für die Tatsache, dass Personen unter Nutzung verschiedener physischer Hilfsmittel (siehe Abschnitt 6.2.3) und geleitet durch einen **Standardorganisationsplan** bestimmte **Aufgaben** erfüllen, erwiesen.

Beispiele für **rechnerbasierte Anwendungsbausteine** sind ein digitales Bildarchiv oder ein Patientenverwaltungssystem; ein Beispiel für einen **papierbasierten Anwendungsbaustein** ist ein Patientenkarteiverwaltungssystem in einer Ambulanz (siehe auch Abbildung 6.2b).

Die Bezeichnung *Anwendungsbaustein (application component)* spiegelt eine Informationssystembetrachtung wider, die die mehrfache Verwendung von Komponenten eines Informationssystems bei der Implementierung von Anwendungssystemen berücksichtigt. **Anwendungsbausteine** können zu komplexen Anwendungssystemen zusammengesetzt werden, wobei die **Anwendungsbausteine**, die mehrfach benötigte Funktionalitäten bereitstellen, für die Zusammensetzung mehrerer Anwendungssysteme verwendet werden können. Damit wird u. a. eine Annäherung an Dienst- bzw. Schnittstellenkategorien, wie sie z. B. im OMA Reference Model (vgl. Abschnitt 4.2.3) oder im TOGAF Technical Reference Model (TOGAF TRM) (vgl. Abschnitt 4.2.5) definiert werden, erreicht.

Anwendungsbausteine können über **Anwendungsbausteinkonfigurationen** zusammengefasst und den **Aufgaben** der fachlichen Ebene zugeordnet werden (Abbildung 6.2a). Eine **Anwendungsbausteinkonfiguration** ist eine Menge: Sie enthält bestimmte **Anwendungsbausteine**, die gemeinsam das Erledigen einer bestimmten **Aufgabe** unterstützen. Eine einfache Visualisierung der Zuordnung von **Anwendungsbausteinen** zu **Aufgaben** zeigt Abbildung 6.4.

¹ Klassennamen des 3LGM² sind im Text in einer anderen Schriftart gesetzt.

6.2 Die Ebenen des 3LGM² und ihre Hauptklassen

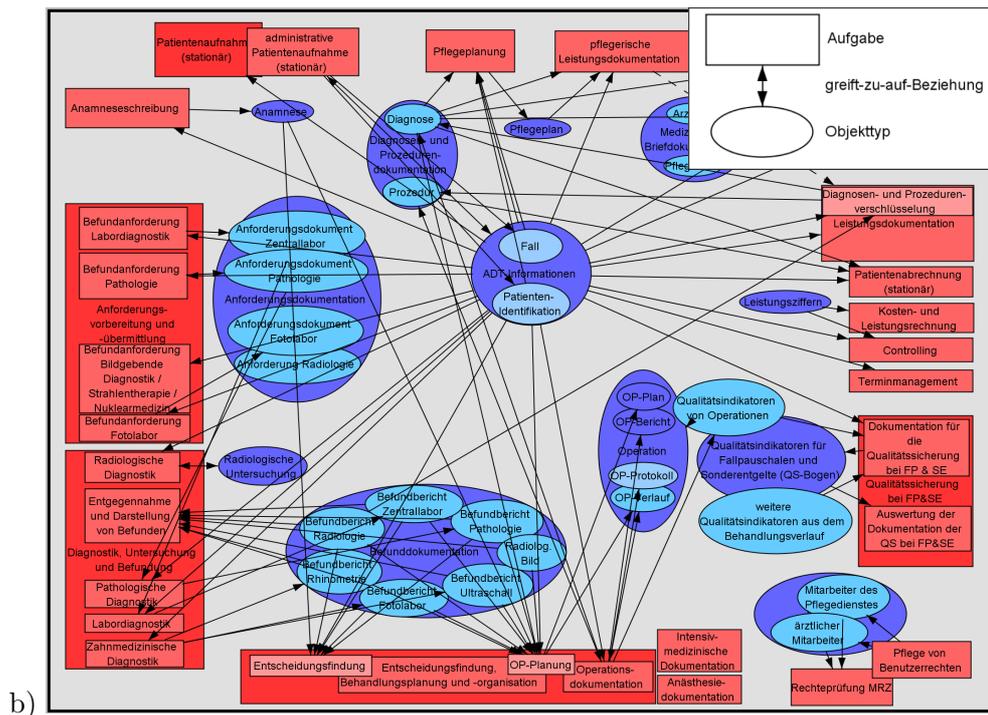
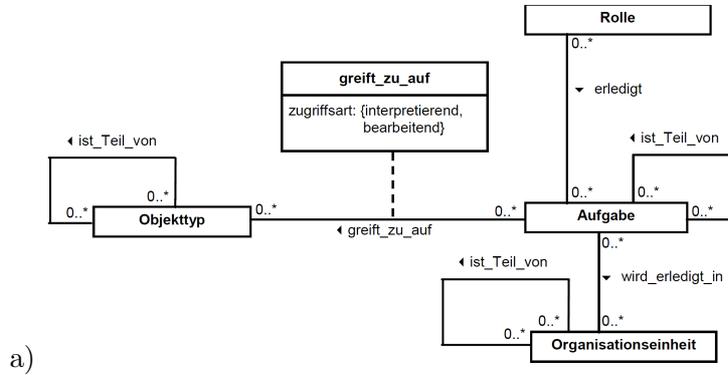


Abbildung 6.1: Spezifikationen der fachlichen Ebene des 3LGM² mit der UML (vgl. [WINTER et al. 2003], S. 546, Abb. 1) (a) und Auszug der fachliche Ebene des Informationssystems des UKL (b)

6 Einführung in das 3LGM²

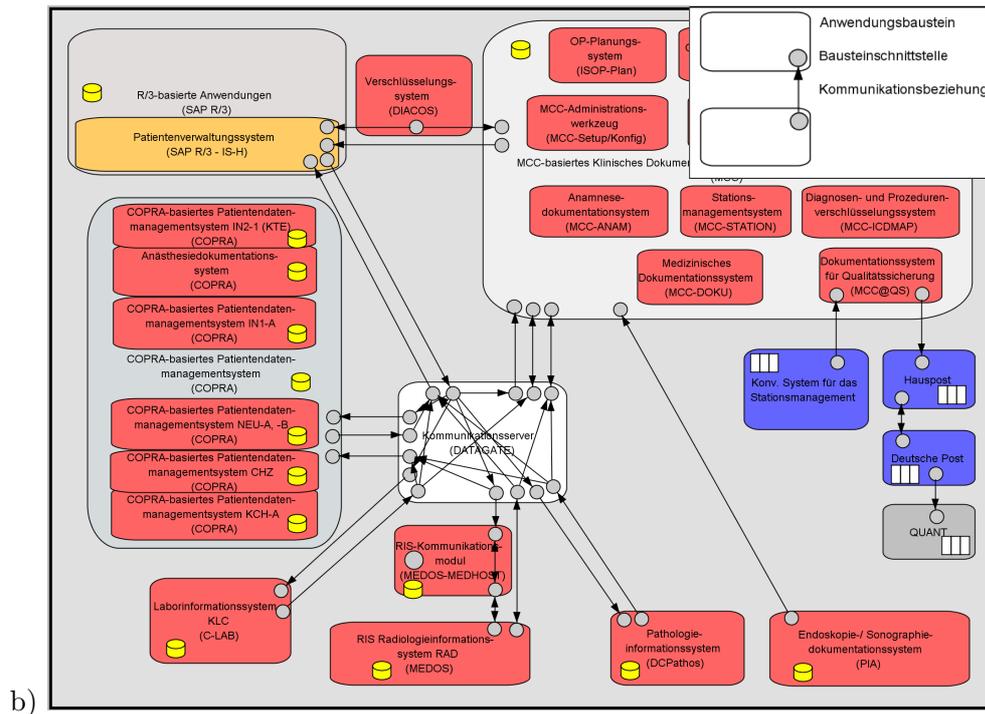
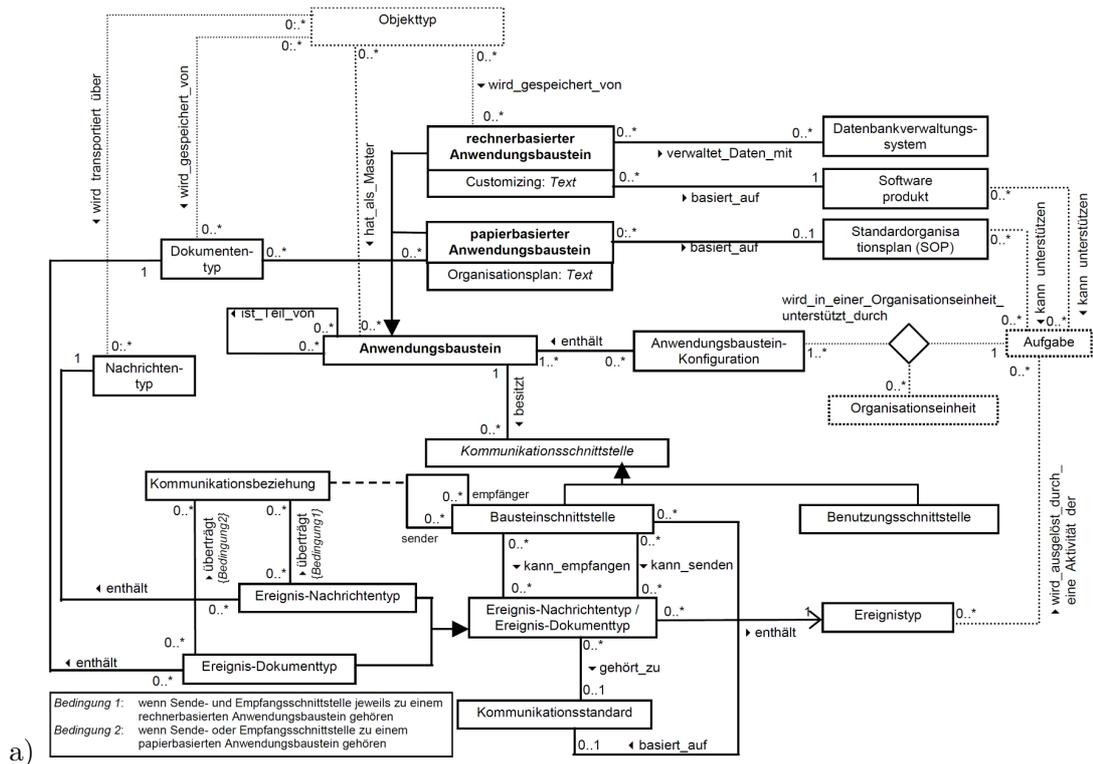


Abbildung 6.2: Spezifikationen der logischen Werkzeugebene des 3LGM² mit der UML (vgl. [WINTER et al. 2003], S. 547, Abb. 3) (a) und Auszug der logischen Werkzeugebene des Informationssystems des UKL (b)

6.2 Die Ebenen des 3LGM² und ihre Hauptklassen

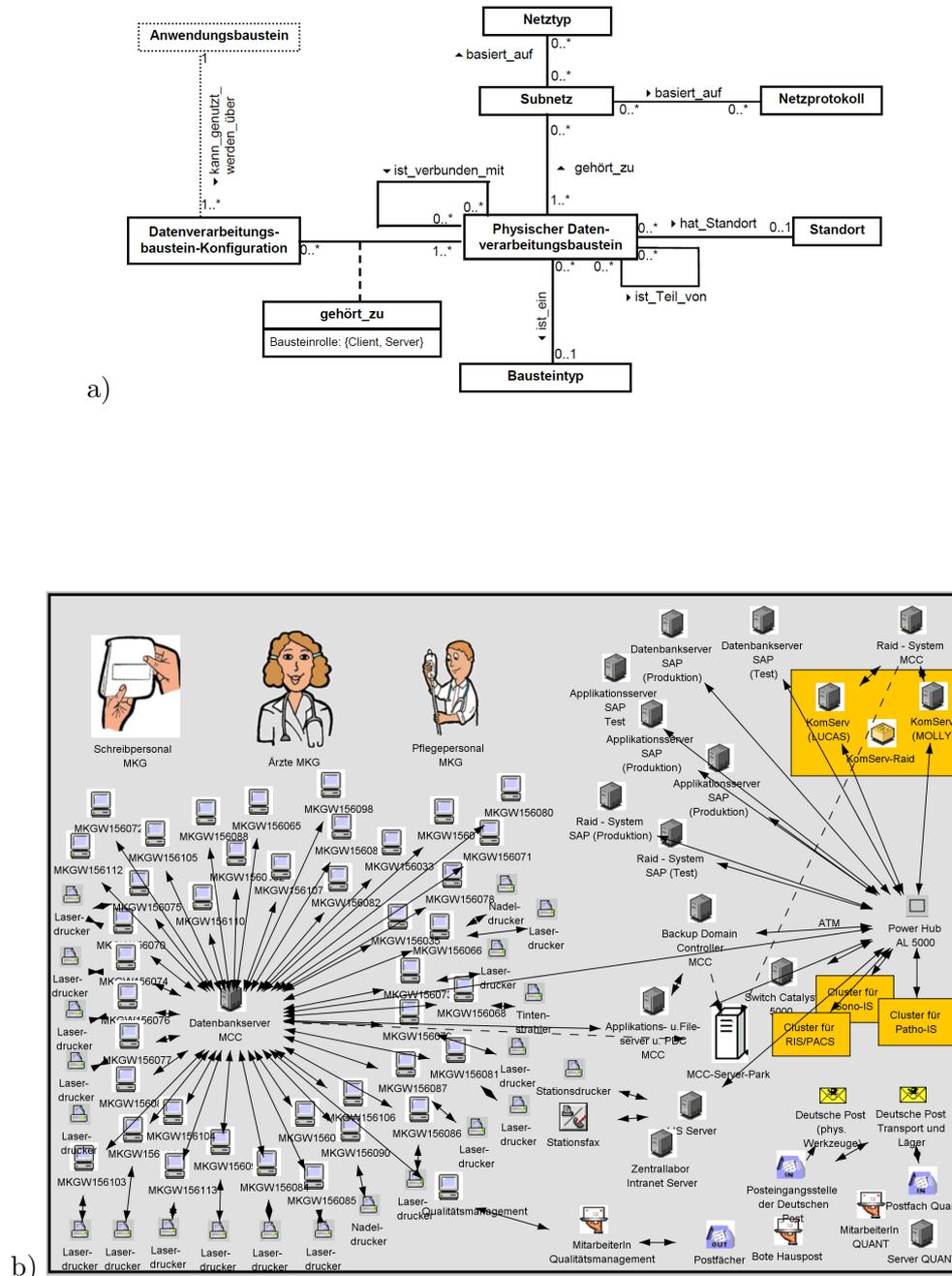


Abbildung 6.3: Spezifikationen der physischen Werkzeugebene des 3LGM² mit der UML (vgl. [WINTER et al. 2003], S. 548, Abb. 5) (a) und Auszug der physischen Werkzeugebene des Informationssystems des UKL (b)

6.2.3 Physische Werkzeugebene

Auf der physischen Werkzeugebene werden konventionelle **physische Datenverarbeitungsbausteine** wie Karteischränke oder Transportfahrzeuge sowie elektronische **physische Datenverarbeitungsbausteine** wie Serversysteme oder PCs und ihre Netzverbindungen modelliert (Abbildung 6.3a und b).

Physische Datenverarbeitungsbausteine können über **physische Datenverarbeitungsbausteinkonfigurationen** zusammengefasst und den **Anwendungsbausteinen** der logischen Werkzeugebene zugeordnet werden (Abbildung 6.3a). Eine **physische Datenverarbeitungsbausteinkonfiguration** ist eine Menge von **physischen Datenverarbeitungsbausteinen**, auf denen ein bestimmter **Anwendungsbaustein** installiert ist. Eine einfache Visualisierung der Zuordnung von **physischen Datenverarbeitungsbausteinen** zu **Anwendungsbausteinen** zeigt Abbildung 6.4.

6.3 Klassen für die Integrationsmodellierung

6.3.1 Klassen für die Modellierung der Speicherung von Informationen

Die hier vorgestellten Klassen zur Modellierung der Speicherung von Informationen werden nicht nur für die Integrationsmodellierung verwendet. Sie können u. a. verwendet werden, um die Aufteilung von Daten auf Anwendungssysteme und die Merkmale der verwendeten Datenbanksysteme zu beschreiben. Die Beschreibung der Aufteilung von Daten auf Anwendungssysteme ist eine wesentliche Grundlage der Integrationsbewertung in Kapitel 9. Die dafür vorgesehenen Klassen werden daher hier in einem Unterabschnitt zum Abschnitt „Klassen für die Integrationsmodellierung“ vorgestellt.

Zur Modellierung der Speicherung von Informationen enthält das 3LGM² die Klassen (Abbildung 6.2a)

- Anwendungsbaustein,
- Dokumententyp und
- Datenbankverwaltungssystem.

Die Speicherung von Daten in einem rechnerbasierten **Anwendungsbaustein** kann durch ein **Datenbankverwaltungssystem** gesteuert werden. Einem papierbasierten **Anwendungsbaustein** können **Dokumententypen** zugeordnet werden, die zur Speicherung von Informationen verwendet werden.

Die Klasse **Objekttyp** auf der fachlichen Ebene ist mit den Klassen **rechnerbasierter Anwendungsbaustein** und **Dokumententyp** über die Assoziationsbeziehungen **wird_gespeichert_von** verbunden.

An dieser Stelle sei die Assoziationsbeziehung **hat_als_Master** zwischen **Objekttypen** und **Anwendungsbausteinen** hervorgehoben. Mit ihr kann ausgedrückt werden, ob ein **Objekttyp**, der in mehreren **Anwendungsbausteinen** gespeichert wird, einen sogenannten führenden **Anwendungsbaustein** hat. Die Benennung eines führenden **Anwendungsbausteines** für bestimmte **Objekttypen** ist dann wichtig, wenn zur Wahrung einer bestimmten Autonomie von **Anwendungsbausteinen** akzeptiert wird, dass Abweichungen zwischen den in den **Anwendungsbausteinen** gespeicherten Ausprägungen der betreffenden **Objekttypen** entstehen. Aktualisierungen von gespeicherten Daten müssen dann immer im **Master-Anwendungsbaustein** des

6.3 Klassen für die Integrationsmodellierung

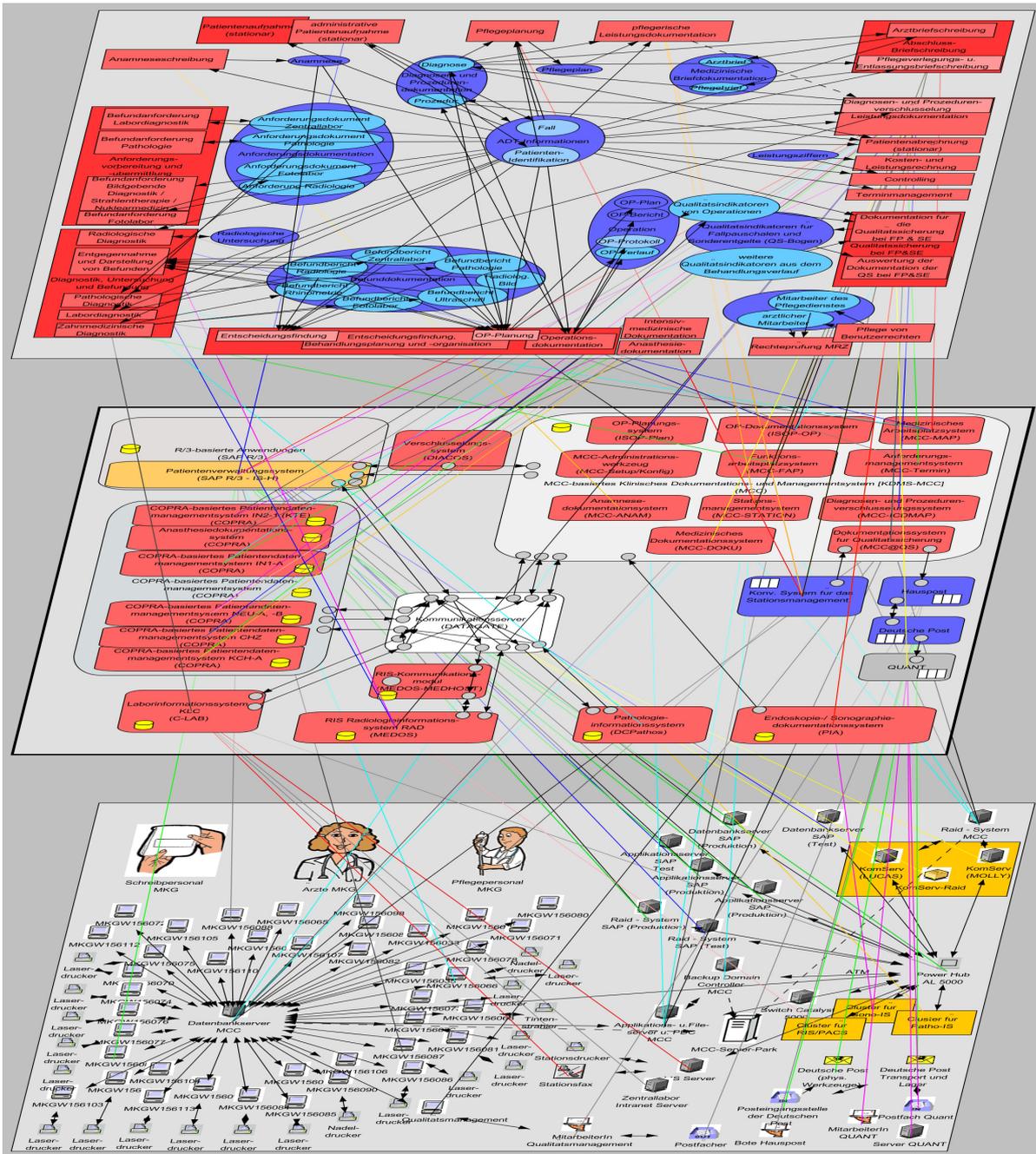


Abbildung 6.4: Auszug des Informationssystems des UKL in Drei-Ebenen-Darstellung

betreffenden Objekttyps erfolgen und von dort ausgehend, mit oder ohne Zeitversatz bzw. Akzeptanz von Abweichungen, an die anderen Anwendungsbausteine verteilt werden.

HL7-Ereignistyp	Beschreibung (Auszüge aus dem Standard)
A01	„ 3.2.1 ADT/ACK - admit / visit notification (event A01) An A01 event is intended to be used for “Admitted” patients only. An A01 event is sent as a result of a patient undergoing the admission process which assigns the patient to a bed. It signals the beginning of a patient’s stay in a healthcare facility. Normally, this information is entered in the primary ADT system and broadcast to the nursing units and ancillary systems. (...)“
A02	„ 3.2.2 ADT/ACK - transfer a patient (event A02) An A02 event is issued as a result of the patient changing his or her assigned physical location. (...)“
A03	„ 3.2.3 ADT/ACK - discharge/end visit (event A03) An A03 event signals the end of a patient’s stay in a healthcare facility. It signals that the patient’s status has changed to “discharged” and that a discharge date has been recorded. The patient is no longer in the facility.“
A08	„ 3.2.8 ADT/ACK - update patient information (event A08) This trigger event is used when any patient information has changed but when no other trigger event has occurred. For example, an A08 event can be used to notify the receiving systems of a change of address or a name change. (...)“

Tabelle 6.1: Beispiele für Ereignistypen im Kommunikationsstandard HL7 (Quelle: [HL7 1999])

6.3.2 Klassen für die Modellierung der Übermittlung von Informationen

Die im Folgenden vorgestellten Klassen erlauben eine einfache Integrationsmodellierung und bilden eine wichtige Grundlage für die in Kapitel 7 unternommene Überarbeitung des 3LGM² hinsichtlich komponentenbasierter Architekturmodellierung und für die Integrationsbewertung in den Kapiteln 9 und 11.

Zur Modellierung der Übermittlung von Informationen enthält das 3LGM² die Klassen (Abbildung 6.2a)

- Benutzungsschnittstelle,
- Bausteinschnittstelle,
- Kommunikationsbeziehung,
- Ereignistyp,
- Nachrichtentyp,
- Dokumententyp und
- Kommunikationsstandard.

Die Klasse **Benutzungsschnittstelle** kann verwendet werden, um Kommunikation, d. h. Übermittlung von Informationen, zwischen **Anwendungsbausteinen** und Benutzern auszudrücken.

Der Datenaustausch zwischen **Anwendungsbausteinen** wird durch das Zuweisen von **Bausteinschnittstellen** zu **Anwendungsbausteinen** und das Verknüpfen der **Bausteinschnittstellen** über Assoziationen der Klasse **Kommunikationsbeziehung** modelliert. Zu einer **Kommunikationsbeziehung** gehören also immer zwei **Bausteinschnittstellen**, je eine am sendenden und eine am empfangenden **Anwendungsbaustein**. Diese Definition entspricht den praktischen Realisierungen der meisten **Kommunikationsbeziehungen**, bei denen i. d. R. zwei Schnittstellenmodule gekauft bzw. selbst programmiert werden müssen.

Bausteinschnittstellen können über **Nachrichtentypen** und deren Bezug zu bestimmten **Ereignistypen** oder **Dokumententypen** näher beschrieben werden (Tabelle 6.1 und Abbildung 6.5). Dazu gehört auch, sofern sinnvoll, der zugrunde liegende **Kommunikationsstandard**.

Die bereits in Abschnitt 6.3.1 vorgestellte Klasse **Dokumententyp** wird nicht nur für die Beschreibung papierbasierter Speicherung verwendet, sondern auch für die Beschreibung pa-

pierbasierter Kommunikation.

6.4 Prozessmodellierung

Auf der Basis der in [BRIGL et al. 2003] beschriebenen Ergänzung des 3LGM² kann eine einfache Modellierung von informationsverarbeitenden Prozessen unternommen werden. Sie ist eine sehr „vorsichtige“ Annäherung an die mächtigen Methoden und Werkzeuge der Geschäftsprozessmodellierung, die dadurch nicht ersetzt werden sollen (vgl. Kapitel 5). Auch hier steht, im Sinne der Ausführungen in Abschnitt 6.1, die Beschränkung der Theorie und des zugehörigen Vokabulars im Vordergrund.

Ein Informationsprozess ist eine Sequenz von Aufgaben, die bestimmte Bedingungen hinsichtlich der Interpretation und Bearbeitung von Objekttypen erfüllen. Jede Aufgabe in einem Informationsprozess, die nicht die erste Aufgabe dieses Informationsprozesses ist, muss einen Objekttyp interpretieren, den eine vorhergehende Aufgabe desselben Informationsprozesses bearbeitet.

Aus Informationsprozessen kann über die Beziehungen zwischen den Elementen der fachlichen Ebene und der logischen Werkzeugebene abgeleitet werden, zwischen welchen Anwendungsbausteinen Objekttypen, d. h. Informationen, übermittelt werden müssen. Eine entsprechende Sequenz von Anwendungsbausteinen inkl. der beteiligten Bausteinschnittstellen wird als Kommunikationsprozess bezeichnet (Abbildung 6.6).

6.5 Das 3LGM² und Architekturstile

Mit den Ausführungen der Abschnitte 3.3 und 3.4 sowie der vorhergehenden Abschnitte dieses Kapitels kann das 3LGM² als semantisches Modell für Architekturstile aufgefasst werden. Die folgenden Ausführungen erläutern, basierend auf Abschnitt 3.3, diesen Gedanken.

Eine Syntax zur Erstellung von 3LGM²-basierten Modellen wurde bisher nicht explizit definiert. Dem am Ende von Abschnitt 6.1 genannten ursprünglichen 3LGM wurde eine graphenbasierte Notierung von Modellen zugrunde gelegt, über die, vereinfacht ausgedrückt, Komponenten als Knoten und Konnektoren als Kanten eines Graphen notiert wurden.

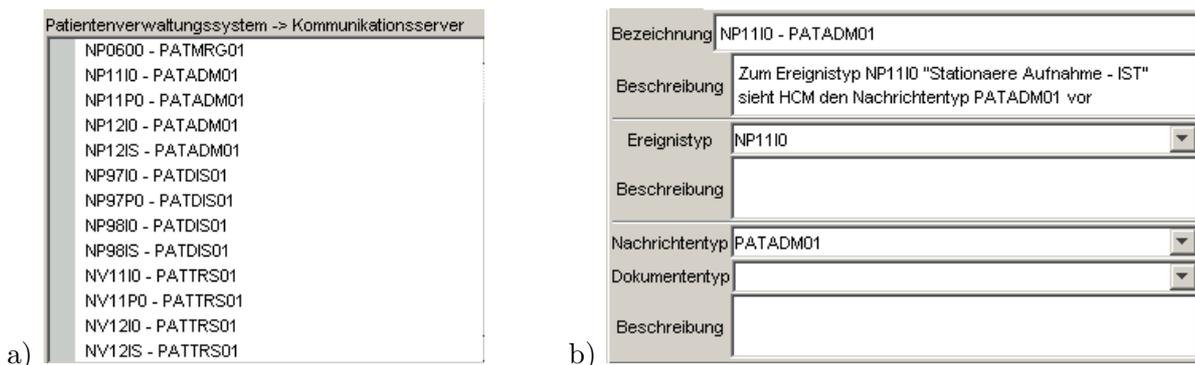


Abbildung 6.5: Kombinationen von Ereignistypen und Nachrichtentypen zur Beschreibung der Kommunikation zwischen den Anwendungsbausteinen Patientenverwaltungssystem und Kommunikationsserver am UKL

6 Einführung in das 3LGM²

Sowohl das zum ursprünglichen 3LGM als auch das zum 3LGM² entwickelte Modellierungswerkzeug präsentieren Modelle grafisch, basierend auf einer definierten Menge von Symbolen (vgl. die Abbildungen von Modellbeispielen in diesem Kapitel, z. B. Abbildung 6.2b). Für die Elemente, die für eine bessere Übersicht nicht grafisch dargestellt werden, stellen sie Listen zur Verfügung (vgl. z. B. Abbildung 6.5). Das Erstellen von Konfigurationen im Sinne von Abschnitt 3.3 erfolgt unmittelbar durch das Erzeugen und Verknüpfen von grafischen Elementen bzw. durch das Erzeugen von Listeneinträgen.

Die Semantik der grafischen Elemente und Listeneinträge wird durch das 3LGM² festgelegt. Die Klassennamen und die Namen der Assoziationsbeziehungen stehen für bestimmte Begriffe, die in Publikationen wie [WINTER et al. 2003] oder [WENDT et al. 2004] erläutert werden. Die Klassen im 3LGM² entsprechen Komponententypen, die Assoziationsbeziehungen im 3LGM² entsprechen Konnektorentypen. Durch die Verknüpfung der 3LGM²-Klassen über die Assoziationsbeziehungen sind die zulässigen Konfigurationen festgelegt.

Wenn Architekturmodellierung als Architekturmodellierung im weiteren Sinn entsprechend Kasten „Grundlagen (5)“ im Abschnitt 3.3 betrachtet wird, dann kann das vollständige 3LGM² als ein semantisches Modell für Architekturstile aufgefasst werden. Legt man die Definition für Architekturmodellierung im engeren Sinn zugrunde, dann entsprechen die 3LGM²-Klassen **rechnerbasierter Anwendungsbaustein** und **papierbasierter Anwendungsbaustein** den zulässigen Komponententypen und die Assoziationsbeziehung **Kommunikationsbeziehung** steht für den einzigen zulässigen Konnektorentyp. Die übrigen Klassen und Assoziationsbeziehungen der logischen Werkzeugebene dienen dann der näheren Beschreibung der Ausprägungen der Komponenten- und Konnektorentypen.

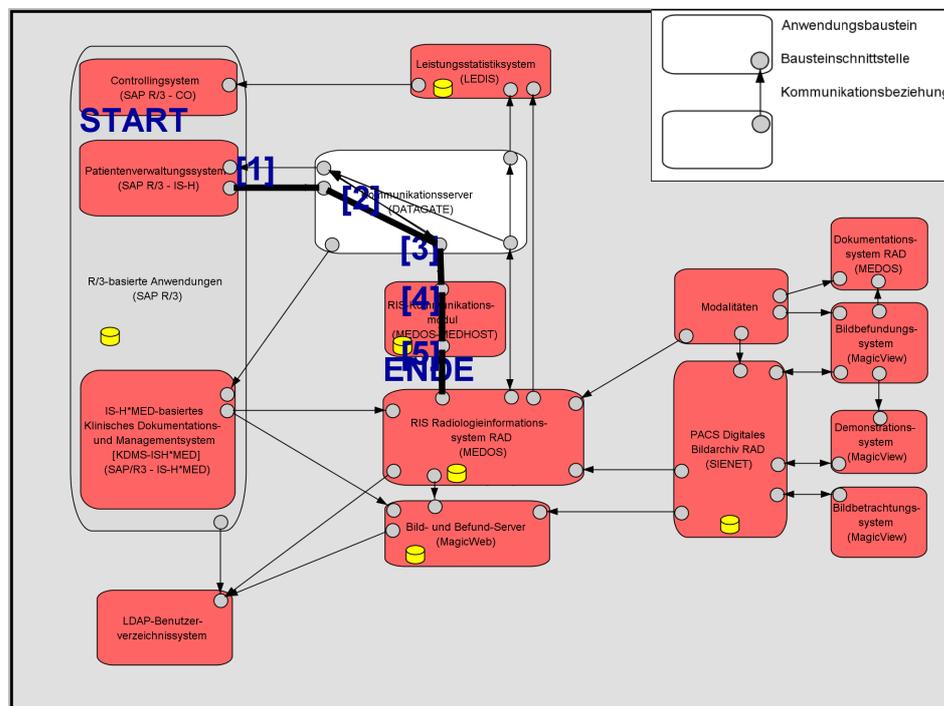


Abbildung 6.6: Ein Kommunikationsprozess zur Übermittlung von ADT-Informationen im Informationssystem des UKL

Das 3LGM² als Architekturstil

Das 3LGM² wurde als generisches Metamodell entwickelt, das nicht auf einen bestimmten Architekturstil zugeschnitten ist. Die hier vorgestellte Version 2 ist jedoch besonders für solche Architekturen geeignet, deren Komponenten durch Austausch von Nachrichten miteinander kommunizieren (vgl. Abschnitt 6.3.2 und Abbildung 6.2a): Eine ausdrückliche Angabe, ob bestimmte **Kommunikationsbeziehungen** synchron oder asynchron arbeiten, ist nicht möglich. Eine ausdrückliche Benennung von **Nachrichtentypen**, die als Anfragen übermittelt werden, und **Nachrichtentypen**, die als Antwort zurückübermittelt werden, sowie ihre Verknüpfung zu Anfragetyp-Anworttyp-Paaren ist ebenfalls nicht möglich.

Durch die Beschränkungen in der Ausdruckfähigkeit des 3LGM² können u. U. nicht alle möglicherweise für das Informationsmanagement relevanten Architekturen angemessen modelliert werden. „Bevorzugt“ werden Architekturen mit nachrichtenbasierter Kommunikation ohne Zusammenhänge der übermittelten Nachrichten. Beispielsweise können Zusammenhänge im Sinne von Parameter-Ergebnis-Paaren nicht oder nur sehr aufwendig ausgedrückt werden. Damit kann das 3LGM² als semantisches Modell für einen bestimmten Architekturstil betrachtet werden.

Durch die Einführung der Klasse **Operation** in das 3LGM² im Abschnitt 7.2.2 des folgenden Kapitels wird eine flexiblere Kommunikationsmodellierung möglich.

7 Flexible Architekturmodellierung — Überarbeitung des 3LGM²

7.1 Begriffe für die Modellierung von Komponenten auf der Basis des 3LGM²

In den Abschnitten 3.2 und 3.3 wurde der Komponentenbegriff als zentrales Element von Architekturen und Architekturstilen herausgearbeitet. In Abschnitt 6.5 wurde u. a. der Zusammenhang zwischen Komponententypen und 3LGM²-Klassen beschrieben.

Für die weiteren Ausführungen wird die Definition für *Architekturmodell im engeren Sinn* zugrunde gelegt (vgl. Kasten „Grundbegriffe (5)“ in Abschnitt 3.4). Die mit dem 3LGM² modellierbaren Komponententypen werden also durch die Klassen **rechnerbasierter Anwendungsbaustein** und **papierbasierter Anwendungsbaustein** festgelegt (vgl. Abschnitt 6.5).

Standards wie das Referenzmodell für offene verteilte Informationsverarbeitung (RM-ODP), insbesondere Teil 2 „Foundations“, das OMG Object Model oder das OMG Component Model, die Sprachen für die Beschreibung von interagierenden Komponenten definieren, legen eine Vielzahl von Begriffen fest (vgl. Abschnitte 4.2.2 und 4.2.3). Viele davon werden für eine ausführliche Spezifikation, die dann in eine Implementierung mündet, benötigt. Das 3LGM² wurde nicht für die ausführliche Softwarespezifikation entwickelt, sondern für die Unterstützung des Managements von Informationssystemen. Dabei wurde auch eine Beschränkung des durch das 3LGM² bereitgestellten Vokabulars angestrebt, um die Anwendung zu erleichtern (vgl. Abschnitt 6.1).

Hauptsächlich aus den genannten Standards RM-ODP, OMG Object Model und OMG Component Model sowie aus der Methode der Ereignisgesteuerten Prozessketten (vgl. Abschnitt 5.2) wurden die folgenden Begriffe ausgewählt, die bei der Architekturbeschreibung mit Hilfe des 3LGM² zur Verfügung stehen bzw. korrespondierende Begriffe haben sollten:

- *Objekt* bzw. *Komponente*,
- *Ereignistyp*,
- *Schnittstelle* and
- *Operation*.

Wie bereits beschrieben, werden ausführbare oder handelnde Komponenten als **Anwendungsbausteine** modelliert. Die für die Modellierung der Interaktion von **Anwendungsbausteinen** unternommene Überarbeitung des 3LGM² unter Berücksichtigung der genannten Begriffe wird im folgenden Abschnitt ausführlicher vorgestellt.

Das überarbeitete 3LGM² wird im Folgenden mit 3LGM_A² bezeichnet. Das A weist auf die Überarbeitung hinsichtlich der Architekturmodellierung hin.

7.2 Überarbeitung des 3LGM²

7.2.1 Ereignistypen

Der Begriff *Ereignistyp* hat eine große Bedeutung für die Modellierung von Prozessen. Er ist bereits in der dieser Arbeit zugrunde gelegten Version 2 des 3LGM² als Klasse enthalten. Seine

Einbettung in das 3LGM², d. h. seine Assoziationsbeziehungen zu anderen Klassen, wird hier überarbeitet.

Die Begriffe *Ereignistyp*, *Ereignisquelle* und *Ereignissenke* sind, neben anderen Begriffen, wesentliche Elemente des OMG Component Models zur Beschreibung der Steuerung der Interaktion von Komponenten (vgl. Abschnitt 4.2.3).

Das RM-ODP enthält die Begriffe *Ereignis* und *Ereignistyp* nicht. Für das Benennen des initiierenden Objektes und des reagierenden Objektes bzgl. einer bestimmten Kommunikation werden die Rollen *initiierendes Objekt* (*initiating object*) und *antwortendes Objekt* (*responding object*) definiert, für das Benennen von Informationsquellen die Rolle *produzierendes Objekt* (*producer object*) und für das Benennen von Informationssenzen die Rolle *konsumierendes Objekt* (*consumer object*). Zusätzlich werden die Rollen *Client-Objekt* (*client object*) und *Server-Objekt* (*server object*) definiert ([ISO/IEC JTC 1 1996b], S. 13, Definitionen 13.3.1-13.3.6).

Ereignistypen und Prozessmodellierung

An dieser Stelle wird die in Kapitel 5 kurz vorgestellte „Modellierungswelt“ der Geschäftsprozessmodellierung in die Überarbeitung einbezogen. Bei der Vorstellung in Kapitel 5 wurde u. a. die Bedeutung der Begriffe *Ereignis* und *Ereignistyp* hervorgehoben.

In Kommunikationsstandards wie HL7 werden die Begriffe *Ereignis* und *Ereignistyp* ebenfalls verwendet. In HL7 werden Ereignistypen im Zusammenhang mit Anwendungsfällen (use cases) definiert ([HL7 1999]). Die Anwendungsfallbeschreibungen können als einfache Prozess-Referenzmodelle betrachtet werden. Über die im Standard angewendete unmittelbare Zuordnung von Nachrichtentypen zu Ereignistypen wird eine Verknüpfung zwischen der anwendungssystemunabhängigen Prozessbetrachtung und der Betrachtung der Kommunikation von Anwendungssystemen hergestellt.

Ereignistypen im 3LGM²

In Abschnitt 6.4 wurden kurz die Möglichkeiten der Modellierung von Informations- und Kommunikationsprozessen auf der Basis des 3LGM² beschrieben. Bisher wurden **Ereignistypen** auf der logischen Werkzeugebene des 3LGM² modelliert und konnten über die Assoziationsbeziehung `wird_ausgelöst_durch_eine_Aktivität_der_Aufgaben` zugeordnet werden. Dabei konnte nicht unmittelbar beschrieben werden, ob der **Ereignistyp** in Zusammenhang mit dem Zugriff auf alle mit der betreffenden **Aufgabe** assoziierten **Objekttypen** steht oder nur mit dem Zugriff auf eine bestimmte Teilmenge dieser **Objekttypen**. Über die Assoziation mit **Nachrichten-** und **Dokumententypen**, die wiederum mit **Objekttypen** assoziiert sind, konnten die Beziehungen zu **Objekttypen** mittelbar hergestellt werden. Dieser „Umweg“ entspricht jedoch nicht der Bedeutung von **Ereignistypen** für die fachliche Ebene.

Die fachliche Ebene des 3LGM² wird hier durch das „Verschieben“ der Klasse **Ereignistyp** und die Ergänzung von Assoziationsbeziehungen zur `greift_zu_auf`-Beziehung überarbeitet (Abbildung 7.1).

Die Assoziationsbeziehungen `löst_aus` und `wird_ausgelöst_durch` sind unmittelbar für die Integration von **Anwendungsbausteinen** von Bedeutung:

1. Löst ein **Ereignistyp** einen Zugriff auf **Objekttypen** aus, wird also eine Aktivität einer **Aufgabe** ausgelöst, dann muss die zugehörige Funktionalität entweder unmittelbar

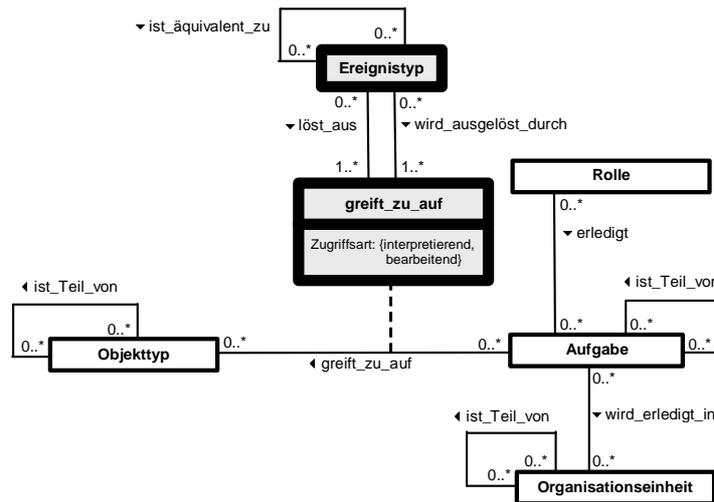


Abbildung 7.1: Fachliche Ebene des 3LGM²_A; hervorgehoben sind die überarbeiteten Klassen.

in den unterstützenden **Anwendungsbausteinen** implementiert sein, oder diese müssen die Funktionalität über Kommunikation mit einem anderen **Anwendungsbaustein** erwerben. Letzteres gilt sehr oft für die Funktionalität zur Unterstützung bestimmter Teil-**Aufgaben**, die zu mehreren übergeordneten **Aufgaben** gehören und deutet auf die Nutzung spezialisierter Komponenten und die Forderung nach funktionaler Integration hin (siehe Abschnitte 9.3.2).

2. Wird ein **Ereignistyp** durch bearbeitenden Zugriff auf **Objekttypen** ausgelöst, hat also eine bearbeitende Aktivität einer **Aufgabe** stattgefunden, dann muss die damit verbundene Änderung von Daten ggf. an andere **Anwendungsbausteine**, welche interpretierende¹ **Aufgaben** unterstützen, weitergegeben werden. Das gilt sehr oft für **Objekttypen**, die bei der Erfüllung vieler unterschiedlicher **Aufgaben** interpretiert werden und deutet auf die Forderung nach Datenintegration hin (siehe Abschnitt 9.3.1).

Die Überarbeitung des 3LGM² hinsichtlich der **Ereignistypen** enthält auch die Assoziationsbeziehung **ist_äquivalent_zu** für **Ereignistypen**. Damit können beispielsweise **Ereignistypen** aus verschiedenen **Kommunikationsstandards** mit gleicher Bedeutung verknüpft werden. Tabelle 7.1 zeigt Beispiele für äquivalente **Ereignistypen** der **Kommunikationsstandards** HL7 und SAP-HCM. Die Tabelle lässt auch erkennen, warum eine transitive Auswertung der hier definierten Äquivalenzbeziehung nicht sinnvoll ist: Beispielsweise werden die im **Kommunikationsstandard** SAP-HCM definierten **Ereignistypen** NP11I0 und NP11P0 im **Kommunikationsstandard** HL7 über den **Ereignistyp** A01 ausgedrückt. NP11I0 und NP11P0 sind jedoch nicht äquivalent. In HL7 wird die entsprechende Unterscheidung durch zusätzliche Angaben innerhalb der übermittelten Nachrichten erreicht. Sofern eine Unterscheidung auch für HL7-Nachrichten erforderlich ist, kann sie mit dem 3LGM² nur über Modellierung „eigener“ auf A01 basierender **Ereignistypen** erreicht werden.

¹ Hier wird das Attribut **Zugriffsart** der **greift_zu_auf**-Beziehung zwischen den Klassen **Aufgabe** und **Objekttyp** verwendet.

HL7-Ereignistyp		HCM-Ereignistyp	
A01	(admit / visit notification)	NP11I0 NP11P0	(Aufnahme anlegen (Ist)), (Aufnahme anlegen (Plan))
A02	(transfer a patient)	NV11I0 NV11P0	(Verlegung anlegen (Ist)), (Verlegung anlegen (Plan))
A03	(discharge/end visit)	NP97I0 NP97P0	(Entlassung anlegen (Ist)), (Entlassung anlegen (Plan))
A08	(update patient information)	NP12I0 NV12I0 NP98I0	(Aufnahme ändern), (Verlegung ändern), (Entlassung ändern)
A11	(cancel admit / visit notification)	NP12IS	(Aufnahme stornieren)
A12	(cancel transfer)	NV12IS	(Verlegung stornieren)
A13	(cancel discharge / end visit)	NP98IS	(Entlassung stornieren)

Tabelle 7.1: Beispiele für äquivalente Ereignistypen in den Kommunikationsstandards HL7 und SAP-HCM (vgl. Tabelle 6.1; Quellen: [HL7 1999] und [SAP 2005])

7.2.2 Schnittstellen und Operationen

Die Klasse **Bausteinschnittstelle** im 3LGM² entspricht grundsätzlich den in den Abschnitten 4.2.2 und 4.2.3 zitierten Definitionen für *Schnittstelle (interface)*. Die unmittelbaren Assoziationsbeziehungen zu den Klassen **Ereignistyp-Nachrichtentyp** und **Ereignistyp-Dokumententyp** (vgl. Abbildung 6.2a) und die weiterführenden Assoziationsbeziehungen zu **Nachrichtentyp**, **Dokumententyp** und **Ereignistyp** betonen den Ansatz der nachrichtenbasierten Kommunikation zwischen Komponenten und die dafür entwickelten Technologien, z. B. Kommunikationsserver (vgl. Abschnitt 4.4.3). Damit ist beispielsweise eine Architekturmodellierung, die die Anforderung und Nutzung von Diensten in den Vordergrund stellt (vgl. Abschnitte 4.2.1, 4.2.3 und 4.2.5), nicht bzw. nur eingeschränkt möglich. Gleiches gilt auch für Architekturen, die den in Abschnitt 4.4.3 vorgestellten EAI-Architekturmustern entsprechen, mit Ausnahme des Musters *Integrationsbote*.

Operationen als flexible Elemente zur Kommunikationsmodellierung

Mit den bisherigen Klassen und Assoziationsbeziehungen des 3LGM² ist es also nicht bzw. nur schwer möglich, eine über den einfachen Nachrichten- bzw. Dokumentenaustausch hinausgehende Interaktion zu modellieren. Insbesondere kann der Zusammenhang zwischen den möglicherweise bei einer Interaktion initial übergebenen Daten und den als Antwort zurückgegebenen Daten nicht modelliert werden. Diese Tatsache führt zur Forderung der Berücksichtigung des Begriffes *Operation*.

In dieser Arbeit wird angenommen, dass jede Art der Interaktion von Komponenten über Operationen im Sinne des im OMG Object Model definierten Operationsbegriffes beschrieben werden kann (vgl. die zitierte Definition und Erklärung für *Operation* in Abschnitt 4.2.3). Die Annahme wird im Zusammenhang mit der Annahme getroffen, dass die Anforderungen bzgl. detaillierter Kommunikationsmodellierung aus Sicht des Managements von Informationssystemen geringer sind als aus Sicht der Softwareentwicklung.

In vielen Fällen wird die Interaktion von Komponenten über eine Sequenz von aufeinanderfolgenden Operationsaufrufen modelliert werden müssen².

² Vgl. dazu die Definition für *Aktivität* als Graph von Aktionen im RM-ODP ([ISO/IEC JTC 1 1996b], S. 4, Definition 8.5).

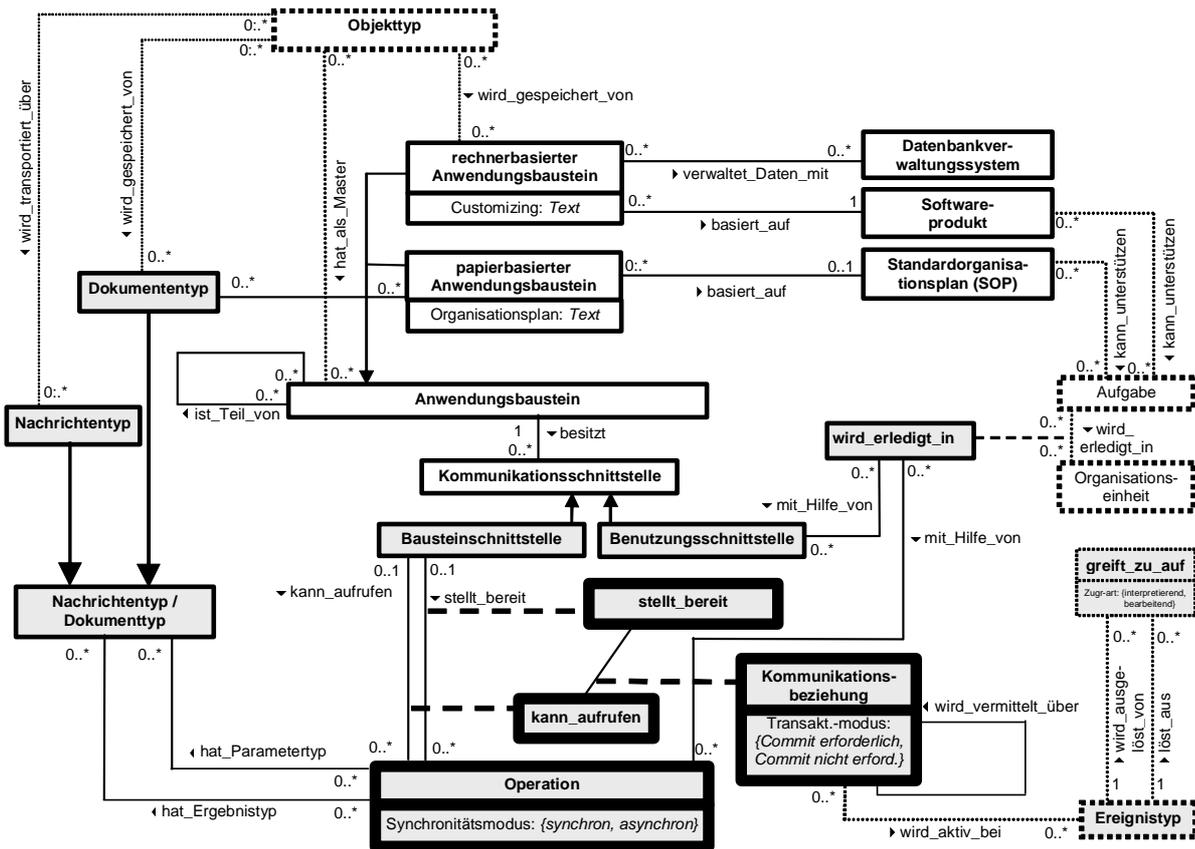


Abbildung 7.2: Logische Werkzeugebene des 3LGM²_A; hervorgehoben sind die neu eingeführte Klasse **Operation** und die überarbeiteten Assoziationsbeziehungen.

Operationen im 3LGM²

Auf der Basis der Ausführungen zum Begriff *Operation* wird hier die Klasse **Operation** auf der logischen Werkzeugebene in das 3LGM²_A eingeführt. Mit der Einführung werden auch die Assoziationsbeziehungen der Klassen **Bausteinschnittstelle** und **Ereignis-Nachrichtentyp/Ereignis-Dokumententyp** überarbeitet, insbesondere die Beziehung **Kommunikationsbeziehung**. Die Modellierung von Kommunikation ist infolge dessen nur noch über das Modellieren von **Operationen** und **Operationsaufrufen** möglich (Abbildung 7.2).

Den von einem Anwendungsbaustein zur Verfügung gestellten **Operationen** können Parameter- und Ergebnistypen zugeordnet werden. Diese können mit den bereits im 3LGM² definierten Klassen **Nachrichtentyp** und **Dokumententyp** modelliert werden.

Eine **Kommunikationsbeziehung** zwischen zwei **Bausteinschnittstellen** ist nur dann möglich, wenn eine der beiden eine bestimmte **Operation** bereitstellt, die von der anderen aufgerufen werden kann. Aus diesem Grund ist die **Kommunikationsbeziehung** im überarbeiteten 3LGM² eine Assoziationsbeziehung zwischen den beiden Assoziationsklassen **stellt_bereit** und **kann_aufrufen**.

Für die Verwendung in den Kapiteln zur Architekturbewertung werden hier, analog zu den Mengendefinitionen in Kapitel 6, semiformale Mengendefinitionen angegeben:

Kasten 7.1: Das Weglassen der Konfigurationen im 3LGM_A²

Im 3LGM² sind für die Verknüpfung von **Aufgaben** mit **Anwendungsbausteinen** und für die Verknüpfung von **Anwendungsbausteinen** mit **physischen Datenverarbeitungsbausteinen** Konfigurationsklassen vorgesehen. Diese wurden im 3LGM_A² weggelassen, um Modellierer zu mehr Ausführlichkeit und Präzision bei der 3LGM_A²-basierten Modellierung zu „zwingen“.

Wird beispielsweise eine **Aufgabe** mit einer **Anwendungsbausteinkonfiguration** mit mehr als einem **Anwendungsbaustein** verknüpft, dann wird dabei nicht ausgedrückt, wofür die einzelnen **Anwendungsbausteine** tatsächlich bei der Erledigung der betreffenden **Aufgabe** verwendet werden. Damit werden Analysen wie die in Kapitel 9 beschriebenen Überprüfungen der Erfüllung von Integrationsanforderungen erschwert. Es kann u. a. nicht unterschieden werden, ob die **Anwendungsbausteine** jeweils alle der von der betreffenden **Aufgabe** zugegriffenen **Objekttypen** benötigen oder nur einzelne.

OP := { x | x ist eine **Operation**.} (7.1)

SB := { x | x ist eine **stellt_bereit**-Beziehung.} (7.2)

KA := { x | x ist eine **kann_aufrufen**-Beziehung.} (7.3)

Die Bedeutung des Mengenkürzels KOMB für Kommunikationsbeziehungen ändert sich im Vergleich zu der aus Abschnitt 6.3.2, da die „neuen“ Kommunikationsbeziehungen in einem 3LGM_A²-basierten Modell Beziehungen zwischen **stellt_bereit**-Beziehungen und **kann_aufrufen**-Beziehungen sind.

Attribute für die qualitative Beschreibung der Kommunikation

Die eingeführte Klasse **Operation** hat das Attribut *Synchronitätsmodus* mit den möglichen Werten *synchron* und *asynchron*. Es wird in Abschnitt 10.3 zur Bewertung von Abhängigkeiten von **Anwendungsbausteinen** verwendet. Aufrufe von synchronen **Operationen** blockieren den aufrufenden **Anwendungsbaustein** solange, bis die **Operation** ausgeführt und das evtl. zugehörige Ergebnis tatsächlich zurückgegeben wurde. Aufrufe von asynchronen **Operationen** blockieren den aufrufenden **Anwendungsbaustein** nicht. Diese einfache Modellierung der (A)Synchronität über ein Attribut wird angewendet, da das 3LGM² nicht für den ausführlichen Softwareentwurf entwickelt wurde. Bei der Implementierung asynchroner Kommunikation mit ErgebnISRückgaben sind i. d. R. mindestens zwei Aufrufe von Funktionen bzw. Prozeduren notwendig: ein Aufruf für die Übermittlung der Anfrage und ein Aufruf für die Rückübermittlung des Ergebnisses³.

Bei der überarbeiteten Assoziationsklasse **Kommunikationsbeziehung** wurde das Attribut *Transaktionsmodus* mit den möglichen Werten *Commit erforderlich* und *Commit nicht erforderlich* ergänzt. Wie das Attribut *Synchronitätsmodus* der Klasse **Operation** wird es in Abschnitt 10.3 zur Bewertung der Abhängigkeit von **Anwendungsbausteinen** verwendet. Ein **Commit** wird hier in Anlehnung an den **Commit**-Begriff in der Datenbanktheorie (siehe z. B. [VOSSEN 2000], S. 582-596 oder [RAHM 1994], S. 111-131) vereinfacht als das Bestätigen des Erfolges eines Operationsaufrufes aufgefasst: Bei einer **Kommunikationsbeziehung**, die ein **Commit** erfordert, wird die Transaktion im aufrufenden **Anwendungsbaustein**, die zum Operationsaufruf führte, zurückgenommen, wenn das **Commit** nicht geliefert wird.

³ Die Anfrage muss nicht notwendigerweise eine „echte“ Anfrage nach bestimmten anwendungsbezogenen Daten, z. B. Patientendaten, sein. Sie kann beispielsweise auch eine Aufforderung zur Speicherung bzw. Weiterverarbeitung bestimmter Daten sein, wobei der Erfolg der Speicherung bzw. Weiterverarbeitung über eine Statusnachricht zurückübermittelt wird.

Operationen und die fachliche Ebene des 3LGM_A²

Die Klasse `Operation` hat zwei unmittelbare Assoziationsbeziehungen zu Klassen der fachlichen Ebene.

Die Beziehung `wird_aktiv_bei` zur Klasse `Ereignistyp` repräsentiert den Zusammenhang zwischen der Modellierung von `Informationsprozessen` auf der fachlichen Ebene und von `Kommunikationsprozessen` auf der logischen Werkzeugebene (vgl. Abschnitt 6.4). Abhängig vom Eintreten bestimmter fachlicher Ereignisse müssen im betrachteten Informationssystem u. U. Kommunikationsbeziehungen aktiv werden. Dadurch werden dann den `Anwendungsbausteinen` die die betreffenden `Aufgaben` unterstützen, relevante Daten übermittelt oder benötigte Funktionalität bereitgestellt.

Mit der Beziehung `mit_Hilfe_von` zwischen der Assoziationsklasse `wird_erldigt_in` und der Klasse `Operation` kann modelliert werden, welche Funktionalität durch `Operationen` bereitgestellt wird. Bei sehr detaillierter Betrachtung der Funktionalitätsbereitstellung ist eine entsprechend detaillierte Modellierung von `Aufgaben` erforderlich. Wenn z. B. die Bereitstellung von Funktionalität zur Suche bestimmter Informationen betrachtet werden soll, muss eine zugehörige `Aufgabe` modelliert worden sein.

Über `Operationen` und `Kommunikationsbeziehungen` wird die Bereitstellung von Funktionalität von `Anwendungsbausteinen` für andere `Anwendungsbausteine` und die Nutzung der Funktionalität modelliert. Die Bereitstellung von Funktionalität von `Anwendungsbausteinen` für ihre Benutzer wird durch die Assoziationsbeziehung `mit_Hilfe_von` zwischen der Assoziationsklasse `wird_erldigt_in` und der Klasse `Benutzungsschnittstelle` modelliert. Auf eine detailliertere Modellierung der `Benutzungsschnittstellen` wie bei den `Bausteinschnittstellen` wird hier verzichtet.

7.3 Transformation von Nachrichtentypen

In vielen Informationssystemen werden bei der Datenübermittlung Nachrichten bestimmter `Nachrichtentypen` in Nachrichten anderer `Nachrichtentypen` transformiert⁴. Sehr oft erfolgt dabei auch eine Umbenennung des der betreffenden Nachricht zugeordneten `Ereignistyps`. Das geschieht vor allem bei Anwendung unterschiedlicher `Kommunikationsstandards`, da unterschiedliche `Kommunikationsstandards` gleiche oder ähnliche `Nachrichtentypen` oft unterschiedlich benennen.

Das Transformieren von `Nachrichtentypen` kann über `Kommunikationsbeziehungen` modelliert werden, bei denen die aufrufende `Bausteinschnittstelle` gleichzeitig die bereitstellende `Bausteinschnittstelle` ist. Dabei ist der Quell-`Nachrichtentyp` der aufgerufenen `Operation` als Parametertyp zugeordnet, der Ziel-`Nachrichtentyp` folglich als `Ergebnistyp`.

7.4 Dienste

7.4.1 Spezifikationen für Dienste

Informationale und funktionale Spezifikationen für Informationssystemkomponenten legen oft nicht ausdrücklich fest, ob sie durch eine einzige Komponente oder eine Kombination mehrerer

⁴ Im Folgenden wird kurz „Transformation von Nachrichtentypen“ o. ä. geschrieben.

Komponenten zu erfüllen sind. Daher wird in diesem Zusammenhang von *Diensten* (bzw. Spezifikationen von Diensten), nicht von Komponenten, gesprochen.

Definition 7.1 Ein *Dienst* stellt bestimmte Funktionalitäten zur Informationsverarbeitung zur Verfügung. Er kann durch einen oder mehrere Komponenten realisiert sein.

Ende — Definition

Dienstspezifikationen können nach zwei Dienstkategorien unterschieden werden: Anwendungsdienste und Vermittlungsdienste. Spezifikationen für *Anwendungsdienste* beinhalten u. a.

- Informationen, die in einem oder mehreren Anwendungsbereichen verarbeitet werden, z. B. Fallinformationen oder Befundinformationen,
- Funktionalitäten zur Verarbeitung der Informationen, z. B. Operationen⁵ zur Befundpräsentation,
- Strukturen zum Austausch der Informationen, z. B. Nachrichtentypen für die Befundübermittlung, und
- Ereignistypen, die das Aktivieren/Abfragen von Funktionalitäten auslösen, z. B. einen Ereignistyp der das Vorliegen von Befunden ausdrückt.

Diese vier Teilspezifikationen können unterschiedlich stark ausgeprägt sein. Spezifikationen für Anwendungsdienste sind u. a. im HL7-Kommunikationsstandard⁶, in anwendungsbereichspezifischen Dienstspezifikationen zur OMA und in der HISA enthalten. Beispiele für Anwendungsdienste sind der Person Identification Service (PIDS) aus den OMA-Diensten für das Gesundheitswesen oder der Subject of Care Healthcare Common Services (S-HCS) der HISA.

Spezifikationen für *Vermittlungsdienste* beinhalten u. a.

- Informationen, die zur Vermittlung von Diensten benötigt werden, z. B. Adressen von Dienstimplementierungen oder Verschlüsselungsmethoden,
- Funktionalitäten für die Vermittlung von Kommunikationsverbindungen zum Aufruf von Diensten, z. B. Operationen zur Lokalisierung von Dienstimplementierungen,
- Strukturen zum Austausch vermittlungsspezifischer Informationen, z. B. Nachrichtentypen für das Registrieren von Dienstimplementierungen in einem dafür vorgesehenen Verzeichnis oder das Abfragen der Lokalisation von Dienstimplementierungen aus diesem Verzeichnis, und
- Ereignistypen, die das Aktivieren/Abfragen einer vermittlungsspezifischen Funktionalität auslösen, z. B. einen Ereignistyp, der das Bereitstehen einer neuen Dienstimplementierung ausdrückt.

Zu diesen Spezifikationen gehören u. a. der CORBA-Standard und das DCOM.

⁵ In der Beschreibung der Dienstspezifikationen sind unächst keine Begriffe aus dem 3LGM² gemeint. Deshalb erfolgt keine Anwendung der Schriftart für 3LGM²-Klassen.

⁶ Im HL7-Kommunikationsstandard wird nicht ausdrücklich von Diensten gesprochen, jedoch enthält er Spezifikationen für Anwendungsdienste in der in dieser Arbeit unterstellten Bedeutung.

7.4.2 Modellierung von Diensten mit dem 3LGM²

Anwendungsdienste

Zur Beschreibung von Anwendungsdiensten (vgl. 7.4.1) stellt das überarbeitete 3LGM² folgende Klassen zur Verfügung:

- **Objektyp** auf der fachlichen Ebene zur Beschreibung der anwendungsspezifischen Informationen,
- **Aufgabe** auf der fachlichen Ebene als inhaltliche Vorgabe für die bereitzustellende Funktionalität,
- **Nachrichtentyp** auf der logischen Werkzeugebene zur Beschreibung von Strukturen zum Informationsaustausch über **Bausteinschnittstellen** beim Eintreten von bestimmten Ereignistypen und
- **Anwendungsbaustein**, **Operation** und **Nachrichtentyp** auf der logischen Werkzeugebene zur Beschreibung der Bereitstellung von implementierter Funktionalität durch Komponenten für andere Komponenten.

Vermittlungsdienste

Zur Beschreibung von Vermittlungsdiensten (vgl. 7.4.1) stellt das 3LGM² auf der logischen Werkzeugebene dieselben Klassen wie für die Modellierung von Anwendungsdiensten zur Verfügung. Zusätzlich wird

- die Assoziationsbeziehung **vermittelt** zur Beschreibung der Tatsache, dass eine Kommunikationsbeziehung durch eine oder mehrere andere Kommunikationsbeziehungen vermittelt wird,

zur Verfügung gestellt.

7.5 Kommunikationsverbindungen

Bei der Übertragung von Informationen werden Daten oft von einem Start-Anwendungsbaustein über mehrere Anwendungsbausteine hinweg zu einem Ziel-Anwendungsbaustein übermittelt.

Für die Modellierung der direkten Kommunikation zweier Anwendungsbausteine sieht das 3LGM² bereits die Assoziationsklasse **Kommunikationsbeziehung** vor. Über den Begriff *Kommunikationsverbindung* wird die Kommunikation zwischen Anwendungsbausteinen auf eine Folge von **Kommunikationsbeziehungen** verallgemeinert. Dabei wird die Assoziationsbeziehung **wird_vermittelt_über** im 3LGM² berücksichtigt (Abbildung 7.2).

Definition 7.2 Eine *Kommunikationsverbindung* ist eine Folge von **Kommunikationsbeziehungen**. Dabei gilt für jede **Kommunikationsbeziehung**, mit Ausnahme der letzten,

- dass sie die nach ihr stehende **Kommunikationsbeziehung** vermittelt,
- dass sie gemeinsam mit einer oder mehreren nach ihr stehenden **Kommunikationsbeziehungen** die danach stehende **Kommunikationsbeziehung** vermittelt oder
- dass ihre anbietende **Bausteinschnittstelle** aufrufende **Bausteinschnittstelle** einer folgenden **Kommunikationsbeziehung** ist.

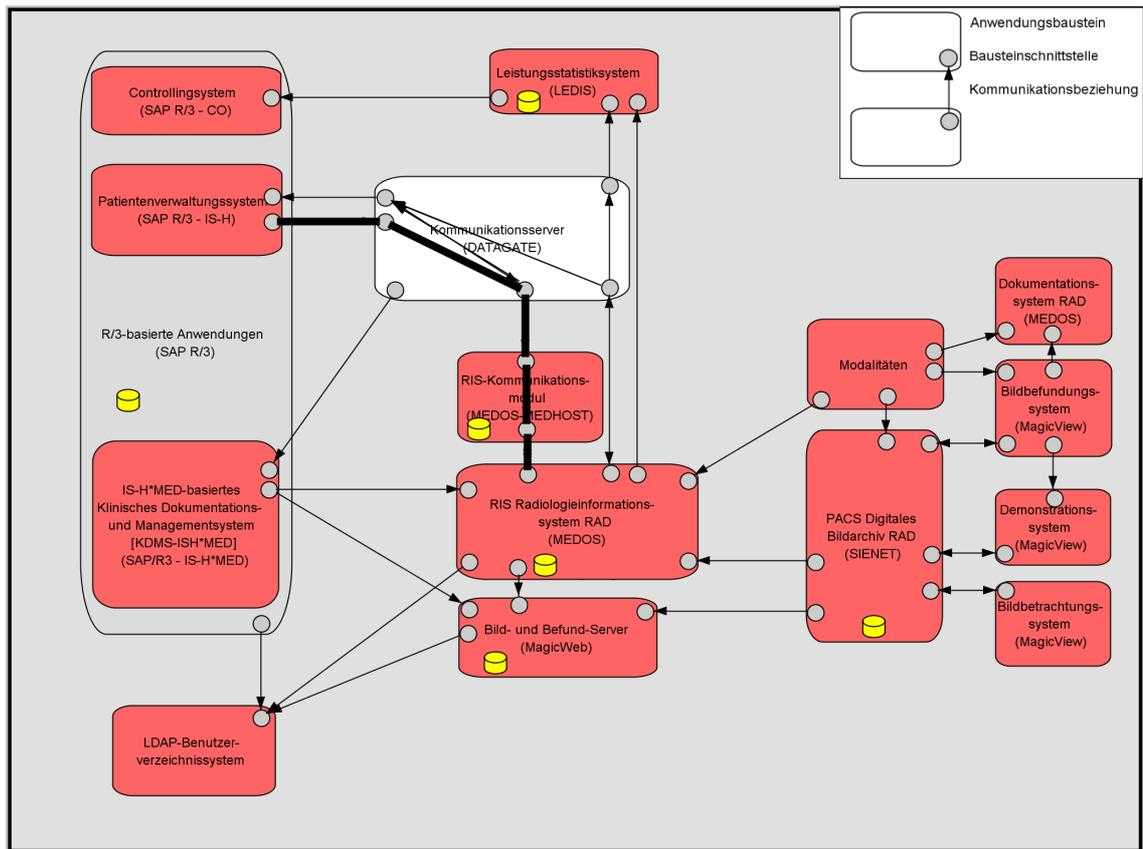


Abbildung 7.3: Eine Kommunikationsverbindung zur Übertragung von Falldaten im Informationssystem des UKL

Ende — Definition

Die Abbildungen 7.3 und 7.6 zeigen Beispiele für Kommunikationsverbindungen ohne vermittelnde Kommunikationsbeziehungen. Abbildung 7.5 zeigt ein Beispiel für Kommunikationsverbindungen mit vermittelnden Kommunikationsbeziehungen.

7.6 Beispiele für die Modellierung verschiedener Architekturstile

Die in diesem Abschnitt beschriebenen Beispiele sind auf die Architekturmodellierung mit Hilfe der Klassen der logischen Werkzeugebene bezogen. Die Anwendung der in Abschnitt 7.2.1 beschriebenen Überarbeitung der fachlichen Ebene erfolgt in Kapitel 9.

7.6.1 Modellierung von HISA-basierten Architekturen

Informationssystemkomponenten, die den Vorgaben des HISA-Standards *EN 12967-1* (vgl. Abschnitt 4.2.4) entsprechen, können auf der Basis des 3LGM² folgendermaßen modelliert werden:

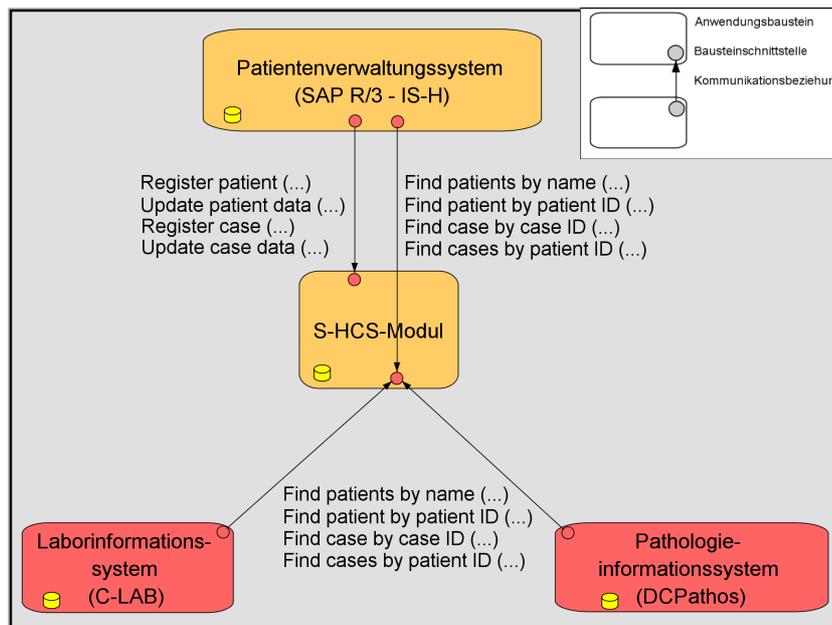


Abbildung 7.4: Eine einfache OMA-basierte Architektur

1. Informationsklassen werden auf **Objekttypen** abgebildet. Den **Objekttypen** werden **Nachrichtentypen** als Parametertypen und Ergebnistypen für die Dienstoperationen zugeordnet.
2. Informationssystemkomponenten, die die Funktionalitätsbeschreibung implementieren, werden auf **Anwendungsbausteine** abgebildet.
3. Elemente der Funktionalitätsbeschreibungen werden auf **Operationen** abgebildet.

Beispiel

Ein einfaches Modell einer Implementierung von Diensten der Gruppe „Subjects of Care Healthcare Common Services (S-HCS)“ könnte, abhängig von der Implementierung, u. a. folgende Elemente enthalten (Abbildung 7.4):

- die **Objekttypen** *Patient* und *Fall* (vgl. [CEN TC251 1997], S. 15-18),
- einen **Anwendungsbaustein** *S-HCS-Modul* mit einer **Bausteinschnittstelle**, der die **Operationen** *Find patient by name*, *Find patient by patient ID*, *Find case by case ID* und *Find case by patient ID* zugeordnet sind, und einer weiteren **Bausteinschnittstelle**, der die **Operationen** *Register patient*, *Update patient data*, *Register case* und *Update case data* zugeordnet sind (vgl. [CEN TC251 1997], S. 39),
- die **Nachrichtentypen** *Patient query message*, *Patient query result message* und *Patient admittance message*, die den **Operationen** als Parameter bzw. Ergebnistypen zugeordnet sind,
- die **Anwendungsbausteine** *Patientenverwaltungssystem*, *Laborinformationssystem* und *Pathologieinformationssystem*, die die S-HCS nutzen,

- je eine Bausteinschnittstelle für die letztgenannten **Anwendungsbausteine**, über die sie die **Operationen** zur Abfrage von Patienten- und Falldaten der S-HCS aufrufen, und
- ein Bausteinschnittstelle für das *Patientenverwaltungssystem* **Anwendungsbausteine**, über die es die **Operationen** zur Registrierung (Speicherung) von Patienten- und Falldaten der S-HCS aufruft.

Eine vollständiges Modell einer Implementierung der Dienstgruppe müsste um weitere Elemente ergänzt werden.

7.6.2 Modellierung von OMA-basierten Architekturen

Komponenten eines OMA-basierten Informationssystems können auf der Basis des 3LGM² folgendermaßen modelliert werden:

1. Ein ORB wird auf einen **Anwendungsbaustein** abgebildet.
2. Die vom ORB und den anderen Komponenten bereitgestellten Operationen werden auf 3LGM²-**Operationen** abgebildet.
3. Definitionen für die Parametertypen und die Ergebnistypen der Operationen werden auf **Nachrichtentypen** abgebildet.

Da sowohl die OMA als auch die CORBA, die Spezifikation für die zentrale Komponente der OMA, keine anwendungsbezogenen Informations- oder Funktionalitätsbeschreibungen enthalten, können zunächst keine Bezüge zur fachlichen Ebene des 3LGM² hergestellt werden. Wenn spezielle Dienstspezifikationen der OMG, z. B. die Spezifikation des Person Identification Service, angewendet werden, können die enthaltenen Datentypen in einem 3LGM²-Modell als **Nachrichtentypen** mit **Objekttypen** in Beziehung gesetzt werden. Die speziellen **Operationen** können mit **Aufgaben** in Beziehung gesetzt werden.

Beispiel

Ein einfaches Modell einer Implementierung von Diensten zur Patientenverwaltung in einer OMA-basierten Architektur könnte u. a. folgende Elemente enthalten (Abbildung 7.5):

- die **Objekttypen** *Patient* und *Fall*,
- einen **Anwendungsbaustein** *PIDS-Modul* (PIDS=Patient Identification Service) mit einer **Bausteinschnittstelle**, der die **Operationen** *find_or_register_ids* und *find_candidates* zugeordnet sind (vgl. [OMG 2001c], S. 2-28 - 2-32 und 2-19 - 2-21),
- die **Nachrichtentypen** entsprechend der PIDS-Spezifikation, die den **Operationen** des PIDS als Parameter bzw. Ergebnistypen zugeordnet sind,
- einen **Anwendungsbaustein** *TOS-Modul* (TOS=Trading Object Service) mit einer **Bausteinschnittstelle**, der die **Operationen** *export* und *query* zugeordnet sind (vgl. [OMG 2000f], S. 2-9 - 2-17 und 2-3 - 2-8),
- die **Nachrichtentypen** entsprechend der TOS-Spezifikation, die den **Operationen** des TOS als Parameter bzw. Ergebnistypen zugeordnet sind,
- einen **Anwendungsbaustein** *ORB* mit **Bausteinschnittstellen**, die den IDL Stubs und den Skeletons für die beteiligten Komponenten entsprechen,

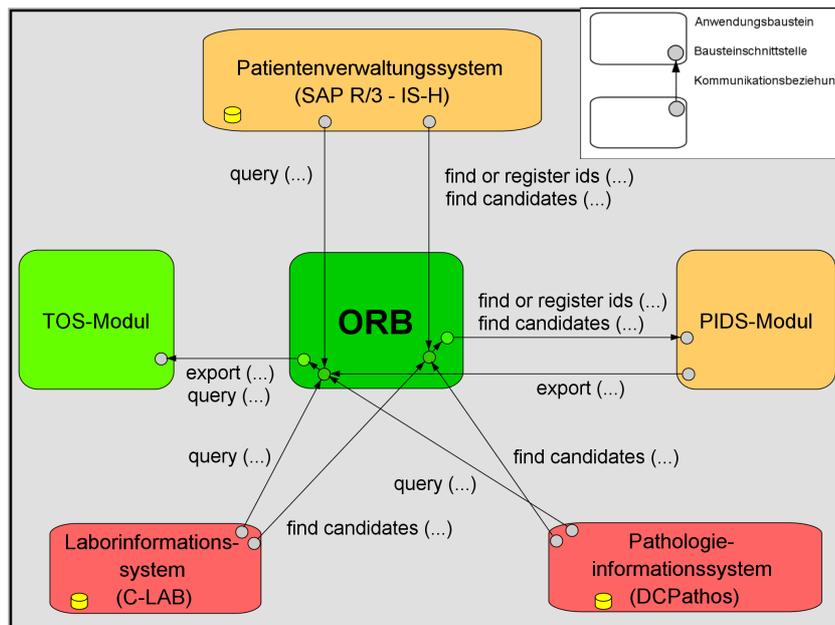


Abbildung 7.5: Eine einfache OMA-basierte Architektur

- die Anwendungsbausteine *Patientenverwaltungssystem*, *Laborinformationssystem* und *Pathologieinformationssystem*, die den TOS und den PIDS nutzen,
- je eine Bausteinschnittstelle für die letztgenannten Anwendungsbausteine, über die sie die Operationen des TOS aufrufen, und
- je eine Bausteinschnittstelle für die letztgenannten Anwendungsbausteine, über die sie die Operationen des PIDS aufrufen.

7.6.3 Modellierung von Kommunikationsserver-basierten Architekturen

Komponenten eines Kommunikationsserver-basierten Informationssystems können auf der Basis des 3LGM² folgendermaßen modelliert werden:

1. Ein Kommunikationsserver wird auf einen **Anwendungsbaustein** abgebildet.
2. Schnittstellenmodule werden auf **Bausteinschnittstellen** abgebildet.
3. Nachrichtenstrukturdefinitionen werden auf **Nachrichtentypen** abgebildet.
4. Routendefinitionen für die Weitergabe von Nachrichten zwischen den **Bausteinschnittstellen** des Kommunikationsservers werden auf **Kommunikationsbeziehungen** von Bausteinschnittstellen innerhalb des entsprechenden Anwendungsbausteines abgebildet.

Da das Kommunikationsserverkonzept nicht auf bestimmte Unternehmensaufgaben oder Informationen beschränkt ist, können keine Bezüge zur fachlichen Ebene hergestellt werden⁷.

⁷ Mit einigen Kommunikationsserverprodukten werden anwendungsspezifische Nachrichtenstrukturdefinitionen, z. B. für HL7-Nachrichten, zur Verwendung in Filter- und Übersetzungsdefinitionen mitgeliefert. Trotzdem

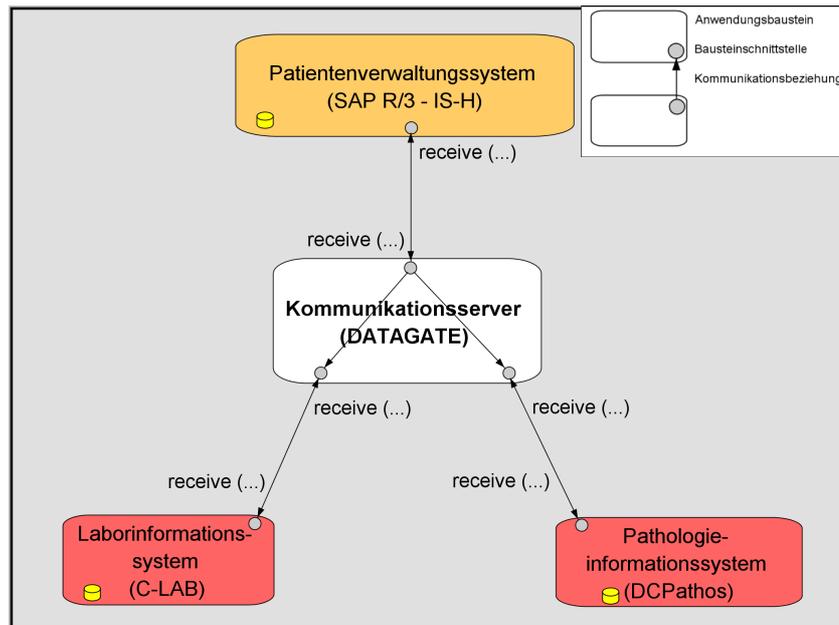


Abbildung 7.6: Eine einfache Kommunikationsserverarchitektur

Beispiel

Ein einfaches Modell einer Implementierung von Komponenten zur Patientenverwaltung in einer Kommunikationsserver-basierten Architektur könnte u. a. folgende Elemente enthalten (Abbildung 7.6; vgl. [HL7 1999], Kapitel 3):

- die Objekttypen *Patient* und *Fall*,
- einen Anwendungsbaustein *Kommunikationsserver*,
- eine Bausteinschnittstelle des Kommunikationsservers, die die Operation *receive* mit dem Nachrichtentyp *ADT* als Parametertyp bereitstellt und die die gleiche Operation mit dem Nachrichtentyp *ACK* als Parameter aufrufen kann,
- zwei Bausteinschnittstellen des Kommunikationsservers, die jeweils die Operation *receive* mit dem Nachrichtentyp *ADT* als Parametertyp aufrufen können und die die gleiche Operation mit dem Nachrichtentyp *ACK* als Parameter bereitstellen,
- einen Anwendungsbaustein *Patientenverwaltungssystem* mit einer Bausteinschnittstelle, die die Operation *Receive* mit dem Nachrichtentyp *ADT* als Parametertyp aufrufen kann und die die gleiche Operation mit dem Nachrichtentyp *ACK* als Parameter bereitstellt, und
- die Anwendungsbausteine *Laborinformationssystem* und *Pathologieinformationssystem* mit je einer Bausteinschnittstelle, die die Operation *Receive* mit dem Nachrichtentyp *ADT* als Parametertyp bereitstellt und die die gleiche Operation mit dem Nachrichtentyp *ACK* als Parameter aufrufen kann.

sind auch diese Produkte nicht auf einen bestimmten Anwendungsbereich beschränkt.

7.7 Einführung der Klasse Begriffssystem

Unabhängig von der Modellierung verschiedener Architekturstile wird in diesem Abschnitt zur Modellierung der semantischen Integration (vgl. Abschnitt 2.2.4) ergänzend eine weitere Klasse auf der fachlichen Ebene des $3LGM_A^2$ eingeführt: die Klasse **Begriffssystem** (Abbildung 7.7).

$$\underline{\text{BGS}} := \{bgs \mid bgs \text{ ist ein } \text{Begriffssystem.}\} \quad (7.4)$$

Wenn eine **Aufgabe** auf einen **Objektyp** zugreift, dann kann über die in das $3LGM_A^2$ eingeführte Assoziationsbeziehung **unter_Berücksichtigung_von** zusätzlich ausgedrückt werden, dass dabei ein bestimmtes **Begriffssystem** berücksichtigt wird. Ein Beispiel ist die Berücksichtigung des ICD10-Diagnosenkataloges bei der Diagnosedokumentation (Abbildung 7.8).

Der Nutzen der Modellierung von **Begriffssystemen** liegt u. a. darin, dass

- dokumentiert wird, welche **Begriffssysteme** verwendet werden,
- abgeleitet werden kann, in welchen **Anwendungsbausteinen** welche **Begriffssysteme** benötigt werden, und
- abgeleitet werden kann, in welchen **Anwendungsbausteinen** welche **Begriffssysteme** gespeichert sind und wie sie ggf. zwischen **Anwendungsbausteinen** ausgetauscht werden.

Alle dieser Argumente charakterisieren die Thematik der semantischen Integration aus der Sicht des Managements von Informationssystemen. Die Bestimmung von entsprechenden Integrationsanforderungen und ihre Überprüfung wird in den Abschnitten 9.3.3 und 9.5.3 behandelt.

Auch **Begriffssysteme** werden i. d. R. geändert, z. B. mit einer neuen Version aktualisiert. Für die Modellierung der Tatsache, dass ein **Begriffssystem** durch eine bestimmte **Aufgabe**

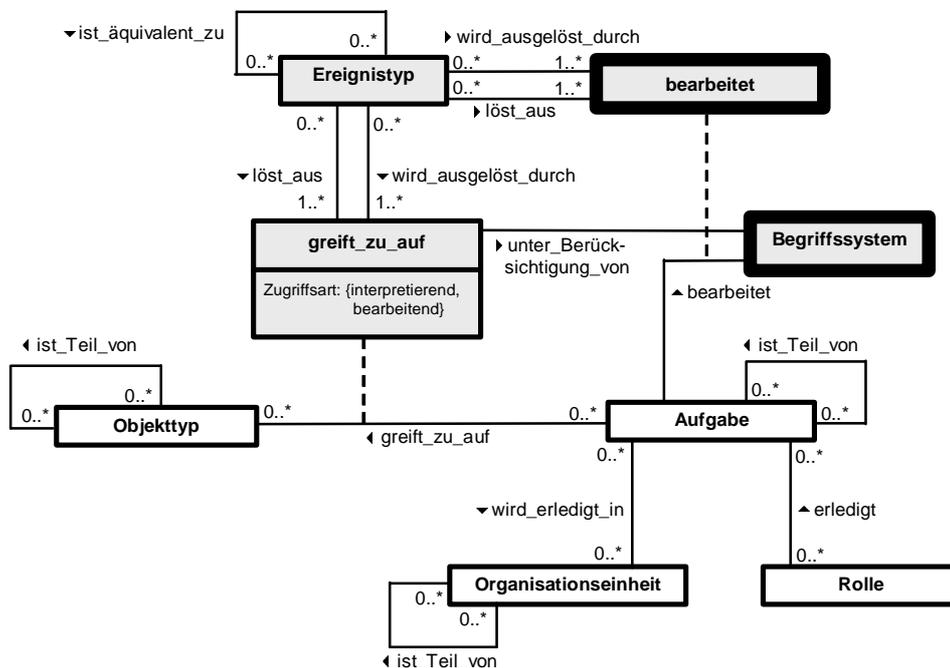


Abbildung 7.7: Erweiterte fachliche Ebene des $3LGM_A^2$; hervorgehoben sind die neu eingeführte Klasse **Begriffssystem** und die zuvor in Abschnitt 7.2.1 überarbeiteten Klassen.

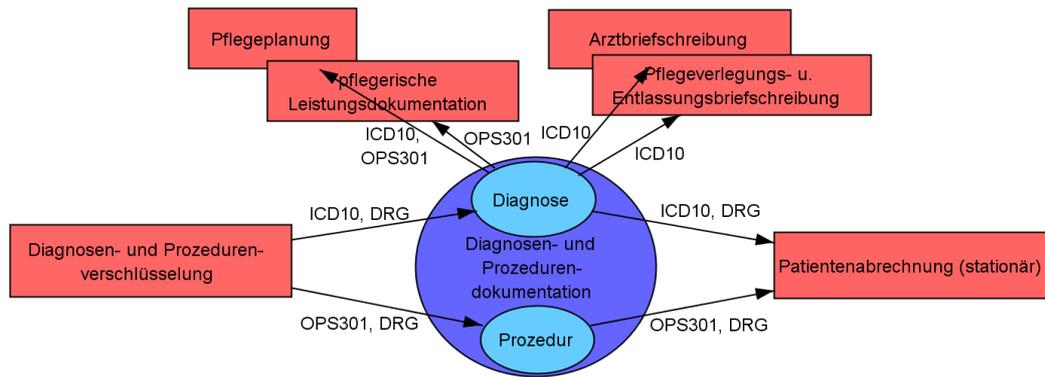


Abbildung 7.8: Beispiele für die Zuordnung von Begriffssystemen zu `greift_zu_auf`-beziehungen: Bei der Interpretation und Bearbeitung von Diagnosen und Prozeduren werden die Diagnosenklassifikation *ICD10*, die Prozedurenklassifikation *OPS301* und das Abrechnungssystem *DRG* angewendet.

bearbeitet wird, z. B. durch Bereitstellung von Aktualisierungen, wird die zusätzliche Assoziationsbeziehung `bearbeitet` zwischen den Klassen `Aufgabe` und `Begriffssystem` eingeführt. Der bearbeitende Zugriff auf ein `Begriffssystem` kann, ähnlich dem Zugriff auf `Objekttypen`, durch `Ereignistypen` ausgelöst werden oder `Ereignistypen` auslösen. Deshalb wird die Assoziationsbeziehung `bearbeitet` mit der Klasse `Ereignistyp` assoziiert.

Die Klasse `Begriffssystem` ist wie die Klasse `Objekttyp` mit `Anwendungsbaustein`, `Dokumententyp` und `Nachrichtentyp` assoziiert, d. h. auch `Begriffssysteme` können gespeichert und kommuniziert werden. Um für ein `Begriffssystem` ausdrücken zu können, dass es einen `Master-Anwendungsbaustein` hat, ist die Klasse ebenfalls, wie auch `Objekttyp`, über eine Assoziationsbeziehung (`hat_als_Master`) unmittelbar mit der Klasse `Anwendungsbaustein` assoziiert (Abbildung 7.9; vgl. Abschnitt 6.3.1).

Abbildung von Begriffssystemen

In manchen Anwendungsszenarios kann der Sachverhalt bestehen, dass in unterschiedlichen miteinander kommunizierenden `Anwendungsbausteinen` unterschiedliche `Begriffssysteme` zur Interpretation der Daten zur Verfügung stehen. In diesen Fällen ist eine Abbildung von verschiedenen `Begriffssystemen` auf ein führendes oder, allgemeiner, eine gegenseitige Abbildung der `Begriffssysteme` aufeinander erforderlich. Ein Ansatz zur Erfüllung dieser speziellen Integrationsanforderung ist die Verwendung von Terminologieservern ([INGENERF and DIEDRICH 1997]).

Das Thema der Abbildung von `Begriffssystemen` wird hier nicht ausführlich behandelt. Durch Ergänzung einer reflexiven Assoziationsbeziehung der Klasse `Begriffssystem` kann ausgedrückt werden, dass verschiedene `Begriffssysteme` mindestens teilweise vergleichbare Begriffe beschreiben und daher gemeinsam innerhalb eines bestimmten Anwendungsbereiches verwendet werden können. Bei einer einfachen Analyse der Kommunikation können dann mehrere `Begriffssysteme`, die über diese Assoziationsbeziehung miteinander in Beziehung stehen, wie ein einziges behandelt werden. Die Assoziationsbeziehung könnte z. B. den Namen `„ist_gleichwertig_zu“` oder `„ist_Ersatz_fuer“` erhalten.

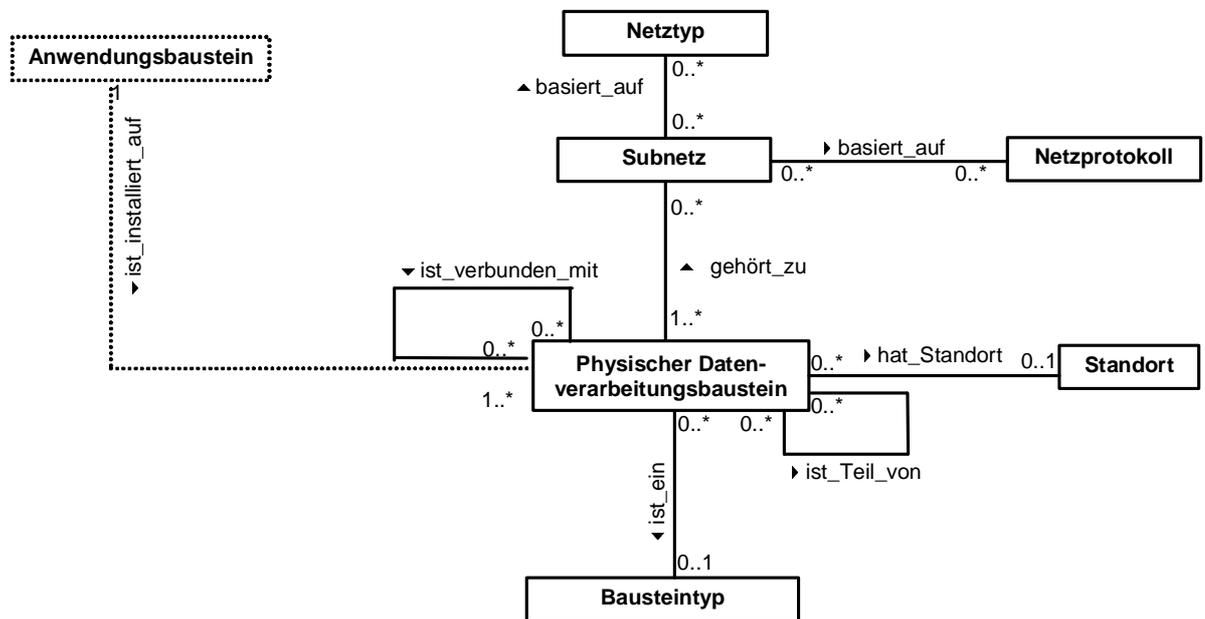


Abbildung 7.10: Physische Werkzeugebene des 3LGM_A²

Teil III

Architekturbewertung

8 Theoretische Vorbereitung der Integrationsbewertung

8.1 Vorbemerkungen

Dieses Kapitel bereitet, aufbauend auf den Grundlagen zum 3LGM² in Kapitel 6 und der Erweiterung des 3LGM² zum 3LGM_A² in Kapitel 7, die Erarbeitung von Ansätzen zur Integrationsbewertung in den folgenden Kapiteln vor. Es werden Interpretationsregeln für das 3LGM_A² und weitere theoretische Hilfsmittel definiert, die die Formalisierung der Bewertung ermöglichen. Dazu gehören

- Interpretationsregeln für Modelle, die Elementhierarchien auf der Basis von `ist_Teil_von`-Beziehungen enthalten,
- Prädikate, die bestimmten Assoziationsbeziehungen des 3LGM_A² entsprechen,
- ergänzende Prädikate für das Ausdrücken komplexerer Sachverhalte unter Nutzung der erstgenannten Prädikate und
- eine Tupelnotation für Kommunikationsbeziehungen.

Hinweise

Hinweis zur Syntax: Das Kapitel greift auf die in den Kapiteln 6 und 7 vorgestellten Elemente des 3LGM² bzw. 3LGM_A² zurück. Wie dort sind die Begriffe des 3LGM² bzw. 3LGM_A² in einer anderen Schriftart gesetzt. Auch die hier eingeführten Prädikate sind in einer anderen Schriftart gesetzt.

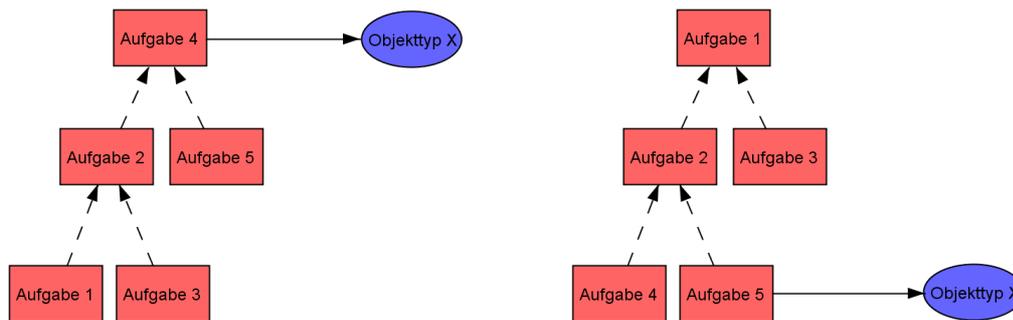
Hinweise zur Ausdrucksweise: Anstelle der längeren Formulierung „Eintreten eines Ereignisses des Typs *ergt*“ wird im Folgenden immer kurz „Eintreten des Ereignistyps *ergt*“ bzw. kürzer „Eintreten von *ergt*“ geschrieben. Ebenfalls zur Vereinfachung wird für einen Anwendungsbau-stein *awb* und einen Objekttyp *objt* immer „*awb* benötigt *objt*“ anstelle von „*awb* benötigt Daten zu *objt*“ geschrieben.

8.2 Die Interpretation von Elementhierarchien

Das 3LGM_A² sieht, wie bereits das 3LGM², `ist_Teil_von`-Beziehungen für verschiedene Elementtypen vor (vgl. Abbildungen 6.1a, 6.2a und 6.3a). Damit können Hierarchien von Elementen desselben Typs (von Instanzen derselben Elementklasse) gebildet werden (Abbildung 8.1). Im Zusammenhang mit der Bildung solcher Elementhierarchien muss die Interpretation der übrigen Assoziationsbeziehungen, z. B. `greift_zu_auf`, präzisiert werden.

Wenn keine über `ist_Teil_von`-Beziehungen modellierten Elementhierarchien existieren, können die übrigen Assoziationsbeziehungen des 3LGM_A² zunächst ohne zusätzliche Vereinbarungen interpretiert werden. Eine Instanz der Assoziationsbeziehung `greift_zu_auf` zwischen einer Aufgabe *auf*₁ und einem Objekttyp *objt* bedeutet ohne zusätzliche Definitionen, dass bei der Erfüllung von *auf*₁ auf Informationen zu Objekten des Typs *objt* zugegriffen wird¹.

¹ Zur Vereinfachung wird ein derartiger Sachverhalt im folgenden in der verkürzten Form „*auf*₁ greift auf *objt*“



Aufgabe 1 greift auf Objekttyp X zu (bearbeitend).



Abbildung 8.1: Interpretation von `ist_Teil_von`-Beziehungen

Fraglich ist nun, ob auf *objt* auch dann von *auf* zugegriffen wird, wenn beide nicht unmittelbar über eine `greift_zu_auf`-Beziehung verknüpft sind. Wie wird ein Modell interpretiert, in dem *auf*₁ mit Hilfe von `ist_Teil_von`-Beziehungen über- oder untergeordnete Aufgaben zugeordnet wurden, die mit *objt* verknüpft sind?

Für die Auswertung von `ist_Teil_von`-Beziehungen wird hier definiert:

Definition 8.1 Für ein Modellelement *me* in einem $3LGM^2_A$ -basierten Modell gelten alle Angaben bzw. Beziehungen zu anderen Modellelementen, die für übergeordnete Elemente von *me* gelten. Die Gültigkeit besteht dabei auch transitiv über mehrere aufeinanderfolgende `ist_Teil_von`-Beziehungen.

Angaben bzw. Beziehungen, die für untergeordnete Elemente, d. h. Teilelemente, von *me* gelten, gelten ohne weitere Angaben oder Beziehungen nicht notwendigerweise für *me*.

Ende — Definition

Bezogen auf die genannten Beispielelemente *auf*₁ und *objt* bedeutet die Definition, dass das Zugreifen auf *objt* durch eine übergeordnete Aufgabe *auf*₂ von *auf*₁ auch das Zugreifen auf *objt* durch *auf*₁ impliziert. Wenn die zusätzliche Aufgabe *auf*₂ Teilaufgabe von *auf*₁ ist, kann aus dem Zugreifen von *auf*₂ auf *objt* jedoch nicht das Zugreifen von *auf*₁ auf *objt* abgeleitet werden (vgl. Abbildung 8.1).

8.3 Mengen für Klassen des $3LGM^2_A$

In den theoretischen Vorbereitungen der folgenden Abschnitte und auch bei der Anwendung dieser Vorbereitungen in den folgenden Kapiteln werden sehr oft Mengensymbole angegeben. Diese Mengensymbole werden verwendet, um die Zugehörigkeit von bestimmten Variablen zu

zu“ o. ä. formuliert.

den Instanzmengen der Klassen des $3LGM_A^2$ zu kennzeichnen. Die Angabe $auf \in \underline{AUF}$ bedeutet beispielsweise, dass die Variable auf zur Menge aller **Aufgaben** gehört, d. h. dass auf für eine Instanz der Klasse **Aufgabe** steht.

Für die verwendeten Mengensymbole werden hier semiformale Definitionen angegeben:

1. Mengen für Klassen der fachlichen Ebene des $3LGM_A^2$:

$$\underline{AUF} := \{x \mid x \text{ ist eine Aufgabe.}\} \quad (8.1)$$

$$\underline{GZA} := \{x \mid x \text{ ist eine greift_zu_auf-Beziehung.}\} \quad (8.2)$$

$$\underline{OBJT} := \{x \mid x \text{ ist ein Objekttyp.}\} \quad (8.3)$$

$$\underline{OE} := \{x \mid x \text{ ist eine Organisationseinheit.}\} \quad (8.4)$$

$$\underline{ROL} := \{x \mid x \text{ ist eine Rolle.}\} \quad (8.5)$$

$$\underline{WEI} := \{x \mid x \text{ ist eine wird_erledigt_in-Beziehung.}\} \quad (8.6)$$

$$(8.7)$$

2. Mengen für Klassen der logischen Werkzeugebene des $3LGM_A^2$:

$$\underline{AWB} := \underline{RAWB} \cup \underline{PAWB} \quad (8.8)$$

$$\underline{BNS} := \{x \mid x \text{ ist eine Benutzungsschnittstelle.}\} \quad (8.9)$$

$$\underline{BSS} := \{x \mid x \text{ ist eine Bausteinschnittstelle.}\} \quad (8.10)$$

$$\underline{DBVS} := \{x \mid x \text{ ist ein Datenbankverwaltungssystem.}\} \quad (8.11)$$

$$\underline{DOKT} := \{x \mid x \text{ ist ein Dokumententyp.}\} \quad (8.12)$$

$$\underline{ERGT} := \{x \mid x \text{ ist ein Ereignistyp.}\} \quad (8.13)$$

$$\underline{KA} := \{x \mid x \text{ ist eine kann_aufrufen-Beziehung.}\} \quad (8.14)$$

$$\underline{KOMB} := \{x \mid x \text{ ist eine Kommunikationsbeziehung.}\} \quad (8.15)$$

$$\underline{KOMS} := \{x \mid x \text{ ist ein Kommunikationsstandard.}\} \quad (8.16)$$

$$\underline{NACT} := \{x \mid x \text{ ist ein Nachrichtentyp.}\} \quad (8.17)$$

$$\underline{PAWB} := \{x \mid x \text{ ist ein papierbasierter Anwendungsbaustein.}\} \quad (8.18)$$

$$\underline{RAWB} := \{x \mid x \text{ ist ein rechnerbasierter Anwendungsbaustein.}\} \quad (8.19)$$

$$\underline{SB} := \{x \mid x \text{ ist eine stellt_bereit-Beziehung.}\} \quad (8.20)$$

$$\underline{SOP} := \{x \mid x \text{ ist ein Standardorganisationsplan.}\} \quad (8.21)$$

$$\underline{SWP} := \{x \mid x \text{ ist ein Softwareprodukt.}\} \quad (8.22)$$

$$(8.23)$$

3. Mengen für Klassen der physischen Werkzeugebene des $3LGM_A^2$:

$$\underline{PDVB} := \{x \mid x \text{ ist ein physischer Datenverarbeitungsbaustein.}\} \quad (8.24)$$

8.4 Prädikate zu Assoziationsbeziehungen des $3LGM_A^2$

In den folgenden Kapiteln zur Bewertung der Informationssystemarchitektur wird mehrfach ein formales Hilfsmittel benötigt, mit dem Tatsachen hinsichtlich der Beziehungen bestimmter Modellelemente, d. h. bestimmter Instanzen der Klassen des $3LGM_A^2$, ausgedrückt werden kann.

Definition 8.2 Für jede Assoziationsbeziehung des $3LGM_A^2$ sei ein gleichnamiges Prädikat verfügbar. Für die als Parameter anzugebenden Modellelemente zeigt das betreffende Prädikat an, ob zwischen diesen Modellelementen eine Instanz der zugehörigen Assoziationsbeziehung

existiert².

Ende — Definition

Für die fachliche Ebene des $3LGM_A^2$ stehen mit Definition 8.2 u. a. folgende Prädikate zur Verfügung³ (vgl. Kapitel 6.2.1 und Abschnitt 7.2.1):

$$\text{ist_Teil_von}(auf_1 \in \underline{AUF}, auf_2 \in \underline{AUF}) \quad := \quad auf_1 \text{ ist Teil von } auf_2. \quad (8.25)$$

$$\text{greift_zu_auf}(auf \in \underline{AUF}, objt \in \underline{OBJT}) \quad := \quad auf \text{ greift auf } objt \text{ zu.} \quad (8.26)$$

$$\text{loest_aus}(ergt \in \underline{ERGT}, gza \in \underline{GZA}) \quad := \quad ergt \text{ löst aus } gza. \quad (8.27)$$

$$\text{wird_erledigt_in}(auf_1 \in \underline{AUF}, oe \in \underline{OE}) \quad := \quad auf \text{ wird erledigt in } oe. \quad (8.28)$$

Für die logische Werkzeugebene des $3LGM_A^2$ stehen mit Definition 8.2 u. a. folgende Prädikate zur Verfügung (vgl. Abschnitte 6.3.2 und 7.2.2):

$$\text{basiert_auf}(rawb \in \underline{RAWB}, swp \in \underline{SWP}) \quad := \quad rawb \text{ basiert auf } swp \text{ zu.} \quad (8.29)$$

$$\text{besitzt}(awb \in \underline{AWB}, bss \in \underline{BSS}) \quad := \quad awb \text{ besitzt } bss. \quad (8.30)$$

$$\text{stellt_bereit}(bss \in \underline{BSS}, op \in \underline{OP}) \quad := \quad bss \text{ stellt } op \text{ bereit.} \quad (8.31)$$

$$\text{kann_aufrufen}(bss \in \underline{BSS}, op \in \underline{OP}) \quad := \quad bss \text{ kann } op \text{ aufrufen.} \quad (8.32)$$

$$\text{kommunikationsbeziehung}(bss_1 \in \underline{BSS}, bss_2 \in \underline{BSS}, op \in \underline{OP}) \quad := \quad bss_1 \text{ ruft } op \text{ bei } bss_2 \text{ auf.} \quad (8.33)$$

Für die physische Werkzeugebene des $3LGM_A^2$ steht mit Definition 8.2 u. a. folgendes Prädikat zur Verfügung (vgl. Abschnitt 6.2.3):

$$\text{ist_verbunden_mit}(pdvb_1 \in \underline{PDVB}, pdvb_2 \in \underline{PDVB}) \quad := \quad pdvb_1 \text{ ist mit } pdvb_2 \text{ verbunden.} \quad (8.34)$$

Mit der in Abschnitt 8.2 definierten Interpretation der `ist_Teil_von`-Beziehungen lässt sich u. a. folgende Ableitung vornehmen:

$$\begin{aligned} & \text{ist_Teil_von}(auf_1, auf_2) \wedge \text{greift_zu_auf}(auf_2, objt) \\ & \rightarrow \text{greift_zu_auf}(auf_1, objt) \end{aligned}$$

8.5 Zusammengesetzte Prädikate für die Unterstützung der Bewertung

Auf der Basis der Prädikate zu den Assoziationsbeziehungen werden hier weitere Prädikate definiert, die das formale Bewerten der Architektur ermöglichen.

8.5.1 Die Prädikate benötigt und unterstützt

Um die Tatsache, dass ein **Anwendungsbaustein** bestimmte Informationen verarbeitet, formalisiert ausdrücken zu können, wird hier das Prädikat `benoetigt(awb, ergt, objt)` definiert. Das Prädikat gibt für einen **Anwendungsbaustein** *awb* und einen **Objekttyp** *objt* an, ob in *awb* *objt* benötigt wird.

² Die Definition nutzt die Möglichkeit, mit UML definierte Assoziationsbeziehungen als Klassen aufzufassen.

³ Auf eine vollständige Auflistung aller möglichen Prädikate wird hier verzichtet.

Die Berechnungsvorschrift für `benoetigt` ist: *awb* benötigt *objt* genau dann, wenn *awb* eine Aufgabe *auf* unterstützt, die auf *objt* zugreift:

$$\text{benoetigt}(awb \in \underline{\text{AWB}}, objt \in \underline{\text{OBJT}}) := \exists auf \in \underline{\text{AUF}} \left| \begin{array}{l} \text{unterstuetzt}(awb, auf) \wedge \\ \text{greift_zu_auf}(auf, objt) \end{array} \right. \quad (8.35)$$

Das verwendete Prädikat `unterstuetzt` gibt dabei für einen **Anwendungsbaustein** *awb* und eine **Aufgabe** *auf* an, dass *auf* mit Hilfe von *awb* erledigt wird. Die Berechnung wird, mit Ausnahme der erweiterten Form des Prädikates `wird_erledigt_in`, aus Prädikaten zusammengesetzt, die unmittelbar Assoziationsbeziehungen im 3LGM_A^2 entsprechen:

$$\text{unterstuetzt}(awb \in \underline{\text{AWB}}, auf \in \underline{\text{AUF}}) := \begin{array}{l} \exists \left(\begin{array}{l} oe \in \underline{\text{OE}} \\ wei \in \underline{\text{WEI}} \\ bns \in \underline{\text{BNS}} \end{array} \right) \left| \begin{array}{l} \text{wird_erledigt_in}(auf, oe, wei) \wedge \\ \text{mit_Hilfe_von}(wei, bns) \wedge \\ \text{besitzt}(awb, bns) \end{array} \right. \wedge \\ \vee \\ \exists \left(\begin{array}{l} oe \in \underline{\text{OE}} \\ wei \in \underline{\text{WEI}} \\ op \in \underline{\text{OP}} \\ bss \in \underline{\text{BSS}} \end{array} \right) \left| \begin{array}{l} \text{wird_erledigt_in}(auf, oe, wei) \wedge \\ \text{mit_Hilfe_von}(wei, op) \wedge \\ \text{stellt_bereit}(bss, op) \wedge \\ \text{besitzt}(awb, bss) \end{array} \right. \wedge \end{array} \quad (8.36)$$

Die besondere Form des Prädikates `wird_erledigt_in` zur Assoziationsklasse `wird_erledigt_in` des 3LGM_A^2 wurde gewählt, um die ggf. existierende Beziehung zwischen *auf* und der **Organisationseinheit** *org* als Parameter für das Prädikat `mit_Hilfe_von` verwenden zu können. Das Prädikat ist folgendermaßen semiformal definiert:

$$\text{wird_erledigt_in}(auf \in \underline{\text{AUF}}, oe \in \underline{\text{OE}}, wei \in \underline{\text{WEI}}) := \text{auf wird in } oe \text{ erledigt und } wei \text{ ist die Instanz der} \quad (8.37) \\ \text{Assoziationsklasse } \text{wird_erledigt_in}, \text{ die diese} \\ \text{Tatsache ausdrückt.}$$

8.5.2 Das Prädikat `wird_angewendet_auf`

Mit Hilfe des Prädikates `wird_angewendet_auf` kann ausgedrückt werden, dass bei der Interpretation oder Bearbeitung eines **Objekttyps** ein bestimmtes **Begriffssystem** zu berücksichtigen ist bzw. angewendet wird.

Das Prädikat `wird_angewendet_auf` gibt an, dass ein bestimmtes **Begriffssystem** bei Zugriffen auf einen bestimmten **Objekttyp** berücksichtigt wird:

$$\text{wird_angewendet_auf}(bgs \in \underline{\text{BGS}}, objt \in \underline{\text{OBJT}}) := \exists \left(\begin{array}{l} auf \in \underline{\text{AUF}} \\ gza \in \underline{\text{GZA}} \end{array} \right) \left| \begin{array}{l} \text{greift_zu_auf}(auf, objt, gza) \\ \text{unter_Beruecksichtigung_von}(gza, bgs) \end{array} \right. \wedge \quad (8.38)$$

Die hier verwendete erweiterte Form des Prädikates `greift_zu_auf` zur Assoziationsklasse `greift_zu_auf` des 3LGM_A^2 wurde gewählt, um die ggf. existierende Beziehung zwischen *auf* und *objt* als Parameter für das Prädikat `unter_Beruecksichtigung_von` verwenden zu können. Das Prädikat ist in der erweiterten Form folgendermaßen semiformal definiert:

$$\text{greift_zu_auf}(auf \in \underline{\text{AUF}}, objt \in \underline{\text{OBJT}}, gza \in \underline{\text{GZA}}) := \text{auf greift auf } objt \text{ zu und } gza \text{ ist die Instanz} \quad (8.39) \\ \text{der Assoziationsklasse } \text{greift_zu_auf}, \text{ die diese} \\ \text{Tatsache ausdrückt.}$$

8.5.3 Prädikate für die Analyse von Kommunikation

Übermittlung von Informationen

Objekttypen werden zwischen Anwendungsbausteinen dadurch übermittelt, dass Operationen mit bestimmten Nachrichtentypen bzw. Dokumententypen aufgerufen werden. Der Aufruf von Operationen wird durch das Eintreten von Ereignissen gesteuert. In Abschnitt 6.3.2 wurde dazu bereits die Klasse Ereignistyp vorgestellt und in Abschnitt 7.2.1 hinsichtlich ihrer Beziehungen zu anderen Klassen des 3LGM_A² überarbeitet.

Zur Beschreibung der Tatsache, dass ein Objekttyp *objt* bei Eintreten des Ereignistyps *ergt* von Anwendungsbaustein *awb*₁ an Anwendungsbaustein *awb*₂ übermittelt wird, wird hier das Prädikat *wird_uebermittelt* eingeführt:

$$\text{wird_uebermittelt} \left(\begin{array}{l} \text{objt} \in \underline{\text{OBJT}}, \\ \text{ergt} \in \underline{\text{ERGT}}, \\ \text{awb}_1 \in \underline{\text{AWB}}, \\ \text{awb}_2 \in \underline{\text{AWB}} \end{array} \right) := \exists \left(\begin{array}{l} \text{bss}_1 \in \underline{\text{BSS}} \\ \text{bss}_2 \in \underline{\text{BSS}} \\ \text{ka} \in \underline{\text{KA}} \\ \text{sb} \in \underline{\text{SB}} \\ \text{komb} \in \underline{\text{KOMB}} \\ \text{op} \in \underline{\text{OPER}} \\ \text{ndt} \in \underline{\text{NACT}} \cup \underline{\text{DOKT}} \end{array} \right) \left(\begin{array}{l} \text{besitzt}(\text{awb}_1, \text{bss}_1) \quad \wedge \\ \text{besitzt}(\text{awb}_2, \text{bss}_2) \quad \wedge \\ ((\text{kann_aufrufen}(\text{bss}_1, \text{op}, \text{ka}) \wedge \\ \text{stellt_bereit}(\text{bss}_2, \text{op}, \text{sb}) \wedge \\ \text{hat_Parametertyp}(\text{op}, \text{ndt}) \wedge \\ \text{kommunikationsbeziehung}(\text{ka}, \text{sb}, \text{komb})) \vee \\ (\text{kann_aufrufen}(\text{bss}_2, \text{op}, \text{ka}) \wedge \\ \text{stellt_bereit}(\text{bss}_1, \text{op}, \text{sb}) \wedge \\ \text{hat_Ergebnistyp}(\text{op}, \text{ndt}) \wedge \\ \text{kommunikationsbeziehung}(\text{ka}, \text{sb}, \text{komb}))) \wedge \\ \text{wird_repraesentiert_durch}(\text{objt}, \text{ndt}) \quad \wedge \\ \text{wird_aktiv_bei}(\text{komb}, \text{ergt}) \end{array} \right) \quad (8.40)$$

Das verwendete Prädikat *kommunikationsbeziehung* entspricht der Assoziationsklasse *Kommunikationsbeziehung* des 3LGM_A² und ähnelt in seiner erweiterten Form dem in der Definitionsformel 8.36 verwendeten Prädikat *wird_erledigt_in*. Es erlaubt die Weiterverwendung der ggf. existierenden *Kommunikationsbeziehung* zwischen *bss*₁ und *bss*₂ als Parameter für das Prädikat *wird_aktiv_bei*. Das Prädikat ist folgendermaßen semiformal definiert:

$$\text{kommunikationsbeziehung} \left(\begin{array}{l} \text{ka} \in \underline{\text{KA}}, \text{sb} \in \underline{\text{SB}}, \\ \text{komb} \in \underline{\text{KOMB}} \end{array} \right) := \text{ka und sb stehen in einer Kommunikationsbeziehung zueinander und komb ist die Instanz der Assoziationsklasse Kommunikationsbeziehung, die diese Tatsache ausdrückt.} \quad (8.41)$$

Gleiches gilt für die erweiterten Formen der Prädikate *kann_aufrufen* und *stellt_bereit*, die die Weiterverwendung der Variablen *ka* und *sb* als Parameter des Prädikates *kommunikationsbeziehung* ermöglichen:

$$\text{kann_aufrufen}(\text{bss} \in \underline{\text{BSS}}, \text{op} \in \underline{\text{OPER}}, \text{ka} \in \underline{\text{KA}}) := \text{bss kann op aufrufen und ka ist die Instanz der Assoziationsklasse kann_aufrufen, die diese Tatsache ausdrückt.} \quad (8.42)$$

$$\text{stellt_bereit}(\text{bss} \in \underline{\text{BSS}}, \text{op} \in \underline{\text{OPER}}, \text{sb} \in \underline{\text{SB}}) := \text{bss stellt op bereit und sb ist die Instanz der Assoziationsklasse stellt_bereit, die diese Tatsache ausdrückt.} \quad (8.43)$$

Das Prädikat *wird_uebermittelt* ist transitiv auszuwerten, d. h. wenn *wird_uebermittelt(objt, ergt, awb₁, awb₂)* und *wird_uebermittelt(objt, ergt, awb₂, awb₃)* gelten, dann gilt auch *wird_uebermittelt(objt, ergt, awb₁, awb₃)*.

Übermittlung von Informationen zu Begriffssystemen

Im 3LGM_A² ist auch für Begriffssysteme die Modellierung der Speicherung und der Kommunikation vorgesehen (vgl. Abschnitt 7.7). Daher kann auch für Begriffssysteme die Übermittlung formal untersucht werden.

Wie für Objekttypen wird hier ein Prädikat definiert, das die Übermittlung eines Begriffssystems bgs bei Eintreten eines Ereignistyps $ergt$ von Anwendungsbaustein awb_1 an Anwendungsbaustein awb_2 anzeigt:

$$\text{wird_uebermittelt} \left(\begin{array}{l} bgs \in \underline{\text{BGS}}, \\ ergt \in \underline{\text{ERGT}}, \\ awb_1 \in \underline{\text{AWB}}, \\ awb_2 \in \underline{\text{AWB}} \end{array} \right) := \exists \left(\begin{array}{l} bss_1 \in \underline{\text{BSS}} \\ bss_2 \in \underline{\text{BSS}} \\ ka \in \underline{\text{KA}} \\ sb \in \underline{\text{SB}} \\ komb \in \underline{\text{KOMB}} \\ op \in \underline{\text{OPER}} \\ ndt \in \underline{\text{NACT}} \cup \underline{\text{DOKT}} \end{array} \right) \left(\begin{array}{l} \text{besitzt}(awb_1, bss_1) \quad \wedge \\ \text{besitzt}(awb_2, bss_2) \quad \wedge \\ ((\text{kann_aufrufen}(bss_1, op, ka) \wedge \\ \text{stellt_bereit}(bss_2, op, sb) \wedge \\ \text{hat_Parametertyp}(op, ndt) \wedge \\ \text{kommunikationsbeziehung}(ka, sb, komb)) \vee \\ (\text{kann_aufrufen}(bss_2, op, ka) \wedge \\ \text{stellt_bereit}(bss_1, op, sb) \wedge \\ \text{hat_Ergebnistyp}(op, ndt) \wedge \\ \text{kommunikationsbeziehung}(ka, sb, komb))) \wedge \\ \text{wird_repraesentiert_durch}(bgs, ndt) \quad \wedge \\ \text{wird_aktiv_bei}(komb, ergt) \end{array} \right) \quad (8.44)$$

Auch das Prädikat $\text{wird_uebermittelt_b}$ ist, wie wird_uebermittelt , transitiv auszuwerten.

Aufrufen von Funktionalität

Über das in der Formel 8.36 definierte Prädikat unterstuetzt wird ausgedrückt, dass ein bestimmter Anwendungsbaustein Funktionalität zur Erfüllung einer oder mehrerer Aufgaben implementiert. Wie bereits am Ende von Abschnitt 7.2.2 beschrieben, ist bei sehr detaillierter Betrachtung der Funktionalitätsbereitstellung eine entsprechend detaillierte Modellierung von Aufgaben erforderlich.

Das Bereitstellen der implementierten Funktionalität durch den betreffenden Anwendungsbaustein für andere Anwendungsbausteine wird mit Hilfe der Klassen Bausteinschnittstelle und Operation modelliert.

Ähnlich der Übermittlung von Informationen kann auch modelliert werden, dass Anwendungsbausteine Funktionalität, die durch andere Anwendungsbausteine bereitgestellt wird, nutzen bzw. aufrufen. Funktionalität, d. h. die Fähigkeit zur Erfüllung bestimmter Aufgaben, wird von Anwendungsbausteinen dadurch von anderen Anwendungsbausteinen aufgerufen, dass Operationen beim Eintreten bestimmter Ereignistypen aufgerufen werden.

Zur Beschreibung der Tatsache, dass Funktionalität zur Erfüllung einer Aufgabe auf bei Eintreten des Ereignistyps $ergt$ von Anwendungsbaustein awb_1 bei Anwendungsbaustein awb_2 aufgerufen wird, dient hier das Prädikat ruft_auf :

$$\text{ruft_auf} \left(\begin{array}{l} auf, ergt, \\ awb_1, awb_2 \end{array} \right) := \exists \left(\begin{array}{l} bss_1 \in \underline{\text{BSS}} \\ bss_2 \in \underline{\text{BSS}} \\ komb \in \underline{\text{KOMB}} \\ op \in \underline{\text{OPER}} \\ nact \in \underline{\text{NACT}} \end{array} \right) \left(\begin{array}{l} \text{besitzt}(awb_1, bss_1) \quad \wedge \\ \text{besitzt}(awb_2, bss_2) \quad \wedge \\ \text{stellt_bereit}(bss_2, op, sb) \quad \wedge \\ \text{kann_aufrufen}(bss_1, op, ka) \quad \wedge \\ \text{kommunikationsbeziehung}(ka, sb, komb) \wedge \\ \text{unterstuetzt}(op, auf) \quad \wedge \\ \text{wird_aktiv_bei}(komb, ergt) \end{array} \right) \quad (8.45)$$

Das Prädikat ruft_auf ist transitiv auszuwerten, d. h. wenn $\text{ruft_auf}(auf, ergt, awb_1, awb_2)$

und $\text{ruft_auf}(auf, \text{ergt}, awb_2, awb_3)$ gelten, dann gilt auch $\text{ruft_auf}(auf, \text{ergt}, awb_1, awb_3)$.

8.6 Formalisierung von Kommunikationsverbindungen

Aufbauend auf der Definition aus Abschnitt 7.5 wird der Begriff *Kommunikationsverbindung* hier formalisiert.

8.6.1 Kommunikationsverbindungen als Folgen von Kommunikationsbeziehungen

Das 3LGM² definiert *Kommunikationsbeziehungen* als Assoziationsbeziehungen, die zwei Bausteinschnittstellen miteinander verbinden. Für die beiden Bausteinschnittstellen muss gelten, dass die aufgerufene Bausteinschnittstelle eine Operation bereit stellt, die die aufrufende Bausteinschnittstelle aufrufen kann.

Jede *Kommunikationsbeziehung* kann als Tripel (bss_1, bss_2, op) aufgefasst werden:

$$kb = (bss_1, bss_2, op) \left| \begin{array}{l} bss_1 := \text{aufrufende Bausteinschnittstelle} \\ bss_2 := \text{aufgerufene Bausteinschnittstelle} \\ op := \text{aufgerufene Operation} \end{array} \right. \quad (8.46)$$

Jeder Weg zur Übermittlung von Daten bzw. Funktionalität von einem *Anwendungsbaustein* zu einem anderen wird hier als *Kommunikationsverbindung* bezeichnet. Eine *Kommunikationsverbindung* kann dabei aus mehreren *Kommunikationsbeziehungen* bestehen.

Jede *Kommunikationsverbindung* kann als eine Folge von *Kommunikationsbeziehungen* betrachtet werden:

$$kv = \langle kb_i \rangle = \langle kb_1, \dots, kb_n \rangle = \langle (bss_{1,1}, bss_{1,2}, op_1), \dots, (bss_{n,1}, bss_{n,2}, op_n) \rangle \quad (8.47)$$

Es ergibt sich unmittelbar, was unter der Länge einer *Kommunikationsverbindung* verstanden wird: Die Länge einer *Kommunikationsverbindung* ist gleich der Anzahl der zugehörigen *Kommunikationsbeziehungen*:

$$L(kv) := |\langle kb_i \rangle| \quad (8.48)$$

8.6.2 Vermittlung und Vermittlungstiefe

Kommunikationsbeziehungen können, entsprechend der in Abschnitt 7.2.2 beschriebenen Erweiterung des 3LGM², über spezielle *Kommunikationsbeziehungen* vermittelt werden (vgl. auch Abschnitt 7.4.2). In der hier gewählten Tupelnotation für *Kommunikationsbeziehungen* wird Vermittlung als zusätzliches Tupelelement notiert. Eine *Kommunikationsbeziehung* kann also als *Quadrupel* $(bss_1, bss_2, op, \langle vb_1, \dots, vb_n \rangle)$ aufgefasst werden:

$$kb = (bss_1, bss_2, op, \langle vb_1, \dots, vb_n \rangle) \left| \begin{array}{l} bss_1 := \text{aufrufende Bausteinschnittstelle} \\ bss_2 := \text{aufgerufene Bausteinschnittstelle} \\ op := \text{aufgerufene Operation} \\ \langle vb_1, \dots, vb_n \rangle := \text{Vermittlungsverbindung als Folge der zugehörigen Kommunikationsbeziehungen} \end{array} \right. \quad (8.49)$$

In vielen Fällen werden vermittelnde *Kommunikationsbeziehungen* (= *Vermittlungsbezie-*

$$\begin{aligned}
 & (bss_1, bss_2, op, \langle (vbss_{1,1}, vbss_{1,2}, vop_1, \langle (vbs_{1,1,1}, vbs_{1,1,2}, vvop_{1,1}, \langle \rangle) \rangle) \rangle, (vbss_{2,1}, vbss_{2,2}, vop_2, \langle \rangle) \rangle) \\
 & \text{oder übersichtlicher:} \\
 & (bss_1, bss_2, op, \langle (vbss_{1,1}, vbss_{1,2}, vop_1, \langle (vbs_{1,1,1}, vbs_{1,1,2}, vvop_{1,1}, \langle \rangle) \rangle) \rangle, (vbss_{2,1}, vbss_{2,2}, vop_2, \langle \rangle) \rangle)
 \end{aligned}$$

Abbildung 8.2: Beispiel für eine Kommunikationsverbindung mit Vermittlungsbeziehungen

hungen) selbst durch weitere **Kommunikationsbeziehungen** vermittelt. Mit der Anwendung der Quadrupelnotation für die vermittelnden **Kommunikationsbeziehungen** kann diese Tatsache ausgedrückt werden, wodurch sich bei ausführlicher Notation einer **Kommunikationsbeziehung** möglicherweise Verschachtelungen ergeben. Abbildung 8.2 zeigt ein Beispiel für eine Kommunikationsverbindung mit Vermittlungsbeziehungen.

Die Vermittlungstiefe einer Kommunikationsverbindung entspricht der maximalen Schachtelungstiefe der zugehörigen Vermittlungsbeziehungen:

$$T(kv) := \text{max. Schachtelungstiefe der Vermittlungsbeziehungen} \tag{8.50}$$

Im Beispiel von Abbildung 8.2 beträgt die Vermittlungstiefe also 2.

9 Die Erfüllung von Integrationsanforderungen

Die bisherigen Ausführungen zur Modellierung von Informationssystemen auf der Basis des $3LGM_A^2$ waren hauptsächlich auf das Modellieren unterschiedlicher Architekturen ohne das Vornehmen einer Bewertung bezogen. Dieses Kapitel nimmt die am Anfang der Arbeit in Kapitel 1 angekündigte Bewertungsthematik auf.

Nach einer Vorbereitung in den Abschnitten 9.1 und 9.2 wird in Abschnitt 9.3 beschrieben, wie Integrationsanforderungen auf der Basis des $3LGM_A^2$ modelliert bzw. aus existierenden Modellen abgeleitet werden können. Dabei werden die in den Abschnitten 8.2 bis 8.5 definierten Interpretationsregeln sowie Mengen- und Prädikatdefinitionen verwendet.

In den Abschnitten 9.4 und 9.5 wird beschrieben, wie die Erfüllung von Integrationsanforderungen formal auf der Basis von $3LGM_A^2$ -Modellen überprüft werden kann.

9.1 Mengen von Anwendungsbausteinen: Domänen

Mengen von **Anwendungsbausteinen**, für die bestimmte Integrationsanforderungen bestehen oder für die über Kommunikationsverbindungen Integration hergestellt wurde, werden im Folgenden als Domänen bezeichnet. Dazu werden in den Abschnitten 9.3 und 9.4 zwei Domänentypkategorien mit verschiedenen Domärentypen definiert:

Anforderungsdomänen enthalten **Anwendungsbausteine**, für die bestimmte Integrationsanforderungen bestehen. Dazu wurden in Kapitel 2 bereits Anforderungskategorien beschrieben. Zu jeder Anforderungskategorie, mit Ausnahme der physischen Integration (s. u.), wird ein Anforderungsdomärentyp definiert. Über verschiedene Parameter können, bezogen auf ein Modell, konkrete Anforderungsdomänen bestimmt werden.

Kommunikationsdomänen enthalten **Anwendungsbausteine**, bei denen über Kommunikationsverbindungen ein bestimmter Integrationsstatus hergestellt wurde. Es werden zwei Kommunikationsdomärentypen definiert, die die Integration hinsichtlich bestimmter Informationen bzw. hinsichtlich bestimmter Funktionalität fokussieren. Konkrete Kommunikationsdomänen in einem Modell werden, wie Anforderungsdomänen, über verschiedene Parameter bestimmt.

Über das Vergleichen von Anforderungsdomänen mit Kommunikationsdomänen kann die Architektur eines Informationssystems hinsichtlich der Erfüllung von Integrationsanforderungen bewertet werden. In Abschnitt 9.5 werden auf der Basis der Domärentypdefinitionen Anleitungen für das Vergleichen und Beispiele angegeben.

Das systematische Analysieren und Bewerten von physischer Integration wird, im Gegensatz zu den anderen Anforderungskategorien, nicht behandelt. Eine ausführliche Behandlung und Überarbeitung der physischen Werkzeugebene des $3LGM^2$ erfolgt in einer weiteren Dissertation mit dem Arbeitstitel „Modellierung und Bewertung der hardwarenahen Systemkonfigurationen“.

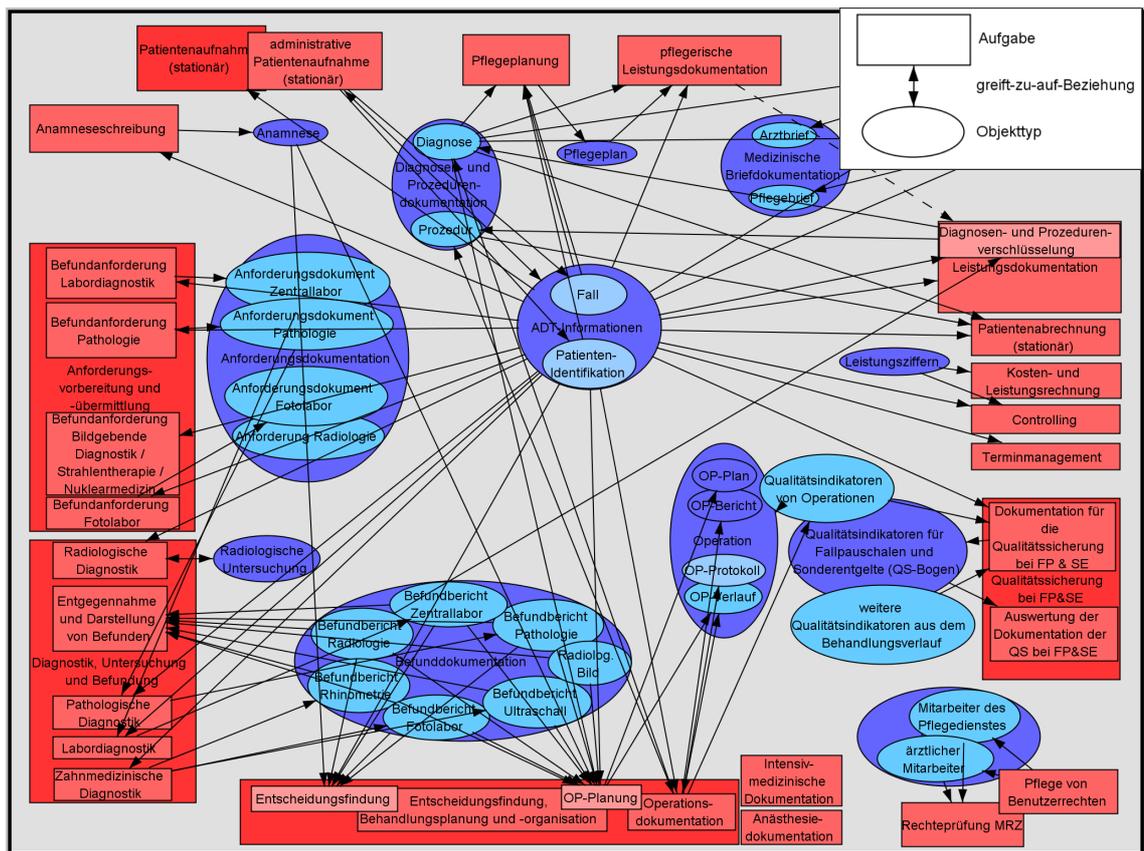


Abbildung 9.1: Auszug aus der fachlichen Ebene des Informationssystems des UKL

von KIS zur Unterstützung des strategischen IM“. Diese Dissertation wird in derselben Arbeitsgruppe erstellt, in der auch der Autor der vorliegenden Arbeit tätig ist.

9.2 Ein Anwendungsszenario

Zur Demonstration der Anwendung des in diesem Kapitel vorgestellten Domänenkonzeptes werden mehrfach Beispiele aus dem Informationssystem des Universitätsklinikums Leipzig (UKL) verwendet.

Das Informationssystem des UKL wurde auf der Basis des 3LGM² modelliert. Die in diesem Kapitel verwendeten Beispiele sind auf Auszüge dieses Modells bezogen. Je nach Beispiel wird eines von zwei Teilmodellen betrachtet: Das erste Teilmodell beschreibt das Zusammenwirken des Patientenverwaltungssystems mit anderen, für die medizinische Informationsverarbeitung genutzten **Anwendungsbausteinen**. Das zweite beschreibt spezieller das Zusammenwirken des Patientenverwaltungssystems und des R/3-basierten Klinischen Dokumentations- und Managementsystems mit dem Radiologieinformationssystem und bestimmten PACS-Komponenten. Abbildung 9.1 zeigt die für beide Teilmodelle gleiche fachliche Ebene des Informationssystems, die Abbildungen 9.2a und b zeigt die beiden unterschiedlichen logischen Werkzeugebenen. Von der physischen Werkzeugebene wird nur ein einziges Element, ein PC, genutzt.

Elemente	Hinweise
<i>Fachliche Ebene</i>	
Aufgabe <i>Patientenaufnahme (stationär)</i>	das Aufnehmen (Registrieren) stationärer Patienten
Aufgabe <i>Diagnosen- und Prozedurenverschlüsselung</i>	Gesetzlich vorgeschriebene Pflicht zur Verschlüsselung von Diagnosen und Prozeduren für die stationäre Abrechnung nach der Diagnosenklassifikation <i>ICD10</i> und der Prozedurenklassifikation <i>OPS301</i>
Aufgabe <i>Rechteprüfung MRZ</i>	Überprüfung von Benutzerrechten im Klinikumsnetzwerk des UKL
Objekttyp <i>Fall</i> → Bearbeitung löst aus: Ereignistyp <i>NP1110</i> → Interpretation löst aus: Ereignistyp <i>Kontextwechsel_Fall</i> → Interpretation wird ausgelöst durch: Ereignistyp <i>RIS_Fall_Abfrage</i>	die im Informationssystem registrierten Behandlungsfälle Kommunikationsstandard <i>SAP-HCM</i> ; wird beim Registrieren einer stationären Patientenaufnahme (Anlegen eines stationären Falles) ausgelöst wird beim Auswählen eines Behandlungsfalles in einem Anwendungsbaustein ausgelöst wird ausgelöst, wenn aus dem Radiologieinformationssystem (RIS) Falldaten aus seinem Kommunikationsmodul abgefragt werden sollen
Objekttyp <i>Diagnose</i> → Bearbeitung löst aus: Ereignistyp <i>DIAPROZ_init</i> → erfordert die Berücksichtigung von: Begriffssystem <i>ICD10</i> → Bearbeitung löst aus: Ereignistyp <i>ICD_neu</i> Begriffssystem <i>DRG</i>	die für die Abrechnung, aber auch für die Erstellung von Arztbriefen zu dokumentierenden Diagnosen wird beim Initiieren der Verschlüsselung von Diagnosen oder Prozeduren ausgelöst bei der Diagnosendokumentation anzuwendende Klassifikation wird beim Einpflegen einer aktualisierten Version des Begriffssystems <i>ICD10</i> ausgelöst bei der Abrechnung anzuwendende Klassifikation für Behandlungsfälle; berücksichtigt die zum Fall dokumentierten ICD10-Diagnosen, OPS301-Prozeduren und weitere Angaben zum Patienten oder zum Behandlungsverlauf
Objekttyp <i>Prozedur</i> → erfordert die Berücksichtigung von: Begriffssystem <i>OPS301</i> Begriffssystem <i>DRG</i>	die für die Abrechnung, aber auch für die Erstellung von Arztbriefen zu dokumentierenden medizinischen Maßnahmen bei der Prozedurendokumentation anzuwendende Klassifikation (s. o.)
Objekttyp <i>Mitarbeiter des Pflegedienstes</i> → Bearbeitung löst aus: Ereignistyp <i>ZUGR_update</i> → Interpretation löst aus: Ereignistyp <i>LOGIN_BBS</i>	Mitarbeiter des Pflegedienstes in der Klinik für Neurochirurgie; umfasst u. a. auch Informationen über Zugriffsberechtigungen dieser Mitarbeiter wird beim Aktualisieren von Zugriffsinformationen, z. B. beim Objekttyp <i>Mitarbeiter des Pflegedienstes</i> ausgelöst wird bei der Anmeldung eines Benutzers am Anwendungsbaustein <i>Bild- und Befundserver</i> ausgelöst
<i>Logische Werkzeugebene</i>	
Anwendungsbaustein <i>Patientenverwaltungssystem</i> → kann über eine Bausteinschnittstelle aufrufen: Operation <i>ADT-Nachrichtenübergabe</i>	zentrales Werkzeug für Aufgaben im Zusammenhang mit der Patientenverwaltung, z. B. das Anlegen von Fällen, das Registrieren von Verlegungen und Entlassungen sowie das Erstellen von Rechnungen; u. a. Master-Anwendungsbaustein für den Objekttyp <i>Fall</i> . nimmt Nachrichten zu ADT-Ereignissen als Parameter entgegen

Tabelle 9.1: Elemente des Beispielmodells des Informationssystems des UKL

9 Die Erfüllung von Integrationsanforderungen

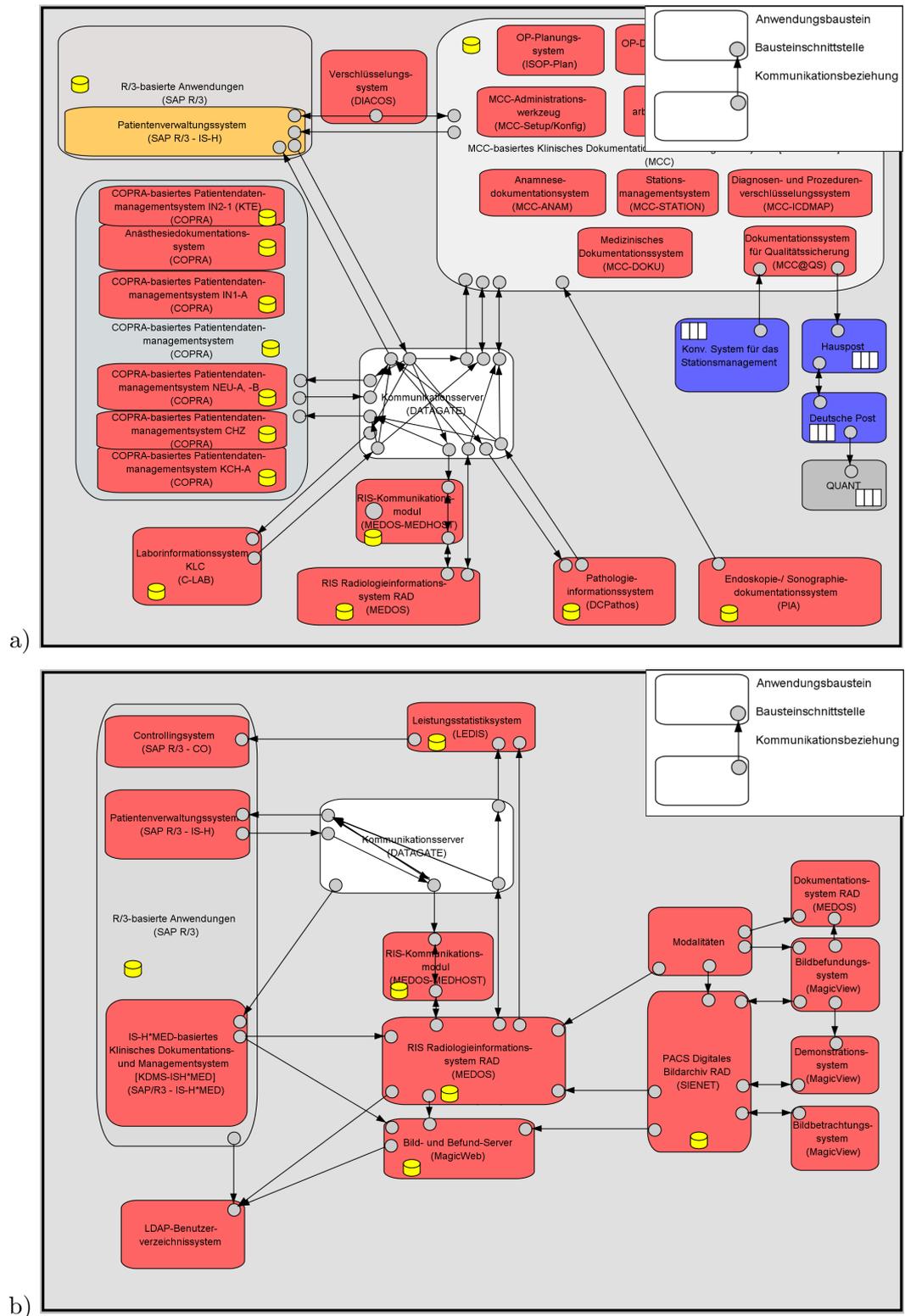


Abbildung 9.2: Auszüge aus der logischen Werkzeugebene des Informationssystems des UKL

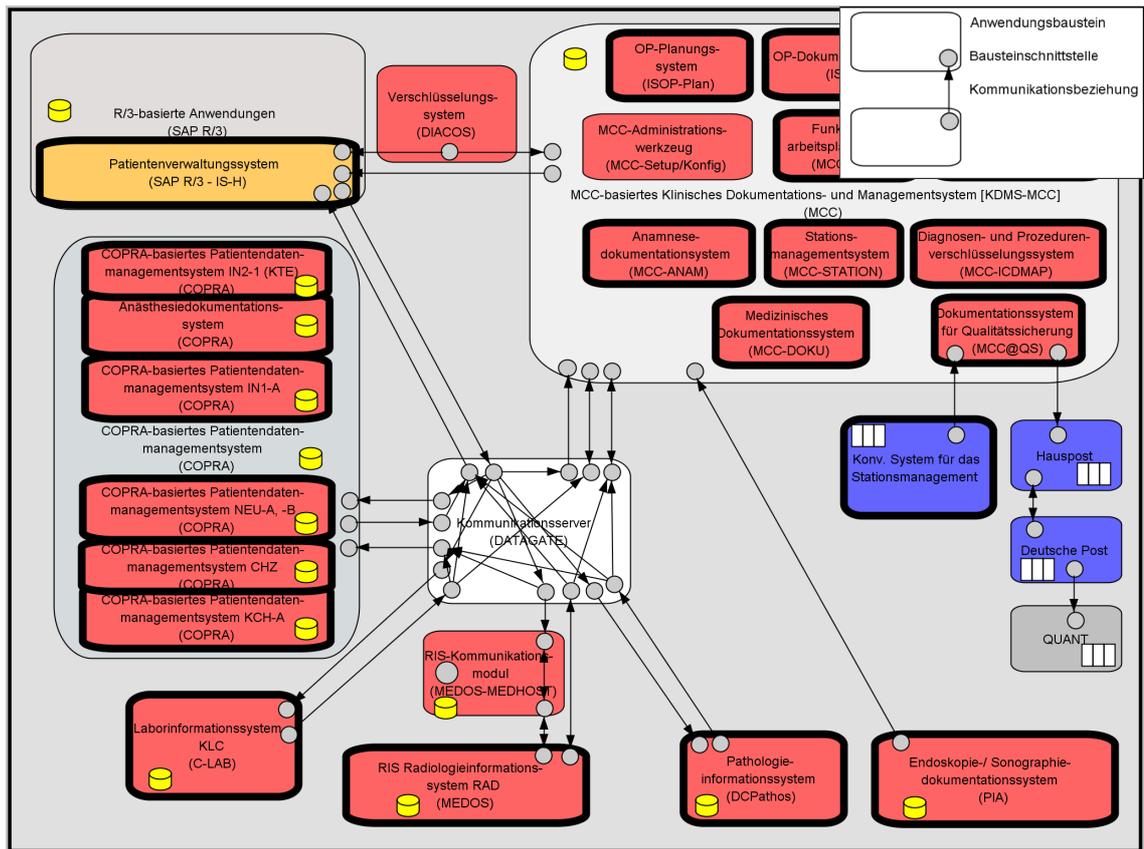


Abbildung 9.3: Datendomäne des Objekttyps Fall

ben. Die Definition sieht vor, dass eine Datendomäne jeweils für einen Objekttyp bestimmt wird.

Definition 9.1 Die **Datendomäne** eines Objekttyps *objt* ist die Menge aller Anwendungsbausteine awb_i , die *objt* benötigen.

$$\underline{DDom}(objt, ergt) := \{awb_i \in \underline{AWB} \mid \text{benoetigt}(awb_i, ergt, objt)\} \quad (9.1)$$

Ende — Definition und Berechnungsvorschrift

Abbildung 9.3 zeigt ein Beispiel für eine Datendomäne. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen Anwendungsbausteine Aufgaben unterstützen, die auf den Objekttyp *Fall* zugreifen.

9.3.2 Funktionale Domänen

Eine Menge von Anwendungsbausteinen, für die die Forderung nach funktionaler Integration besteht, wird hier als funktionale Domäne bezeichnet (vgl. Abschnitt 2.2.3). Das Benötigen einer Funktion in mehreren Anwendungssystemen im Sinne der Definition 2.3 in Abschnitt 2.2.3 wird auf der Basis des $3LGM_A^2$ dadurch ausgedrückt, dass mehrere Anwendungsbausteine

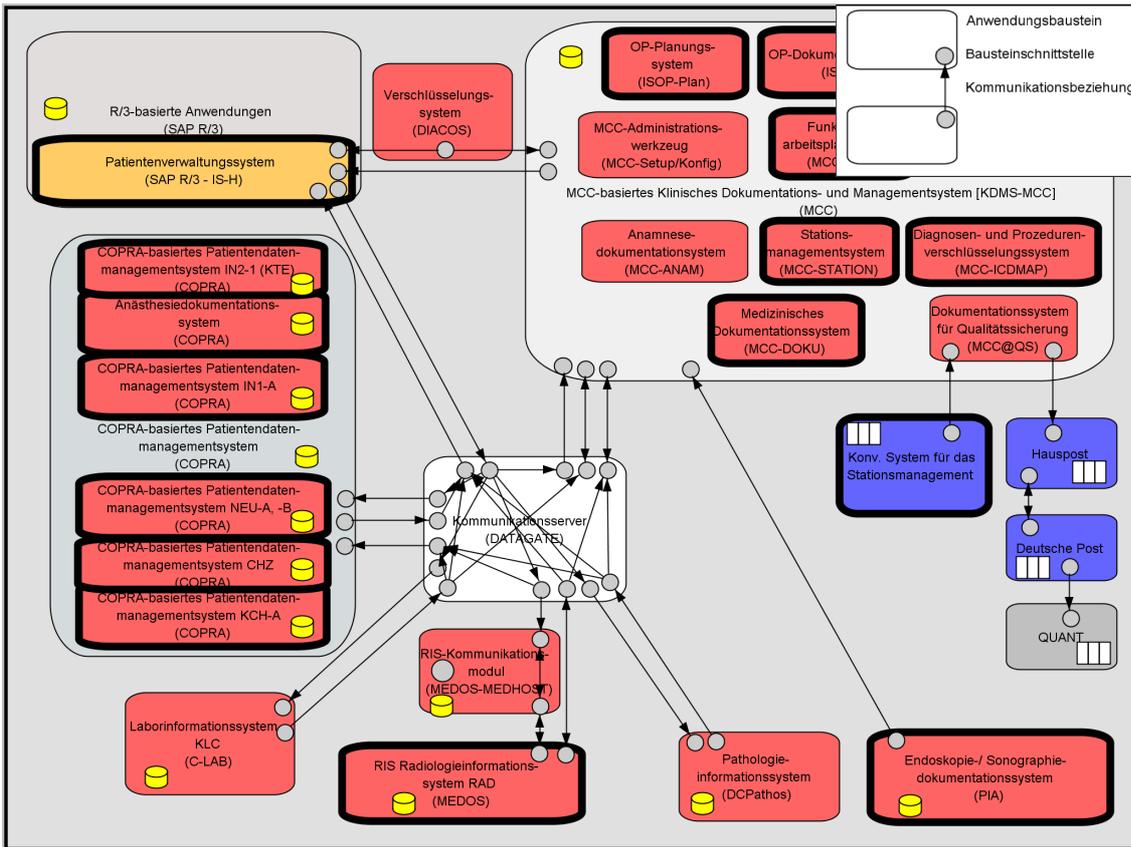


Abbildung 9.4: Funktionale Domäne der Aufgabe Diagnosen- und Prozedurenverschlüsselung

dieselbe Aufgabe unterstützen. Das Unterstützen einer Aufgabe durch einen Anwendungsbau-stein wird entsprechend dem Prädikat *unterstuetzt* (vgl. Definitionsgleichung 8.36) durch gemeinsame Nutzung verschiedener Assoziationsbeziehungen, z. B. *wird_erledigt_in* und *mit_Hilfe_von*, modelliert.

In den meisten Fällen wird eine mehrfach benötigte bzw. unterstützte Funktionalität, d. h. eine bestimmte Aufgabe, Teil anderer Funktionalitäten, d. h. anderer Aufgaben, sein. Beispiele sind die Aufgabe *Laborbefund präsentieren* als Teil der Aufgaben *Chemotherapie planen* und *Entlassung vorbereiten* sowie die Aufgabe *Verschlüsseln von Diagnosen und Prozeduren* als Teil der Aufgaben *Arztbriefschreibung*, *OP-Dokumentation* und *stationäre Leistungsdokumentation*. Durch die in Abschnitt 8.2 definierte Interpretationsregel für Elementhierarchien folgt aus dem Unterstützen einer übergeordneten Aufgabe durch einen Anwendungsbaustein auch das Unterstützen der Teilaufgaben.

Funktionale Integration muss also zwischen den Anwendungsbausteinen hergestellt werden, die dieselbe(n) Aufgabe(n) unterstützen.

Definition 9.2 Die funktionale Domäne einer Aufgabe *auf* ist die Menge aller Anwen-dungsbausteine *awb_i*, die *auf* unterstützen.

$$FDom(auf) := \{ awb_i \in AWE \mid unterstuetzt(awb_i, auf) \} \tag{9.2}$$

Abbildung 9.4 zeigt ein Beispiel für eine funktionale Domäne. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen **Anwendungsbausteine Aufgaben** unterstützen, der als Teilaufgabe die **Aufgabe Diagnosen- und Prozedurenverschlüsselung** untergeordnet ist.

9.3.3 Semantische Domänen

Eine Menge von **Anwendungsbausteinen**, für die die Forderung nach semantischer Integration besteht, wird hier als semantische Domäne bezeichnet (vgl. Abschnitt 2.2.4). Semantische Integration muss zwischen den **Anwendungsbausteinen** hergestellt werden, bei denen Daten mit Hilfe derselben Begriffssysteme interpretiert werden (vgl. auch Exkurs „Semantische Integration“ in Abschnitt 2.2.4). Dazu wurde in Abschnitt 7.7 die Klasse **Begriffssystem** in das $3LGM_A^2$ eingeführt.

Semantische Domänen werden auf der Basis von Datendomänen und des in Abschnitt 8.5.2 definierten Prädikates `wird_angewendet_auf` definiert:

Definition 9.3 Die **semantische Domäne** eines **Begriffssystems** bgs ist die Vereinigung der Datendomänen aller **Objektypen**, auf die bgs angewendet wird.

$$\underline{SDom}(bgs) := \bigcup_{i=0 \dots \underline{OBJT}} \underline{DDom}(objt_i) \mid \text{wird_angewendet_auf}(bgs, objt_i) \quad (9.3)$$

Abbildung 9.5 zeigt ein Beispiel für **Begriffssysteme** auf der fachlichen Ebene und die zugehörigen semantischen Domänen. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen **Anwendungsbausteine Aufgaben** unterstützen, die auf die **Objektypen Diagnose** und **Prozedur** zugreifen und dabei die Begriffssysteme *ICD10*, *OPS301* und *DRG* berücksichtigen.

9.3.4 Kontextdomänen

Eine Menge von **Anwendungsbausteinen**, für die die Forderung nach Kontextintegration besteht, wird hier als Kontextdomäne bezeichnet (vgl. Abschnitt 2.2.5). Kontextintegration muss z. B. für **Anwendungsbausteine** hergestellt werden, die gleichzeitig durch bestimmte Personen an einem oder mehreren **physischen Datenverarbeitungsbausteinen** genutzt werden (Personenkontext).

Herstellen von Kontextintegration bedeutet, dass **Anwendungsbausteine** hinsichtlich einer bestimmten Information, d. h. der Ausprägung eines **Objektyps**, für eine bestimmte Zeit synchronisiert werden. Die Information legt einen Arbeitsrahmen fest, der als Kontext bezeichnet wird: Benutzerkontext, Patientenkontext usw.

Damit können Kontextdomänen, ausgehend von **Objektypen** und **physischen Datenverarbeitungsbausteinen**, genauer definiert werden:

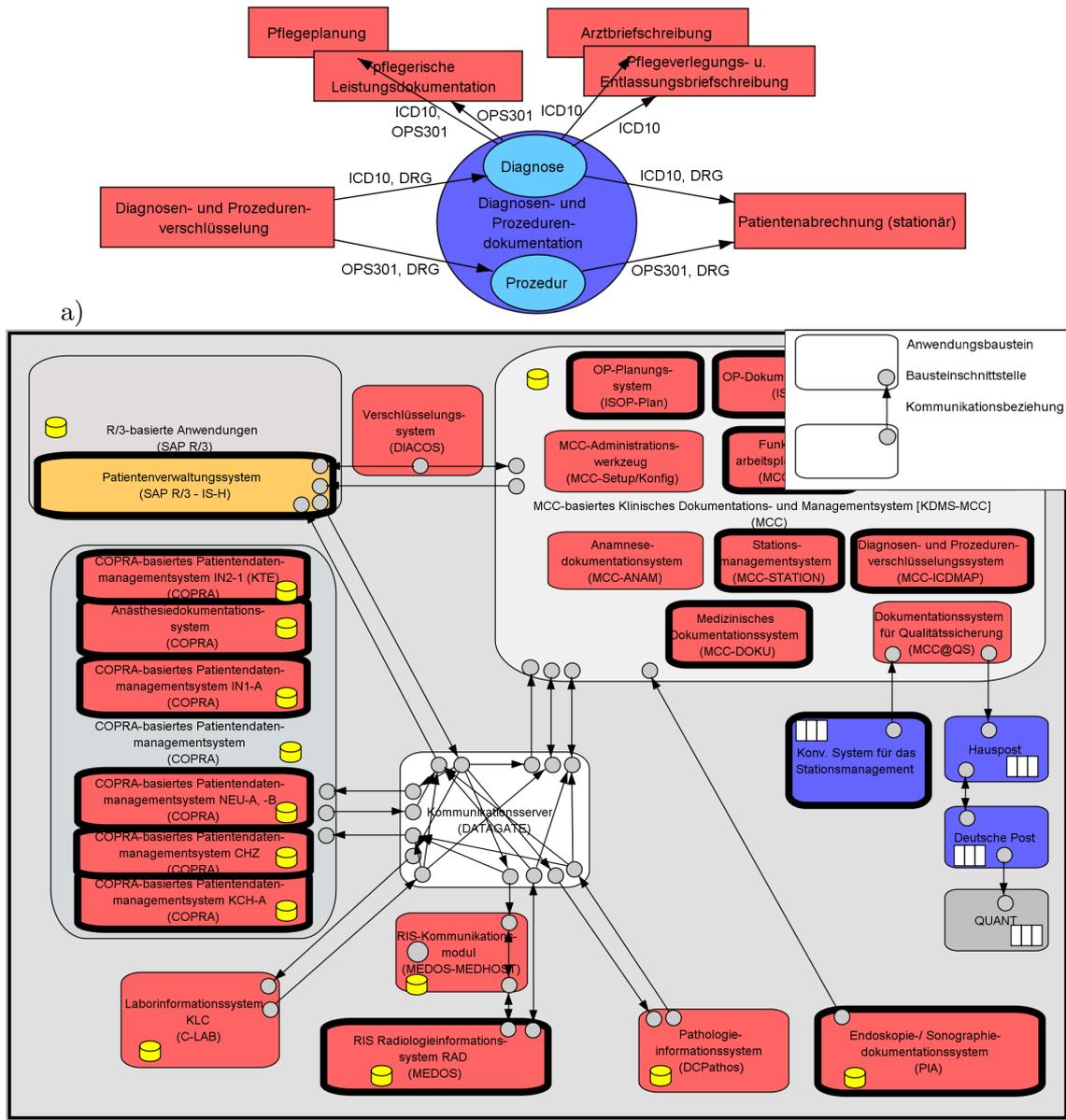


Abbildung 9.5: Zuordnung der Begriffssysteme DRG, ICD10 und OPS301 zu greift_zu_auf-Beziehungen (a) und semantische Domänen der Begriffssysteme (b)

Definition 9.4 Die **Kontextdomäne** eines Objekttyps *objt* und eines physischen Datenverarbeitungsbausteines *pdrv* ist die Datendomäne von *objt*, eingeschränkt auf diejenigen Anwendungsbausteine, die gemeinsam auf *pdrv* installiert sind:

$$KDom(objt, pdrv) := \left\{ awb_i \in \underline{AWB} \mid \begin{array}{l} \text{benoetigt}(awb_i, objt) \\ \text{ist_installiert_auf}(awb_i, pdrv) \end{array} \wedge \right\} \quad (9.4)$$

Werden mehrere physische Datenverarbeitungsbausteine gemeinsam genutzt, muss die Vereinigung der einzelnen Kontextdomänen gebildet werden.

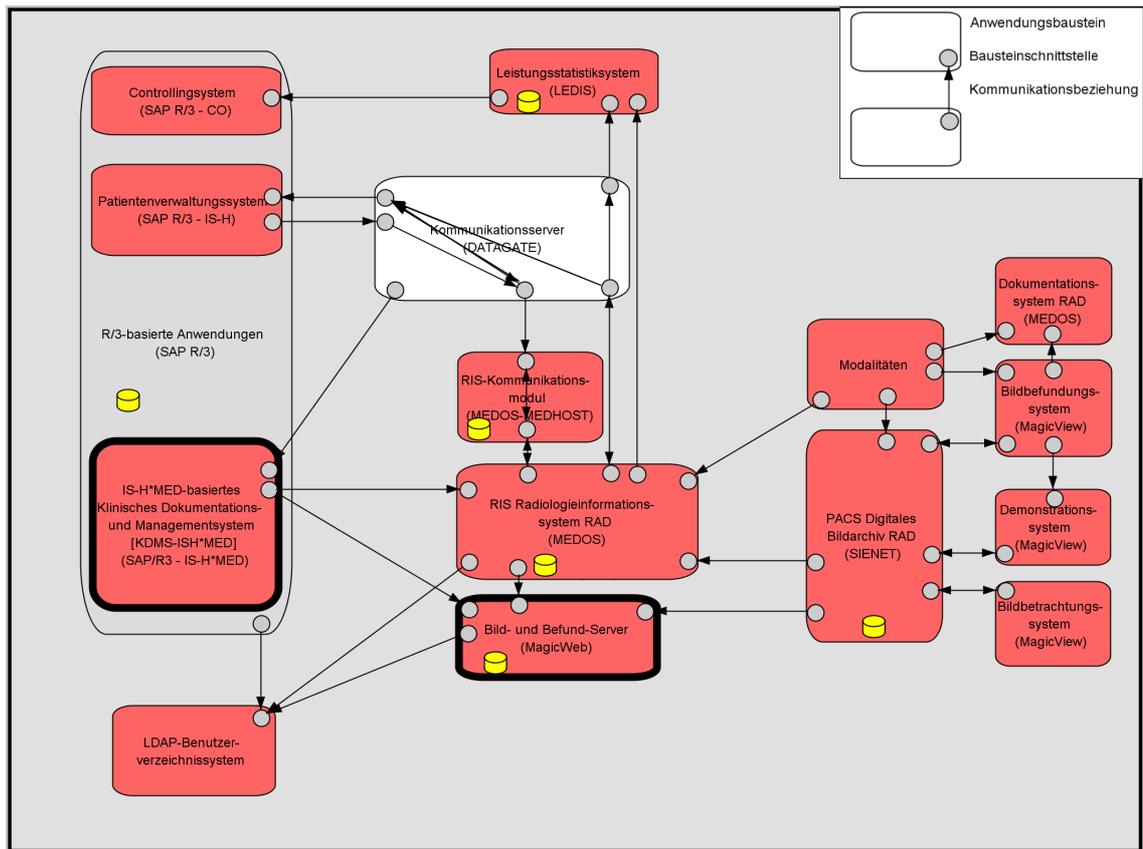


Abbildung 9.6: Kontextdomäne des Objekttyps *Fall* und des physischen Datenverarbeitungsbausteines *PC NCH 15*

Ende — Definition und Berechnungsvorschrift

In der Praxis werden i. d. R. nur die **Teilanwendungsbausteine** der betreffenden **Anwendungsbausteine** auf dem genannten PC installiert sein, mit denen auf zentrale Server-Anwendungsbausteine zugegriffen wird.

Abbildung 9.6 zeigt ein Beispiel für eine Kontextdomäne. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen **Anwendungsbausteine** hinsichtlich des Objekttyps *Fall* synchronisiert werden sollen sowie gemeinsam auf dem physischen Datenverarbeitungsbaustein *PC NCH 15* installiert sind und gemeinsam genutzt werden.

9.3.5 Präsentationsdomänen

Eine Menge von **Anwendungsbausteinen**, für die die Forderung nach Präsentationsintegration besteht, wird hier als **Präsentationsdomäne** bezeichnet (vgl. Abschnitt 2.2.6). Wie Kontextintegration muss Präsentationsintegration für **Anwendungsbausteine** hergestellt werden, die gleichzeitig durch bestimmte Personen an einem oder mehreren **physischen Datenverarbeitungsbausteinen** genutzt werden.

Bei der Präsentationsintegration fällt, im Vergleich zu Kontextdomänen, der Bezug zu Ob-

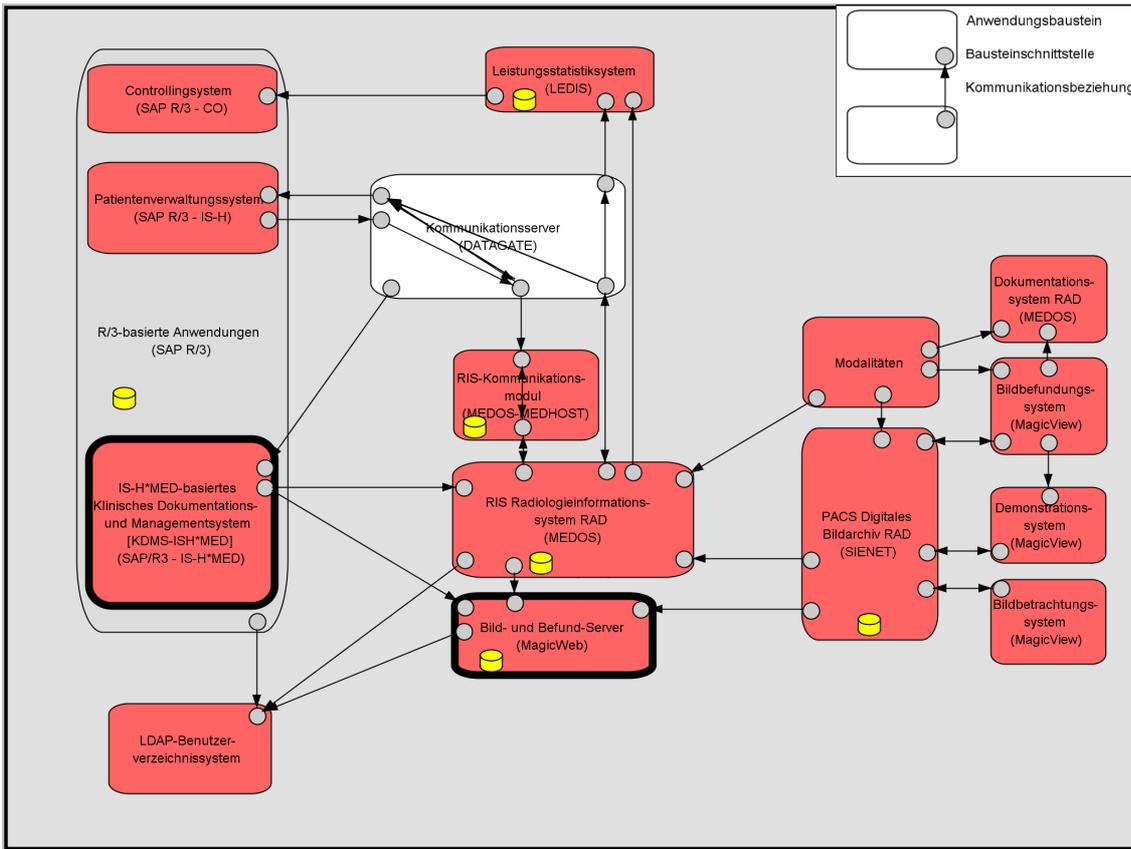


Abbildung 9.7: Präsentationsdomäne des physischen Datenverarbeitungsbausteines PC NCH 15

jekttypen weg, da sie unabhängig von den verarbeiteten Informationen hergestellt werden sollte bzw. muss. Gleiches gilt für den Bezug zu Ereignistypen. Die Menge der Anwendungsbaukästen, für die Präsentationsintegration herzustellen ist, wird also durch einen physischen Datenverarbeitungsbaustein bestimmt.

Damit können Präsentationsdomänen, ausgehend von physischen Datenverarbeitungsbausteinen, genauer definiert werden:

Definition 9.5 Die **Präsentationsdomäne** eines physischen Datenverarbeitungsbausteines $pdvb$ ist die Menge aller Anwendungsbaukästen awb_i , die gemeinsam auf $pdvb$ installiert sind:

$$PDom(pdvb) := \{awb_i \in AWE \mid ist_installiert_auf(awb, pdvb)\} \quad (9.5)$$

Werden mehrere physische Datenverarbeitungsbausteine gemeinsam genutzt, muss die Vereinigung der einzelnen Präsentationsdomänen gebildet werden.

Ende — Definition und Berechnungsvorschrift

Abbildung 9.7 zeigt ein Beispiel für eine Präsentationsdomäne. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen

Anwendungsbausteine gemeinsam auf dem physischen Datenverarbeitungsbaustein *PC NCH 15* installiert sind und gemeinsam genutzt werden.

9.3.6 Zugriffsdomänen

Eine Menge von **Anwendungsbausteinen**, für die die Forderung nach Zugriffsintegration besteht, wird hier als Zugriffsdomäne bezeichnet (vgl. Abschnitt 2.2.7). Zugriffsintegration muss, ähnlich der Präsentationsintegration, zwischen den **Anwendungsbausteinen** hergestellt werden, an denen dieselben Personen arbeiten. Wie bei Kontextdomänen und Präsentationsdomänen wird dieser Sachverhalt über den Bezug zu **physischen Datenverarbeitungsbausteinen** ausgedrückt; auch hier fällt, im Vergleich zu Kontextdomänen, der Bezug zu **Objekttypen** und **Ereignistypen** weg.

Zugriffsdomänen werden also wie Präsentationsdomänen definiert:

Definition 9.6 Die **Zugriffsdomäne** eines physischen Datenverarbeitungsbausteines *pdvb* ist die Menge aller **Anwendungsbausteine** awb_i , die gemeinsam auf *pdvb* installiert sind:

$$\underline{\text{ZDom}}(pdvb) := \{awb_i \in \underline{\text{AWB}} \mid \text{ist_installiert_auf}(awb, pdvb)\} \quad (9.6)$$

Werden mehrere **physische Datenverarbeitungsbausteine** durch dieselben Personen genutzt, muss die Vereinigung der einzelnen Zugriffsdomänen gebildet werden.

Ende — Definition und Berechnungsvorschrift

Alternative Modellierung von Zugriffsdomänen

Alternativ zur beschriebenen Berechnung von Zugriffsdomänen kann folgendes Verfahren angewendet werden: Personeninformationen können über spezielle **Objekttypen**, z. B. *Labormitarbeiter*, modelliert werden, die denjenigen **Aufgaben** zugeordnet sind, die von den betreffenden Personen bearbeitet werden. Dann können Zugriffsdomänen einfach als Datendomänen dieser **Objekttypen** modelliert werden.

9.4 Kommunikationsdomänen

Kommunikationsdomänen sind Mengen von **Anwendungsbausteinen**, bei denen durch Einrichtung von Kommunikationsverbindungen ein bestimmter Integrationsstatus hergestellt wurde. Es spielt dabei zunächst keine Rolle, mit welchen Integrationstechniken, d. h. über welche Arten von Middleware, die Integration hergestellt wurde. Die Tatsache, dass Integration hergestellt wurde, wird hier auf die Feststellung reduziert, dass durch Operationsaufrufe mit bestimmten Parametern Kommunikationspfade zwischen **Anwendungsbausteinen** existieren.

9.4.1 Domänen für die Übermittlung von Informationen

Übermittlungsdomänen für **Objekttypen** und **Begriffssysteme** enthalten **Anwendungsbausteine**, zu denen Informationen übermittelt werden.

Definition 9.7 Die **Übermittlungssdomäne** eines **Objekttyps** *objt*, eines **Ereignistyps** *ergt* und eines **Anwendungsbausteines** *awb* enthält die **Anwendungsbausteine**, zu denen *objt*

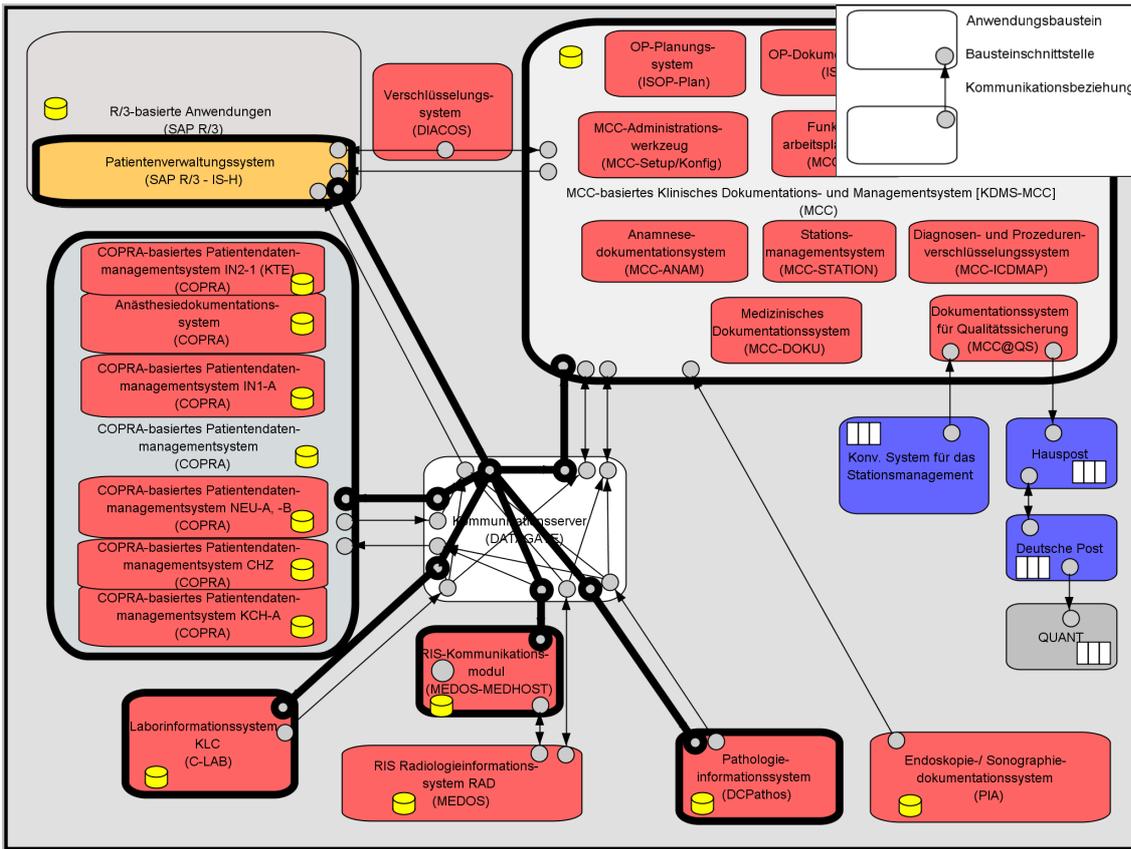


Abbildung 9.8: Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *NP1110* und des Anwendungsbausteines *Patientenverwaltungssystem*

bei Eintreten von *ergt* von *awb* aus übermittelt wird. Für den Austausch von Objekttypen ist es dabei erforderlich, dass die betreffenden Objekttypen in mindestens einem Anwendungsbaustein gespeichert sind, ihrem Master-Anwendungsbaustein.

$$\underline{\text{UebDom}}(\text{objt}, \text{ergt}, \text{awb}) := \left\{ \text{awb}_i \in \underline{\text{AWB}} \mid \begin{array}{l} \text{hat_als_Master}(\text{objt}, \text{awb}) \\ \text{wird_gespeichert_in}(\text{objt}, \text{awb}) \\ \text{wird_uebermittelt}(\text{objt}, \text{ergt}, \text{awb}, \text{awb}_i) \end{array} \wedge \right\} \quad (9.7)$$

Ende — Definition und Berechnungsvorschrift

Insbesondere bei der Betrachtung der Kommunikation wird die Bedeutung der transitiven Auswertung des Prädikates *wird_uebermittelt* deutlich (vgl. Abschnitt 8.5.3). Dadurch können Kommunikationsverbindungen, die aus mehreren einzelnen Kommunikationsbeziehungen zusammengesetzt sind, berücksichtigt werden (vgl. Abschnitt 8.6.1).

Abbildung 9.8 zeigt ein Beispiel für eine Übermittlungsdomäne eines Objekttyps. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass zu allen der hervorgehobenen Anwendungsbausteine, ausgehend vom *Patientenverwaltungssystem* beim Eintreten des Ereignistyps *NP1110*, Nachrichtentypen übermittelt werden, die den Objekttyp *Fall* repräsentieren.

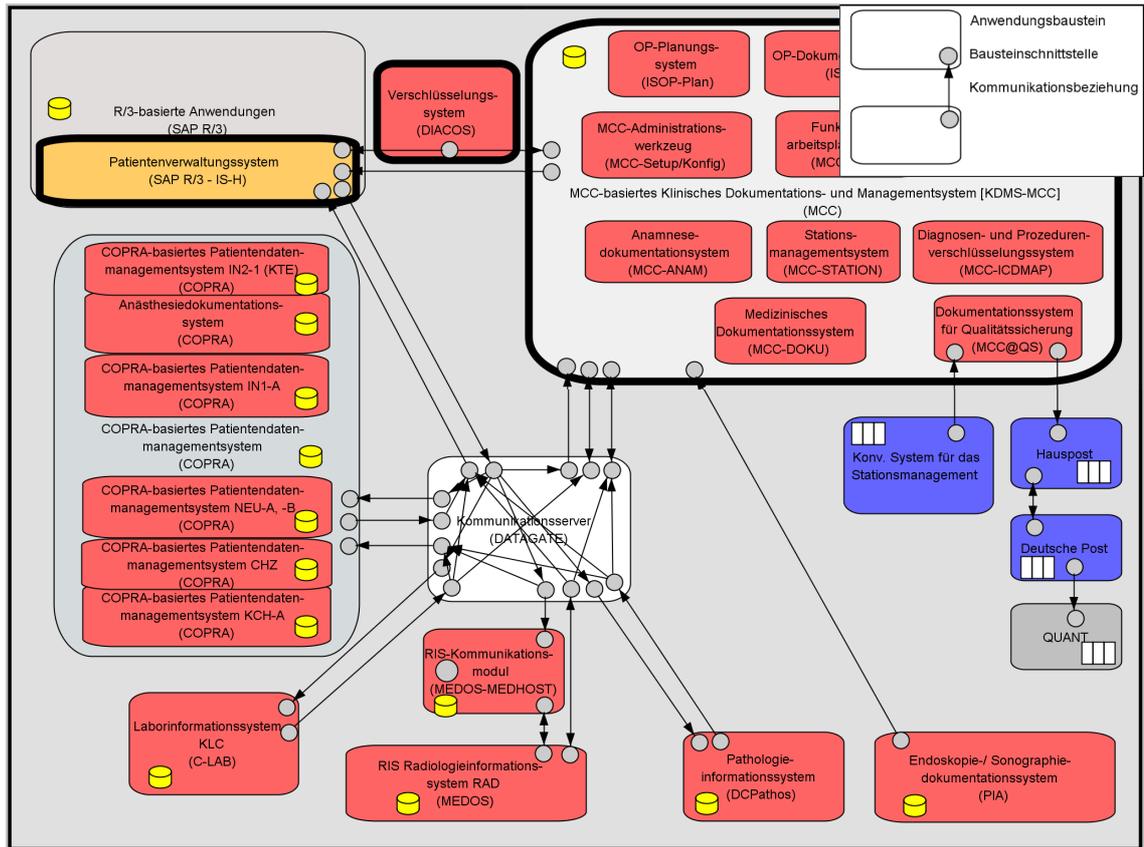


Abbildung 9.9: Aufrufdomäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung*, des Ereignistyps *DIA-PROZ_init* und des Anwendungsbausteines *Verschlüsselungssystem*

Auf der Basis der Ausführungen in Abschnitt 8.5.3 können, wie für Objekttypen, auch für Begriffssysteme Übermittlungsdomänen bestimmt werden:

Definition 9.8 Die **Übermittlungsdomäne** eines Begriffssystems bgs , eines Ereignistyps $ergt$ und eines Anwendungsbausteines awb enthält die Anwendungsbausteine, zu denen bgs bei Eintreten von $ergt$ von awb aus übermittelt wird.

$$\underline{\text{UebDomB}}(bgs, ergt, awb) := \left\{ awb_i \in \underline{\text{AWB}} \mid \begin{array}{l} \text{hat_als_Master}(bgs, awb) \\ \text{wird_gespeichert_in}(bgs, awb) \\ \text{wird_uebermittelt_b}(bgs, ergt, awb, awb_i) \end{array} \wedge \right\} \quad (9.8)$$

Ende — Definition und Berechnungsvorschrift

9.4.2 Domänen für den Aufruf von Funktionalität

Aufrufdomänen für Aufgaben enthalten Anwendungsbausteine, die Funktionalität zur Erfüllung bestimmter Aufgaben von anderen Anwendungsbausteinen aufrufen.

Definition 9.9 Die **Aufrufdomäne** einer Aufgabe auf , eines Ereignistyps $ergt$ und eines Anwendungsbausteines awb enthält die Anwendungsbausteine, die Funktionalität zur Erfüllung

Kasten 9.1: Ereignistypen und Kommunikationsdomänen

Durch die Berücksichtigung von **Ereignistypen** bei der Bestimmung von Kommunikationsdomänen kann analysiert werden, ob bestimmte Übermittlungen von **Objekttypen** oder bestimmte Aufrufe von **Aufgaben** bei allen oder nur einzelnen relevanten **Ereignistypen** stattfinden. Damit kann beispielsweise festgestellt werden, ob die Übermittlung von Falldaten beim Eintreten aller **Ereignistypen**, die bei einer Bearbeitung des **Objekttyps** *Fall* ausgelöst werden, erfolgt, oder nur beim Eintreten einzelner dieser **Ereignistypen**.

Zur vollständigen Realisierung von Datenintegration für einen bestimmten **Objekttyp** gehört u. a. auch die Übermittlung beim Eintreten aller relevanter **Ereignistypen**. Diese bedingung ist in der Praxis oft nicht erfüllt.

lung von *auf* bei Eintreten von *ergt* von *awb* aufrufen.

$$\underline{\text{AufDom}}(auf, ergt, awb) := \left\{ awb_i \in \underline{\text{AWB}} \mid \begin{array}{l} \text{unterstuetzt}(awb, auf) \\ \text{ruft_auf}(auf, ergt, awb_i, awb) \end{array} \wedge \right\} \quad (9.9)$$

Ende — Definition und Berechnungsvorschrift

Abbildung 9.9 zeigt ein Beispiel für eine Aufrufdomäne einer **Aufgabe**. Das Beispiel geht, auf der Basis der Szenariobeschreibung in Abschnitt 9.2, davon aus, dass alle der hervorgehobenen **Anwendungsbausteine** beim Eintreten des Ereignistyps *DIAPROZ_init* die Operation *DiaProz_Verschl* aufrufen, die das *Verschlüsselungssystem* über eine **Bausteinschnittstelle** bereitstellt.

9.5 Anwendung: Prüfung der Erfüllung von Integrationsanforderungen

Die in den vorhergehenden Abschnitten definierten Domänentypen können für eine Bewertung der Integration genutzt werden. Durch den Vergleich von Anforderungsdomänen mit Kommunikationsdomänen kann festgestellt werden, ob Integrationsanforderungen erfüllt sind.

9.5.1 Prüfung auf realisierte Datenintegration

Die Erfüllung der Forderung nach Datenintegration bestimmter **Anwendungsbausteine** erfolgt durch Vergleich der Datendomäne des interessierenden **Objekttyps** mit den Übermittlungsdomänen dieses **Objekttyps**.

Die Forderung nach Datenintegration

- bzgl. eines **Objekttyps** *objt*

ist genau dann erfüllt, wenn

- für alle **Ereignistypen** *ergt_i*, die zu einer Bearbeitung von *objt* führen, und
- für alle **Master-Anwendungsbausteine** *awb_j* von *objt*

die Datendomäne von *objt* Teilmenge jeder Übermittlungsdomäne von *objt* ist. Der Ausdruck

$$\forall ergt_i \in \underline{\text{ERGT}}^*, awb_j \in \underline{\text{AWB}}^* : \underline{\text{DDom}}(objt, ergt_i) \subseteq \underline{\text{UebDom}}(objt, ergt_i, awb_j) \quad (9.10)$$

$\underline{\text{ERGT}}^*$ = Menge der *ergt_i*, die zu einer Bearbeitung von *objt* führen
 $\underline{\text{AWB}}^*$ = Menge der *awb_j*, die **Master-Anwendungsbaustein** für *objt* sind

muss also für den betreffenden **Objekttyp** wahr sein.

9 Die Erfüllung von Integrationsanforderungen

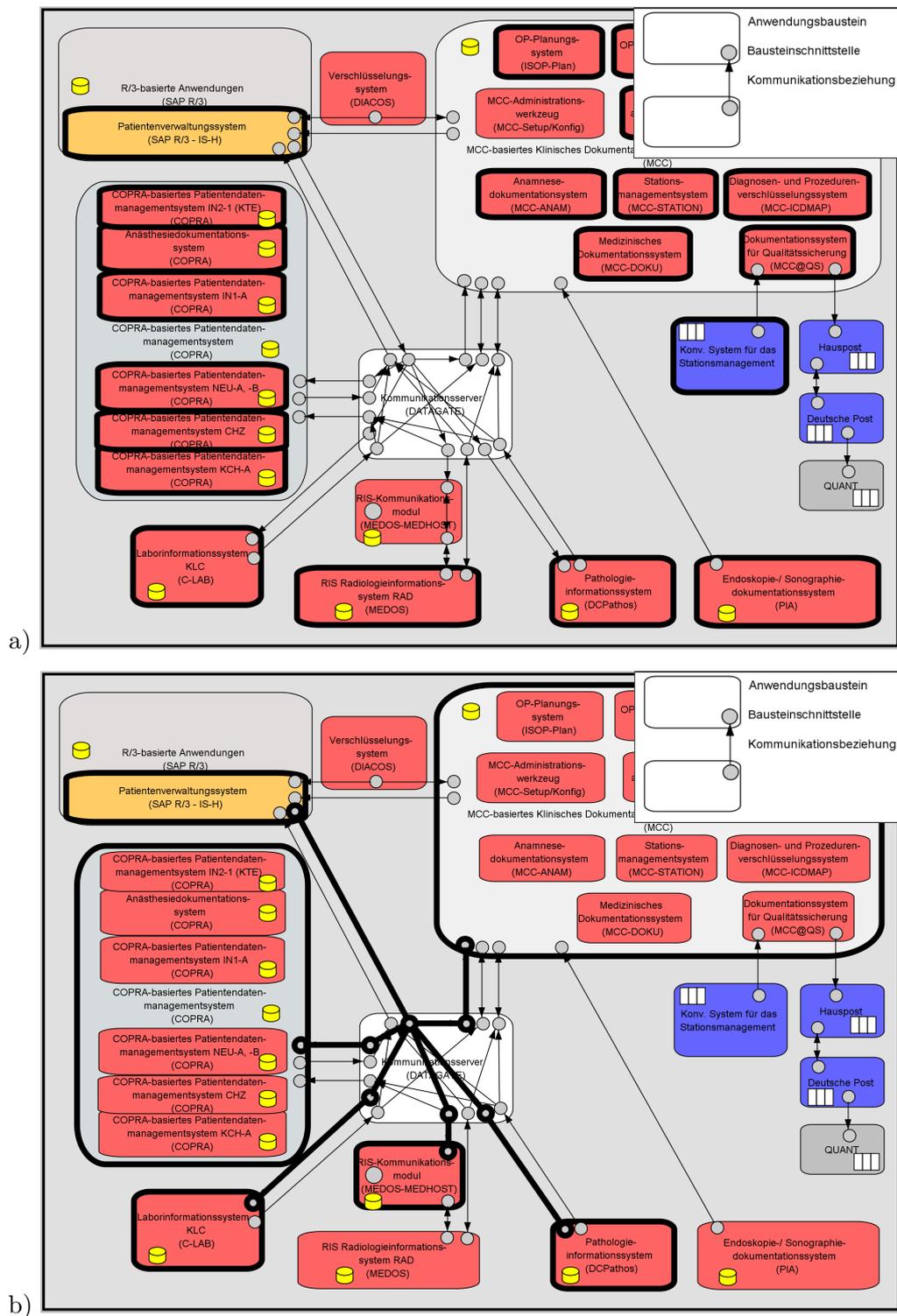


Abbildung 9.10: Vergleich der Datendomäne des Objekttyps *Fall* und des Ereignistyps *NP1110* (a) mit der Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *NP1110* und des Anwendungsbausteines *Patientenverwaltungssystem* (b)

Ein Vergleich der Datendomäne des Objekttyps *Fall* und des Ereignistyps *NP11I0* in der Beispielabbildung 9.3 mit der Übermittlungsdomäne desselben Objekttyps und desselben Ereignistyps sowie des Anwendungsbausteines *Patientenverwaltungssystem* in der Beispielabbildung 9.8 zeigt, dass die in der Datendomäne enthaltenen Anwendungsbausteine *Radiologieinformationssystem RAD* und *Endoskopie-/Sonographiedokumentationssystem* nicht in der Übermittlungsdomäne enthalten sind (Abbildung 9.10). In diesem Beispiel ist die Forderung nach Datenintegration hinsichtlich des Objekttyps *Fall* zunächst nicht vollständig erfüllt. Es sei gegeben, dass für den Objekttyp *Fall*, den Ereignistyp *RIS_Fall_Abfrage* und den Anwendungsbaustein *RIS Kommunikationsmodul* eine weitere Übermittlungsdomäne existiert, die das Radiologieinformationssystem RAD enthält. Damit wird die fehlende unmittelbare Mitgliedschaft in der ursprünglichen Übermittlungsdomäne ausgeglichen. Auch für das Endoskopie-/Sonographiedokumentationssystem sei hier die Existenz einer zusätzlichen Übermittlungsdomäne mit dem MCC-basierten Klinischen Dokumentations- und Managementsystem angenommen.

9.5.2 Prüfung auf realisierte funktionale Integration

Die Erfüllung der Forderung nach funktionaler Integration erfolgt durch Vergleich der funktionalen Domäne der interessierenden Aufgabe mit den Aufrufdomänen dieser Aufgabe.

Die Forderung nach funktionaler Integration

- bzgl. einer Aufgabe *auf*

ist genau dann erfüllt, wenn

- für alle Ereignistypen $ergt_i$, die zu einer Aktivität von *auf* führen, und
- für alle Master-Anwendungsbausteine awb_j von *auf*

die funktionale Domäne von *auf* Teilmenge jeder Aufrufdomäne von *auf* ist. Der Ausdruck

$$\forall ergt_i \in \underline{ERGT}^*, awb_j \in \underline{AWB}^* : \underline{FDom}(auf) \subseteq \underline{AufDom}(auf, ergt_i, awb_j) \quad (9.11)$$

\underline{ERGT}^* = Menge der $ergt_i$, die zu einer Aktivität von *auf* führen
 \underline{AWB}^* = Menge der awb_j , die Master-Anwendungsbaustein für *auf* sind

muss also für die betreffende Aufgabe wahr sein.

Bei der Prüfung wird auch die Möglichkeit berücksichtigt, dass mehrere Master-Anwendungsbausteine für die betrachtete Aufgabe existieren. Diese Möglichkeit steht prinzipiell im Widerspruch zur Idee der funktionalen Integration. Sie wird hier trotzdem eingeräumt, da die praktische Umsetzung u. U. eine zentrale aber trotzdem redundante Bereitstellung einer bestimmten Funktionalität erfordern kann.

Ein Vergleich der funktionalen Domäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung* mit der Aufrufdomäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung*, des Ereignistyps *DIAPROZ_init* und des Anwendungsbausteines *Verschlüsselungssystem* zeigt, dass nur einige Anwendungsbausteine der funktionalen Domäne in der Aufrufdomäne enthalten sind (Abbildung 9.11). In diesem Beispiel ist die Forderung nach funktionaler Integration hinsichtlich der Aufgabe *Diagnosen- und Prozedurenverschlüsselung* nicht bzw. nur teilweise erfüllt.

9 Die Erfüllung von Integrationsanforderungen

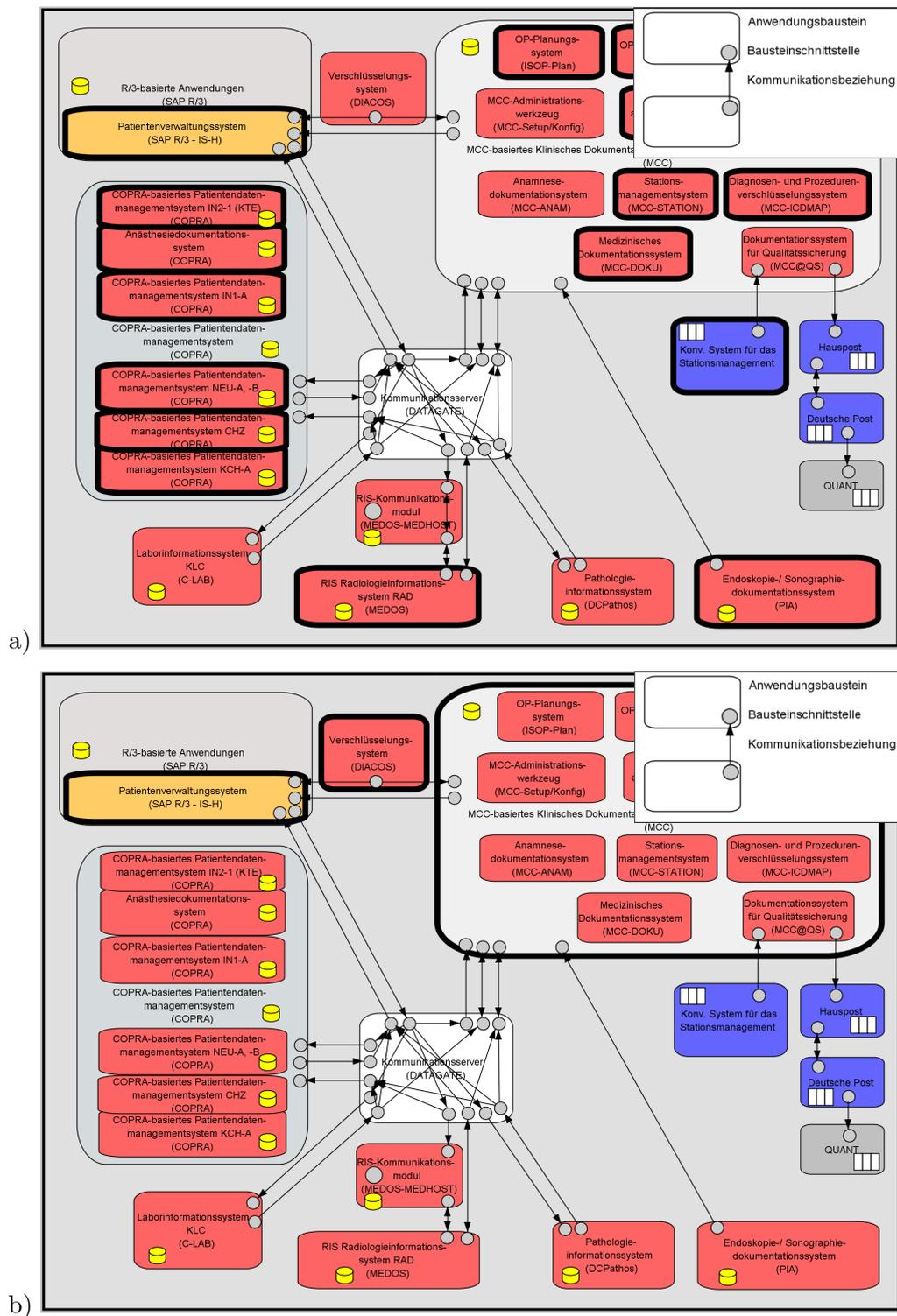


Abbildung 9.11: Vergleich der funktionalen Domäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung* (a) mit der Aufrufdomäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung*, des Ereignistyps *DIAPROZ_init* und des Anwendungsbausteines *Verschlüsselungssystem* (b)

9.5.3 Prüfung auf realisierte semantische Integration

Semantische Integration kann, abhängig davon, ob **Begriffssysteme** autonom oder zentral verwaltet werden, über Datenintegration oder über funktionale Integration realisiert werden.

Semantische Integration bei autonomer Begriffssystemverwaltung

Speichern die **Anwendungsbausteine** einer semantischen Domäne die verwendeten **Begriffssysteme** selbst, dann muss zwischen diesen **Anwendungsbausteinen** Datenintegration hinsichtlich der **Begriffssysteme** realisiert werden. Durch Vergleich der interessierenden semantischen Domänen mit den Übermittlungsdomänen der betreffenden **Begriffssysteme** kann dann eine Prüfung auf Realisierung der semantischen Integration vorgenommen werden.

Bei autonomer Begriffssystemverwaltung ist die Forderung nach semantischer Integration

- bzgl. eines **Begriffssystems** bgs

genau dann erfüllt, wenn

- für alle **Ereignistypen** $ergt_i$, die zu einer Aktualisierung von bgs führen, und
- für alle **Master-Anwendungsbausteinen** awb_j von bgs

die semantische Domäne von bgs Teilmenge jeder Übermittlungsdomäne von bgs , $ergt_i$ und awb_j ist. Der Ausdruck

$$\forall ergt_i \in \underline{ERGT}^*, awb_j \in \underline{AWB}^* : \underline{SDom}(bgs) \subseteq \underline{UebDomB}(bgs, ergt_i, awb_j) \quad (9.12)$$

\underline{ERGT}^* = Menge der $ergt_i$, die zu einer Aktualisierung von bgs führen
 \underline{AWB}^* = Menge der awb_j , die **Master-Anwendungsbaustein** für bgs sind

muss also für das betreffende **Begriffssystem** wahr sein.

Ein Vergleich der semantischen Domäne des **Begriffssystems** $ICD10$ mit der Übermittlungsdomäne desselben **Begriffssystems**, des **Ereignistyps** ICD_neu und seines **Master-Anwendungsbausteines** $Verschlüsselungssystem$ zeigt, dass $ICD10$ zu keinem der **Anwendungsbausteine** kommuniziert wird, die es benötigen (Abbildung 9.12). In diesem Beispiel ist die Forderung nach semantischer Integration NICHT erfüllt.

Semantische Integration bei zentraler Begriffssystemverwaltung

Wird ein zentraler **Anwendungsbaustein** für die Bereitstellung bestimmter **Begriffssysteme** genutzt, dann muss funktionale Integration hinsichtlich der **Aufgabe**, die die betreffenden **Begriffssysteme** nutzt, hergestellt werden. Durch Vergleich der interessierenden semantischen Domänen mit den Aufrufdomänen der betreffenden **Aufgabe** kann eine Prüfung auf Realisierung der semantischen Integration vorgenommen werden.

Bei zentraler Begriffssystemverwaltung ist die Forderung nach semantischer Integration

- bzgl. einer **Aufgabe** auf , die für die Aufgabe des Begriffssystemzugriffs steht, und
- bzgl. eines **Begriffssystems** bgs

genau dann erfüllt, wenn

- für alle **Ereignistypen** $ergt_i$, die zu einer Aktivität von auf führen, und
- für alle **Master-Anwendungsbausteine** awb_j von auf

9 Die Erfüllung von Integrationsanforderungen

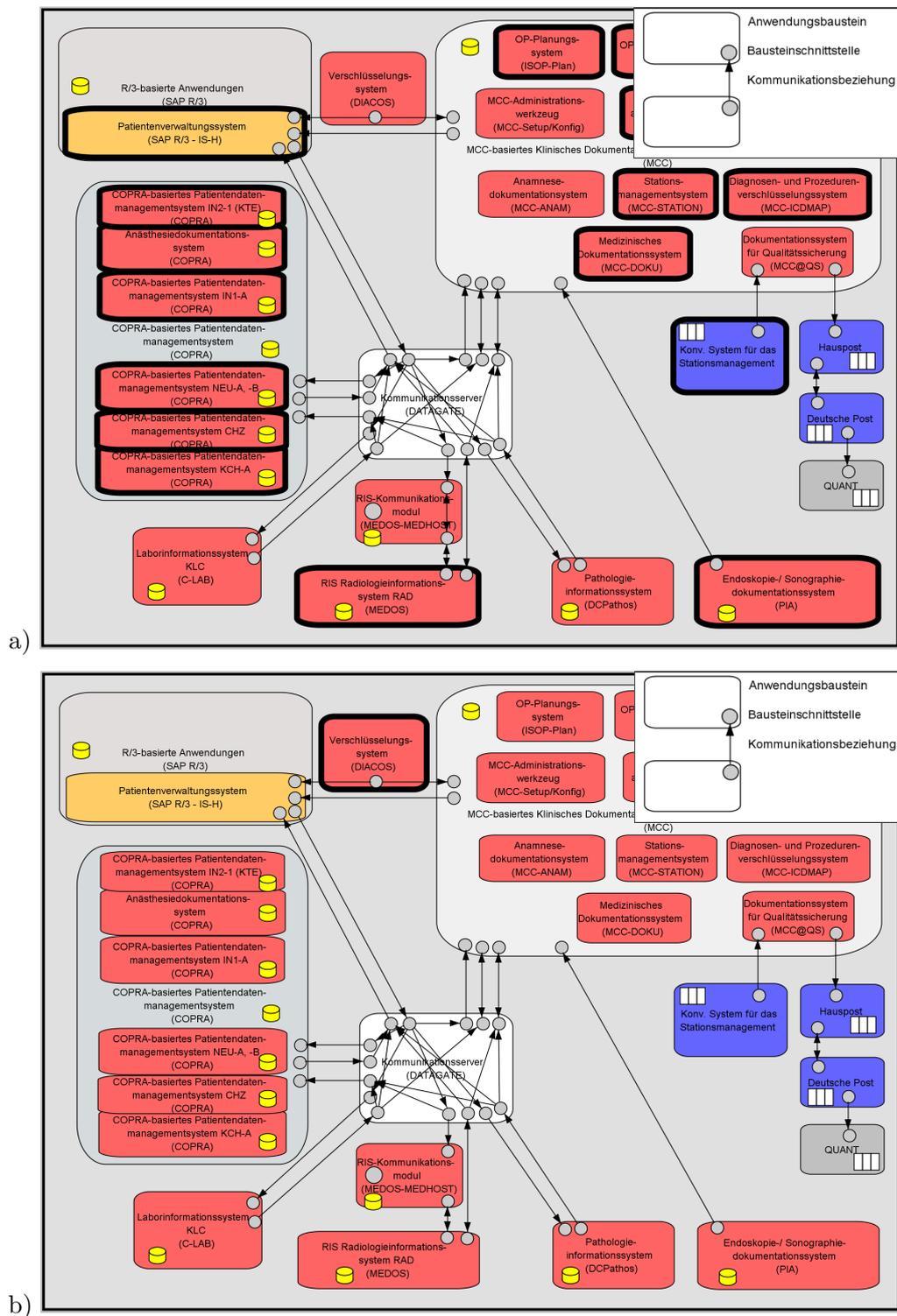


Abbildung 9.12: Vergleich der semantischen Domäne des Begriffssystems *ICD10* (a) mit der Übermittlungsdomäne desselben Begriffssystems, des Ereignistyps *ICD_neu* und des Anwendungsbausteines *Verschlüsselungssystem* (b)

9.5 Anwendung: Prüfung der Erfüllung von Integrationsanforderungen

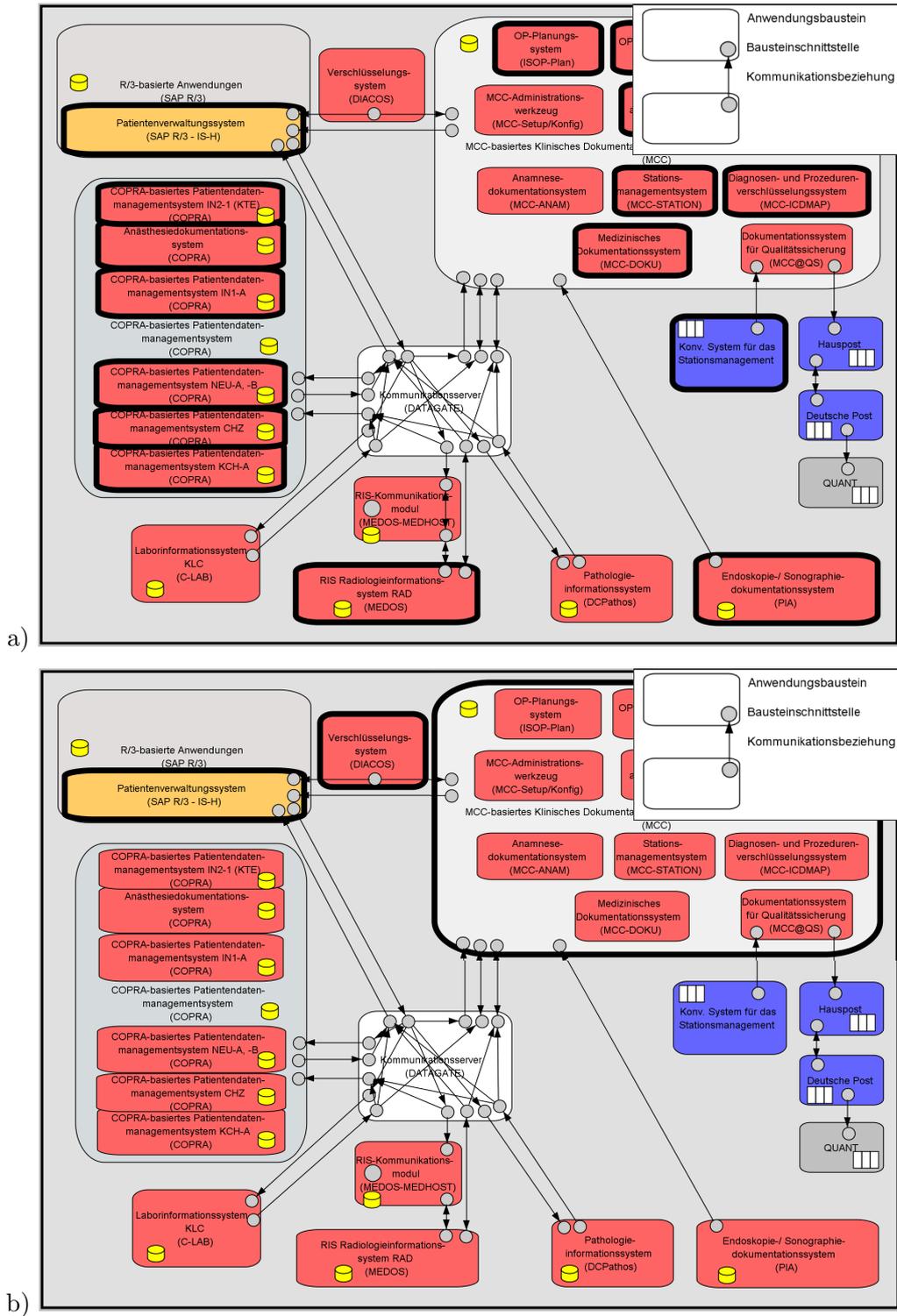


Abbildung 9.13: Vergleich der semantischen Domäne des Begriffssystems *ICD10* (a) mit der Aufrufdomäne der Aufgabe *Diagnosen- und Prozedurenverschlüsselung*, des Ereignistyps *DIAPROZ_init* und des Anwendungsbausteines *Verschlüsselungssystem* (b)

9 Die Erfüllung von Integrationsanforderungen

die semantische Domäne von bgs Teilmenge jeder Aufrufdomäne von auf ist. Der Ausdruck

$$\forall \text{erg}_i \in \underline{\text{ERGT}}^*, \text{awb}_j \in \underline{\text{AWB}}^* : \underline{\text{SDom}}(bgs) \subseteq \underline{\text{AufDom}}(auf, \text{erg}_i, \text{awb}_j) \quad (9.13)$$

$\underline{\text{ERGT}}^*$ = Menge der erg_i , die zu einer Aktivität von auf führen
 $\underline{\text{AWB}}^*$ = Menge der awb_j , die **Master-Anwendungsbaustein** für auf sind

muss also für die betreffende **Aufgabe** und für das betreffende **Begriffssystem** wahr sein.

Ein Vergleich der semantischen Domäne des **Begriffssystems** *ICD10* mit der Aufrufdomäne der **Aufgabe** *Diagnosen- und Prozedurenverschlüsselung*, des **Ereignistyps** *DIAPROZ_init* und des **Anwendungsbausteines** *Verschlüsselungssystem* zeigt, dass einige, aber nicht alle, **Anwendungsbausteine** der semantischen Domäne auch Elemente der Aufrufdomäne sind (Abbildung 9.13). Die Forderung nach semantischer Integration ist in diesem Beispiel nicht bzw. nur teilweise erfüllt. Im Vergleich zum vorhergehenden Beispiel zur autonomen Begriffssystemverwaltung ist jedoch „mehr“ semantische Integration realisiert.

9.5.4 Prüfung auf realisierte Kontextintegration

Die Erfüllung der Forderung nach Kontextintegration erfolgt durch Vergleich der Kontextdomäne des interessierenden **Objekttyps**, des interessierenden **physischen Datenverarbeitungsbausteines** und der **Ereignistypen**, die eine Kontextsynchronisierung erfordern, mit den Übermittlungsdomänen des **Objekttyps** sowie eines speziellen **Objekttyps** für Informationen zur Identifikation von **physischen Datenverarbeitungsbausteinen** und der genannten **Ereignistypen**. Der spezielle **Objekttyp** ist zusätzlich zu modellieren und wird benötigt, da auch Daten zum betroffenen **physischen Datenverarbeitungsbaustein** übermittelt werden müssen.

Die Forderung nach Kontextintegration

- bzgl. eines **Objekttyps** $objt$ und
- bzgl. eines **physischen Datenverarbeitungsbausteines** $pdvb$

ist genau dann erfüllt, wenn

- für alle **Ereignistypen** erg_i , die eine Kontextsynchronisierung erfordern, und
- für alle **Anwendungsbausteine** awb_j , die einen Kontextwechsel auslösen,

die Kontextdomäne von $objt$, $pdvb$ Teilmenge jeder Übermittlungsdomäne von $objt$ und dem speziellen **Objekttyp** $objt^*$ ist. Der Ausdruck

$$\forall \text{erg}_i \in \underline{\text{ERGT}}^*, \text{awb}_j \in \underline{\text{AWB}}^* : \underline{\text{KDom}}(objt, pdvb, \text{erg}_i) \subseteq \begin{array}{c} \underline{\text{UebDom}}(objt, \text{erg}_i, \text{awb}_j) \\ \cup \\ \underline{\text{UebDom}}(objt^*, \text{erg}_i, \text{awb}_j) \end{array} \quad (9.14)$$

$\underline{\text{ERGT}}^*$ = Menge der erg_i , die einen Kontextwechsel bzgl. $objt$ erfordern
 $\underline{\text{AWB}}^*$ = Menge der awb_j , die einen Kontextwechsel bzgl. $objt$ auslösen
 \underline{objt}^* = **Objekttyp** für Informationen zur Identifikation von **physischen Datenverarbeitungsbausteinen**

muss also für den betreffenden **Objekttyp** und den betreffenden **physischen Datenverarbeitungsbaustein** wahr sein.

Ein Vergleich der Kontextdomäne des **Objekttyps** *Fall* und des **Ereignistyps** *Kontextwechsel_Fall* mit der Übermittlungsdomäne desselben **Objekttyps** und desselben **Ereignistyps** sowie des **Anwendungsbausteines** *KDMS-IS-H*MED* zeigt, dass der in der Kontextdomäne

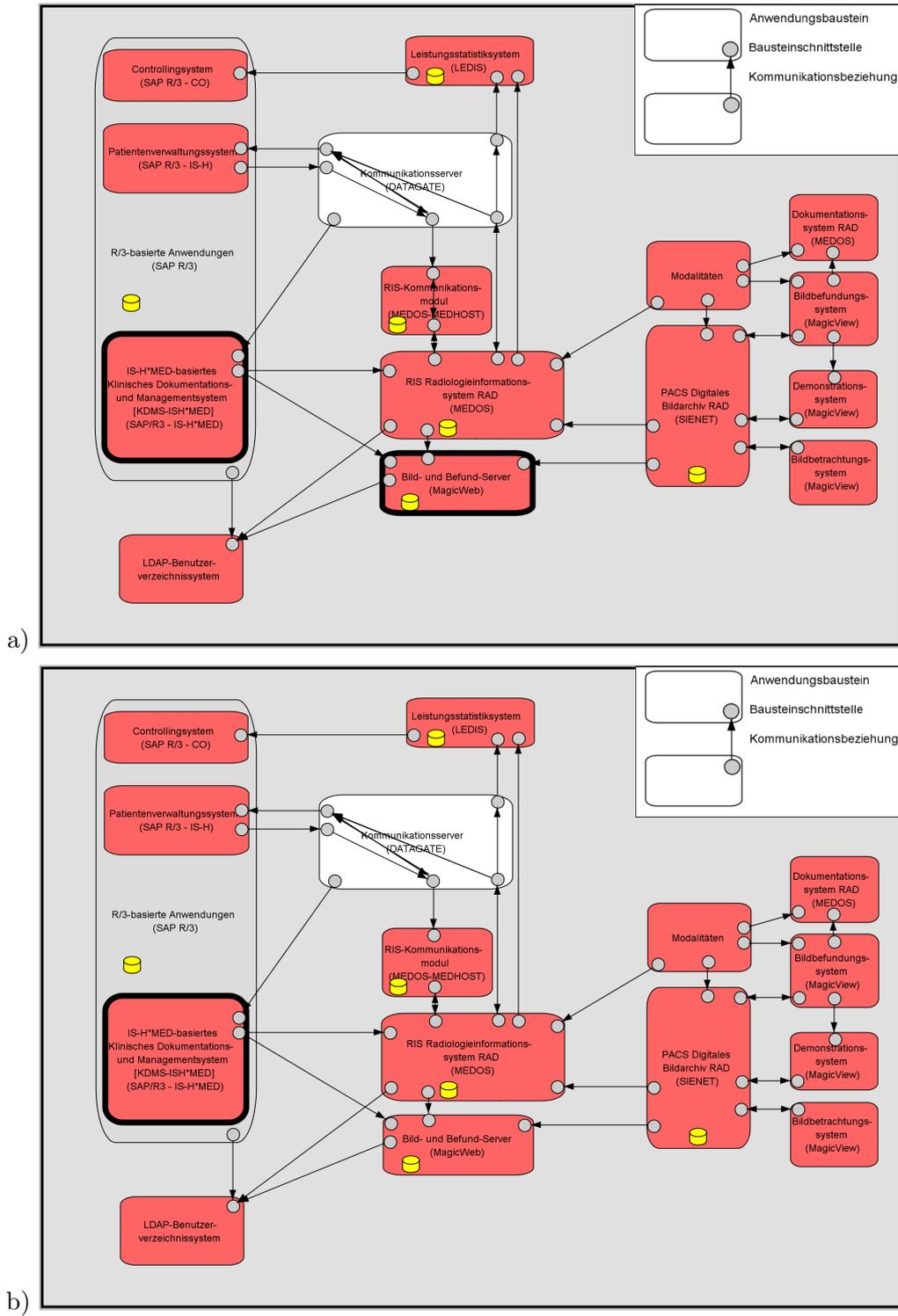


Abbildung 9.14: Vergleich der Kontextdomäne des Objekttyps *Fall*, des physischen Datenverarbeitungsbausteines *PC NCH 15* und des Ereignistyps *Kontextwechsel_Fall* (a) mit der Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *Kontextwechsel_Fall* und des Anwendungsbausteines *KDMS-IS-H*MED* (b)

enthaltene Anwendungsbaustein *Bild- und Befundserver* nicht in der Übermittlungsdomäne enthalten ist (Abbildung 9.14). In diesem Beispiel ist die Forderung nach Kontextintegration hinsichtlich des Objekttyps *Fall* NICHT erfüllt.

9.5.5 Prüfung auf realisierte Präsentationsintegration

Die Prüfung auf Präsentationsintegration kann, im Gegensatz zu den anderen Integrationsprüfungen, nicht unmittelbar durch Vergleich von Anforderungsdomänen mit Kommunikationsdomänen vorgenommen werden. Das ist insofern folgerichtig, als die gleiche Gestaltung von Benutzungsschnittstellen unterschiedlicher Anwendungsbausteine nicht unmittelbar mit der Kommunikation von Daten oder Funktionalität zusammenhängt.

Aus dem Vorliegen von funktionaler Integration bezüglich bestimmter Aufgaben kann u. U. das Vorliegen von Präsentationsintegration abgeleitet werden. Das gilt, wenn die betreffenden Aufgaben mit Dateneingabe durch einen Benutzer oder mit Datenausgabe verbunden sind. Wenn mehrere genutzte Anwendungsbausteine denselben Teil-Anwendungsbaustein zur Erfüllung einer Teil-Aufgabe nutzen, dann liegt zumindest für die Erledigung dieser Teil-Aufgabe an dem betreffenden physischen Datenverarbeitungsbaustein Präsentationsintegration vor.

9.5.6 Prüfung auf realisierte Zugriffsintegration

Zugriffsintegration kann, abhängig davon, ob Zugriffsinformationen autonom oder zentral verwaltet werden, über Datenintegration oder über funktionale Integration realisiert werden.

Zugriffsintegration bei autonomer Zugriffsverwaltung

Führen die Anwendungsbausteine einer Zugriffsdomäne jeweils eigene Datenbanksysteme für die Zugriffsverwaltung, dann muss zwischen diesen Anwendungsbausteinen Datenintegration hinsichtlich der Zugriffsinformationen realisiert werden. Die Zugriffsinformationen müssen in diesem Fall über spezielle Objekttypen für bestimmte Personengruppen modelliert werden, z. B. *Labormitarbeiter* oder *Mitarbeiter des Pflegedienstes*. Durch Vergleich der interessierenden Zugriffsdomänen mit den Übermittlungsdomänen der betreffenden Objekttypen kann dann eine Prüfung auf Realisierung der Zugriffsintegration vorgenommen werden.

Bei autonomer Zugriffsverwaltung ist die Forderung nach Zugriffsintegration

- bzgl. eines Objekttyps *objt*, der für Zugriffsinformationen einer bestimmten Personengruppe steht, und
- bzgl. eines physischen Datenverarbeitungsbausteines *pdrv*, der durch die betreffende Personengruppe benutzt wird

genau dann erfüllt, wenn

- für alle Ereignistypen *ergt_i*, die zu einer Bearbeitung von *objt* führen, und
- für alle Master-Anwendungsbausteine *awb_j* von *objt*

die Zugriffsdomäne von *pdrv* Teilmenge jeder Übermittlungsdomäne von *objt* ist. Der Ausdruck

9.5 Anwendung: Prüfung der Erfüllung von Integrationsanforderungen

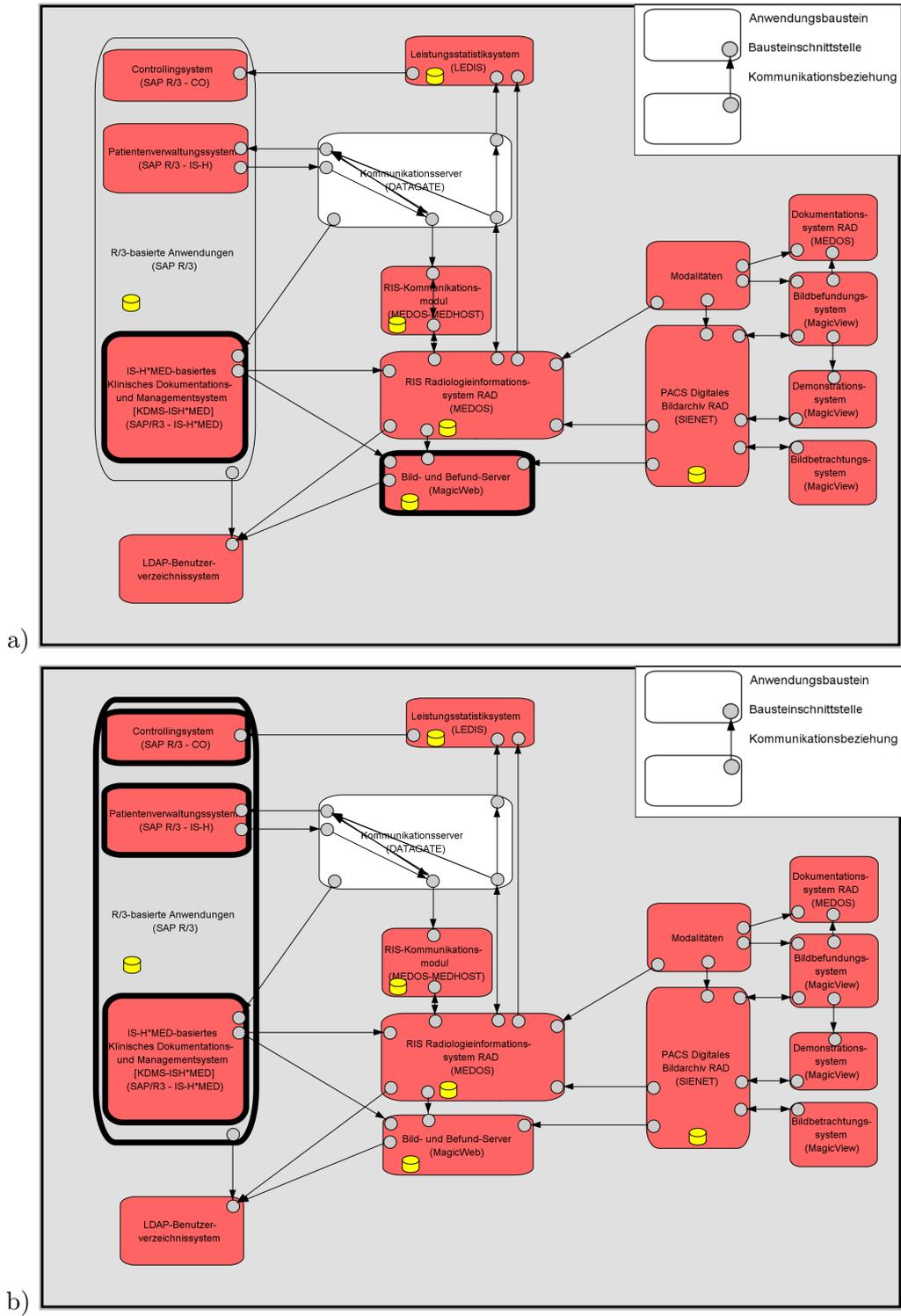


Abbildung 9.15: Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines *PC NCH 15* (a) mit der Übermittlungsdomäne des Objekttyps *Mitarbeiter des Pflegedienstes*, des Ereignistyps *ZUGR_update* und des Anwendungsbausteines *R/3-basierte Anwendungen* (b)

$$\forall \text{ergt}_i \in \underline{\text{ERGT}}^*, \text{awb}_j \in \underline{\text{AWB}}^* : \underline{\text{ZDom}}(\text{pdvb}) \subseteq \underline{\text{UebDom}}(\text{objt}, \text{ergt}_i, \text{awb}_j) \quad (9.15)$$

$\underline{\text{ERGT}}^*$ = Menge der ergt_i , die zu einer Bearbeitung von objt führen
 $\underline{\text{AWB}}^*$ = Menge der awb_j , die Master-Anwendungsbaustein für objt sind

muss also für den betreffenden Objekttyp und den betreffenden physischen Datenverarbeitungsbaustein wahr sein.

Ein Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines *PC NCH 15* mit der Übermittlungsdomäne des Objekttyps *Mitarbeiter des Pflegedienstes*, des Ereignistyps *ZUGR_update* und des Anwendungsbausteines *R/3-basierte Anwendungen*¹ zeigt, dass der Anwendungsbaustein *Bild- und Befundserver* nicht in der Übermittlungsdomäne enthalten ist (Abbildung 9.15). Die Forderung nach Zugriffsintegration ist in diesem Beispiel also NICHT erfüllt.

Zugriffsintegration bei zentraler Zugriffsverwaltung

Wird in einem zentralen Anwendungsbaustein ein zentrales Datenbanksystem für die Zugriffsverwaltung geführt, dann muss funktionale Integration hinsichtlich der Abwicklung von Zugriffskontrollaktivitäten realisiert werden. In diesem Fall muss eine spezielle Aufgabe für die Zugriffskontrolle modelliert werden, z. B. *Rechteprüfung MRZ*. Durch Vergleich der interessierenden Zugriffsdomänen mit den Aufrufdomänen der betreffenden Aufgabe kann dann eine Prüfung auf Realisierung der Zugriffsintegration vorgenommen werden.

Bei zentraler Zugriffsverwaltung ist die Forderung nach Zugriffsintegration

- bzgl. einer Aufgabe *auf*, die für die Aufgabe der zentrale Zugriffsverwaltung steht, und
- bzgl. eines physischen Datenverarbeitungsbausteines *pdvb*, der durch eine bestimmte Personengruppe benutzt wird

genau dann erfüllt, wenn

- für alle Ereignistypen ergt_i , die zu einer Aktivität von *auf* führen, und
- für alle Master-Anwendungsbausteine awb_j von *auf*

die Zugriffsdomäne von *pdvb* Teilmenge jeder Aufrufdomäne von *auf* ist. Der Ausdruck

$$\forall \text{ergt}_i \in \underline{\text{ERGT}}^*, \text{awb}_j \in \underline{\text{AWB}}^* : \underline{\text{ZDom}}(\text{pdvb}) \subseteq \underline{\text{AufDom}}(\text{auf}, \text{ergt}_i, \text{awb}_j) \quad (9.16)$$

$\underline{\text{ERGT}}^*$ = Menge der ergt_i , die zu einer Aktivität von *auf* führen
 $\underline{\text{AWB}}^*$ = Menge der awb_j , die Master-Anwendungsbaustein für *auf* sind

muss also für die betreffende Aufgabe und den betreffenden physischen Datenverarbeitungsbaustein wahr sein.

Ein Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines *PC NCH 15* mit der Aufrufdomäne der Aufgabe *Rechteprüfung MRZ*, des Ereignistyps *LOGIN_BBS* und des Anwendungsbausteines *LDAP-Benutzerverzeichnissystem* zeigt, dass alle Anwendungsbausteine der Zugriffsdomäne auch Elemente der Aufrufdomäne sind (Abbildung 9.16). Die Forderung nach Zugriffsintegration ist in diesem Beispiel erfüllt. Der Mangel an Zugriffsintegration über Datenintegration aus dem Beispiel zur autonomen Zugriffsverwaltung

¹ Dieser Anwendungsbaustein ist übergeordneter Anwendungsbaustein zum in Abschnitt 9.2 genannten KDMS-ISH*MED.

9.5 Anwendung: Prüfung der Erfüllung von Integrationsanforderungen

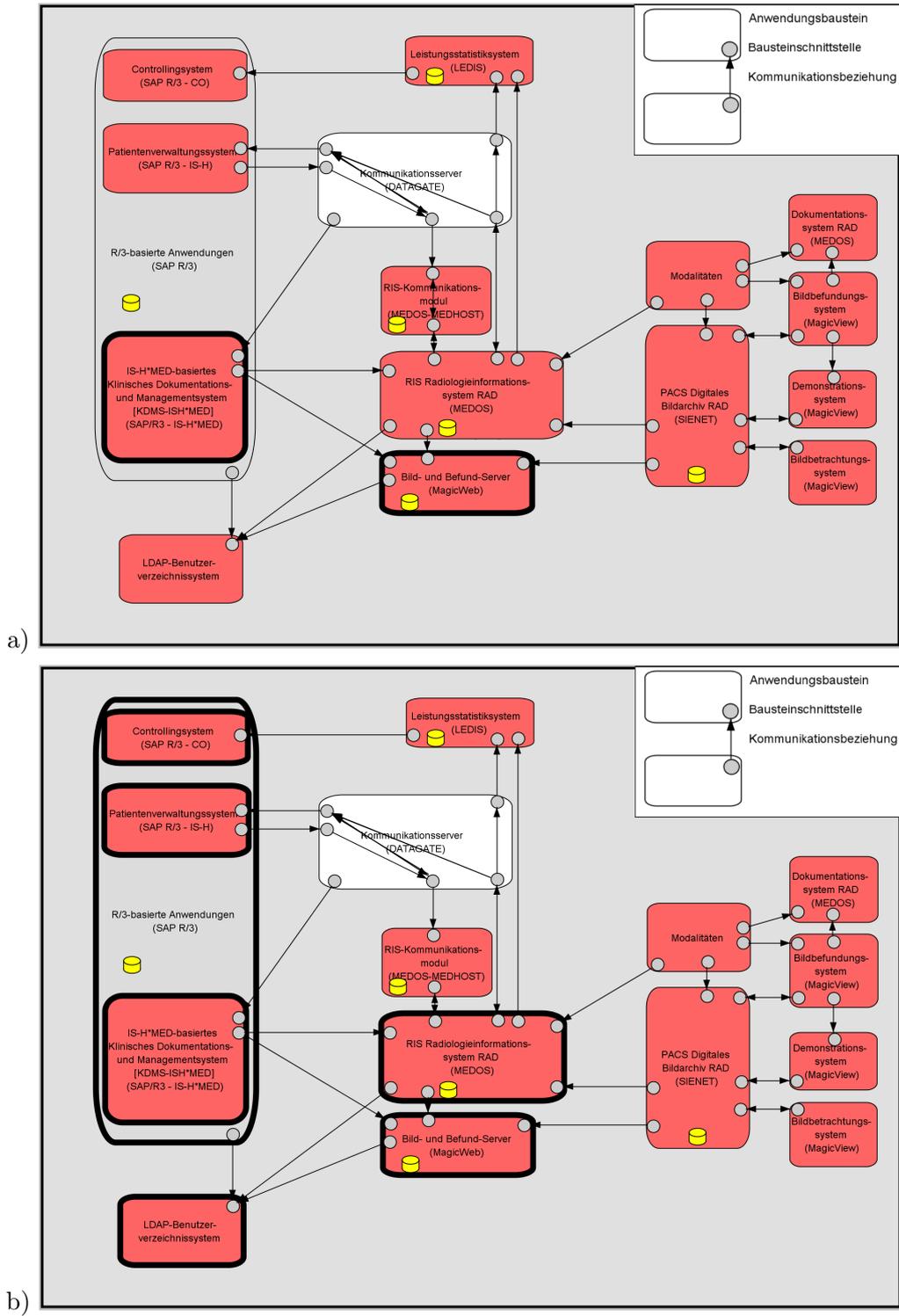


Abbildung 9.16: Vergleich der Zugriffsdomäne des physischen Datenverarbeitungsbausteines *PC NCH 15* (a) mit der Aufrufdomäne der Aufgabe *Rechteprüfung MRZ*, des Ereignistyps *LOGIN_BBS* und des Anwendungsbausteines *LDAP-Benutzerverzeichnisystem* (b)

9 Die Erfüllung von Integrationsanforderungen

ist also hier durch zentrale Zugriffsverwaltung mit funktionaler Integration aufgehoben.

10 Abhängigkeit von Anwendungsbausteinen

10.1 Vorbemerkungen

Im Zusammenhang mit Datenbanken, insbesondere mit verteilten Datenbanken, wird das in der Informatik sehr bekannte Transaktionskonzept ausführlich untersucht (siehe z. B. [VOSSEN 2000], 521-549, [RAHM 1994], S. 111-131). Die damit verbundenen mächtigen Theorien zur Abwicklung und Absicherung von Transaktionen sollen hier nicht diskutiert werden, jedoch können einige der damit verbundenen Begriffe helfen, die hier vorgestellten Bewertungsansätze einzuordnen und zu präzisieren.

Anwendungsbausteine, die Daten untereinander austauschen, können mit verteilten Datenbanken verglichen werden. Bezüglich der Erfüllung von Integrationsanforderungen sind dabei vor allem solche Verteilungen von Interesse, bei denen die beteiligten **Anwendungsbausteine** keine gemeinsamen **Datenbanksysteme** haben. Sie lassen sich nach den Einteilungen in [RAHM 1994] oft mit verteilten Transaktionssystemen oder mit föderativen **Datenbanksystemen** vergleichen (Abbildung 10.1):

„In Mehrrechner-Datenbanksystemen kooperieren mehrere DBVS (bzw. DBVS-Prozesse) zur Bearbeitung von DB-Operationen. Eine verteilte Transaktionsverarbeitung kann jedoch auch außerhalb der DBVS erreicht werden, i.a. unter Kontrolle von TP-Monitoren (...). Wir sprechen in diesem Fall von *Verteilten Transaktionssystemen*.“ ([RAHM 1994], Kapitel 11)

„Föderative Mehrrechner-DBS (federated database systems) streben nach größerer Knotenautonomie im Vergleich zu den integrierten Systemen, wobei die beteiligten DBVS entweder homogen oder heterogen sein können. (...) Kennzeichnende Eigenschaft der föderativen DBS ist, daß jeder Rechner eine eigene Datenbank verwaltet, die durch ein lokales (privates) konzeptionelles Schema beschrieben ist. Durch eine begrenzte Kooperation der DBVS soll es möglich werden, auf bestimmte Daten anderer Rechner zuzugreifen, falls dies von dem „Besitzer“ der Daten zugelassen wird. (...) Idealerweise unterstützen föderative Mehrrechner-DBS trotz Unterschieden bei den lokalen DBVS ein einheitliches Datenmodell bzw. bieten eine gemeinsame Anfragesprache „nach außen“ an. (...)“ ([RAHM 1994], Abschnitt 3.2)

Die Verteilung auf unterschiedliche **physische Datenverarbeitungsbausteine** ist dabei für die hier im Vordergrund stehende Integration von Anwendungssystemen bzw. **Anwendungsbausteinen** nebensächlich. Wichtiger und hinsichtlich der Erfüllung von Integrationsanforderungen anspruchsvoller ist die Aufteilung auf mehrere **Datenbankverwaltungssysteme (DBVS)**¹.

Die Einhaltung des ACID-Prinzips für Transaktionen (atomicity, consistency, isolation, durability) ist nicht immer für alle an einer verteilten Transaktion, d. h. an einer verteilten Datenänderung, beteiligten **Anwendungsbausteine** erwünscht. Jedes einzelne der beteiligten **Datenbanksysteme** muss selbstverständlich die Einhaltung für sich selbst garantieren. In vielen Fällen wird die Autonomie des eine Transaktion auslösenden **Anwendungsbausteines** einer strikten Einhaltung des ACID-Prinzips vorgezogen. Die Einhaltung ist jedoch dann wichtig, wenn der auslösende **Anwendungsbaustein NICHT Master-Anwendungsbaustein** für die von der Transaktion betroffenen Objekttypen ist. In diesem Fall darf die Transaktion im auslösenden **Anwendungsbaustein** nur dann ausgeführt werden, wenn auch der **Master-Anwendungsbaustein**

¹ Die Feststellung soll keinesfalls die teilweise sehr anspruchsvolle Theorie und praktische Bereitstellung anderer Datenbankklassen herabsetzen. Die Integration von **Anwendungsbausteinen** mit integrierten **Datenbanksystemen** oder mit **Datenbanksystemen** der Klasse *Shared-Everything* ist insofern oft einfacher, als die Integration von den Herstellern „mitgeliefert“ wird und nicht erst in speziellen Projekten hergestellt werden muss.

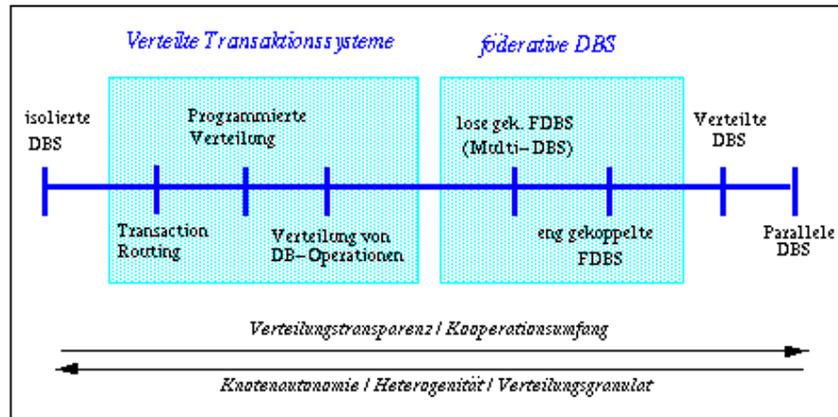


Abbildung 10.1: Lösungsspektrum zur Unterstützung heterogener Datenbanken (Quelle: [RAHM 1994], Abschnitt 10.4)

sie vollständig ausführt. Eine verzögerte oder sogar fehlende Ausführung einer Transaktion in **Anwendungsbausteinen**, die nicht **Master** für die betroffenen **Objekttypen** sind, trotzdem aber zu entsprechenden Datendomänen gehören, wird oft in Kauf genommen.

Der soeben ausgeführte Gedanke lässt sich mit Hilfe des Konzeptes der **Master-Anwendungsbausteine** (vgl. Abschnitt 6.3.1) folgendermaßen zusammenfassen: Die Einhaltung des ACID-Prinzips ist genau dann erforderlich, wenn der eine Transaktion auslösende **Anwendungsbaustein** nicht **Master-Anwendungsbaustein** für die betroffenen **Objekttypen** ist, oder nicht der einzige an der Transaktion beteiligte **Master-Anwendungsbaustein** für die betroffenen **Objekttypen** ist. Die Einhaltung ist zwischen dem auslösenden **Anwendungsbaustein** und allen (anderen) beteiligten **Master-Anwendungsbausteinen** erforderlich. Folglich ist eine sorgfältige Auswahl der **Master-Anwendungsbausteine** notwendig.

Eine verteilte Transaktion kann so definiert sein, dass in allen beteiligten **Anwendungsbausteinen** die gleichen Aktionen auszuführen sind, oder dass in den beteiligten **Anwendungsbausteinen** unterschiedliche Aktionen auszuführen sind, die jedoch zur selben Transaktion gehören. Erstere werden hier als *Abgleichstransaktionen* bezeichnet, letztere als *Kooperationsstransaktionen*.

Transaktionen und das 3LGM²

Das 3LGM² und seine Erweiterung zum 3LGM_A² wurden als Metamodelle für die Unterstützung des Informationsmanagements entwickelt, nicht für die ausführliche Softwarespezifikation oder den Datenbankentwurf. Trotzdem kann auch für das 3LGM_A² ein Bezug zum Transaktionsbegriff hergestellt werden.

Ein bearbeitender Zugriff einer **Aufgabe** auf einen **Objekttyp** kann einen Datenabgleich zwischen **Anwendungsbausteinen** bzw. ihren **Datenbanksystemen** notwendig machen. Jeder **Anwendungsbaustein**, der eine **Aufgabe** unterstützt, die den zuvor bearbeiteten **Objekttyp** interpretiert oder bearbeitet, muss Zugriff auf die aktualisierten Daten, die den betreffenden **Objekttyp** repräsentieren, haben. Der Zugriff kann entweder auf das eigene **Datenbanksystem** erfolgen, oder als Anfrage über Kommunikationsverbindungen auf die **Datenbanksysteme** anderer **Anwendungsbausteine**. Diese **Datenbanksysteme** müssen bei der Bearbeitung des be-

treffenden Objekttyps aktualisiert werden, d. h. es müssen Kommunikationsverbindungen für ihren Abgleich existieren. Diese Aktualisierung kann als Transaktion aufgefasst werden.

10.2 Informationale und funktionale Abhängigkeit

10.2.1 Der informationale Abhängigkeitsgrad

Zwischen Anwendungsbausteinen, an die bestimmte Integrationsanforderungen gestellt werden, bestehen Abhängigkeiten. In vielen Fällen müssen anwendungsbezogene Daten zwischen den Anwendungsbausteinen ausgetauscht werden können. Das wird hier als *informationale Abhängigkeit* der Anwendungsbausteine bezeichnet.

Hinweis: Da in abhängigen Anwendungsbausteinen dieselben Informationen (Objekttypen) u. U. durch unterschiedliche Daten (Datensatztypen und Dokumententypen) repräsentiert werden, die bei der Kommunikation übersetzt werden müssen, wird hier, mit Bezug auf die Informationen, die Bezeichnung *informationale Abhängigkeit* verwendet und nicht eine Bezeichnung wie *Datenabhängigkeit*.

Zur formalen Beschreibung der informationalen Abhängigkeit wird hier zunächst ein Prädikat eingeführt. Das Prädikat $\text{informational_abhaengig}(awb_1, awb_2, objt)$ gibt an, ob der Anwendungsbaustein awb_1 bezüglich des Objekttyps $objt$ informationale abhängig vom Anwendungsbaustein awb_2 ist. Anders ausgedrückt: es gibt an, ob Daten, die den Objekttyp $objt$ repräsentieren, vom Anwendungsbaustein awb_2 zum Anwendungsbaustein awb_1 kommuniziert werden müssen.

$$\text{informational_abhaengig}(awb_1, awb_2, objt) := awb_1 \text{ ist bezüglich } objt \text{ informationale abhängig von } awb_2 \quad (10.1)$$

Der Wahrheitswert des Prädikats $\text{informational_abhaengig}$ hängt davon ab,

- a) ob awb_1 und awb_2 zur Datendomäne von $objt$ gehören und
- b) ob awb_2 Master-Anwendungsbaustein für $objt$ ist.

Die Definition für $\text{informational_abhaengig}$ kann also folgendermaßen präzisiert werden:

$$\text{informational_abhaengig}(awb_1, awb_2, objt) := awb_1, awb_2 \in DDom_{objt} \wedge \text{hat_als_Master}(objt, awb_2) \quad (10.2)$$

Mit Hilfe des definierten Prädikates kann der informationale Abhängigkeitsgrad $AGI(awb)$ eines Anwendungsbausteines awb bestimmt werden. Er wird über die Anzahl derjenigen Anwendungsbausteine bestimmt, zu denen eine informationale Abhängigkeit besteht. Dabei werden die Anwendungsbausteine mehrfach gezählt, zu denen mehrfache Abhängigkeiten bestehen:

$$AGI(awb) := |\{awb_i, objt_j\} | \text{informational_abhaengig}(awb, awb_i, objt_j) \quad (10.3)$$

10.2.2 Der funktionale Abhängigkeitsgrad

Bei funktionaler Integration benötigt ein Anwendungsbaustein Funktionalität, die ein anderer bereitstellt. Das wird hier als *funktionale Abhängigkeit* bezeichnet. Nach der Überarbeitung des 3LGM² in Abschnitt 7.2.2 kann das Bereitstellen von Funktionalität über die Klasse `Operation`

und ihre Assoziationsbeziehung zur Klasse `Aufgabe` modelliert werden.

Auch zur formalen Beschreibung der funktionalen Abhängigkeit wird hier zunächst ein Prädikat eingeführt. Das Prädikat `funktional_abhaengig(awb1, awb2, auf)` gibt an, ob der Anwendungsbaustein *awb*₁ bezüglich der Aufgabe *auf* funktional abhängig vom Anwendungsbaustein *awb*₂ ist. Anders ausgedrückt: es gibt an, ob Operationen, die zur Erfüllung von Aufgabe *auf* dienen, von Anwendungsbaustein *awb*₁ bei Anwendungsbaustein *awb*₂ aufgerufen werden.

$$\text{funktional_abhaengig}(awb_1, awb_2, auf) := awb_1 \text{ ist bezüglich } auf \text{ funktional abhängig von } awb_2 \quad (10.4)$$

Der Wahrheitswert des Prädikats `funktional_abhaengig` hängt davon ab,

- a) ob *awb*₁ und *awb*₂ zur funktionalen Domäne von *auf* gehören und
- b) ob *awb*₂ Master-Anwendungsbaustein für *auf* ist.

Die Definition für `funktional_abhaengig` kann also folgendermaßen präzisiert werden:

$$\text{funktional_abhaengig}(awb_1, awb_2, auf) := awb_1, awb_2 \in FDom_{auf} \wedge \text{hat_als_Master}(auf, awb_2) \quad (10.5)$$

Mit Hilfe des definierten Prädikates kann der funktionale Abhängigkeitsgrad $AGF(awb)$ eines Anwendungsbausteines *awb* bestimmt werden. Er wird über die Anzahl derjenigen Anwendungsbausteine bestimmt, zu denen eine funktionale Abhängigkeit besteht. Dabei werden die Anwendungsbausteine mehrfach gezählt, zu denen mehrfache Abhängigkeiten bestehen:

$$AGF(awb) := |\{awb_i, auf_j\} | \text{funktional_abhaengig}(awb, awb_i, auf_j) \quad (10.6)$$

10.3 Ausführungsabhängigkeit, transaktionale Abhängigkeit und Transaktionsstärke

10.3.1 Die Ausführungsabhängigkeit

Eine zunächst sehr einfach zu beschreibende Abhängigkeit zwischen Anwendungsbausteinen ist die Ausführungsabhängigkeit. Sie beschreibt die Tatsache, dass Anwendungsbausteine bei ihrer Ausführung durch andere Anwendungsbausteine blockiert werden können.

Auf der Basis der in Abschnitt 7.2.2 beschriebenen Überarbeitung des 3LGM² kann die Ausführungsabhängigkeit von Anwendungsbausteinen unter Nutzung des Attributs *Synchronitätsmodus* der Klasse `Operation` bestimmt werden. Zur Formalisierung wird hier der Ausführungsabhängigkeitsgrad eingeführt.

Definition und Berechnungsvorschrift

Definition 10.1 Der **Ausführungsabhängigkeitsgrad** AAG gibt für einen Anwendungsbaustein *awb* an, von wie vielen anderen Anwendungsbausteinen er bei der Ausführung blockiert werden kann.

Zur Berechnung des Ausführungsabhängigkeitsgrades werden die Operationen gezählt, die der Anwendungsbaustein *awb* aufruft und deren Attribut *Synchronitätsmodus* den Wert *synchron*

hat:

$$AAG(awb \in \underline{AWB}) := |\{bss_i \in \underline{BSS}\}| \left| \begin{array}{l} \text{besitzt}(awb, bss_i) \\ \exists \left(\begin{array}{l} bss_j \in \underline{BSS} \\ op \in \underline{OPER} \end{array} \right) \left| \begin{array}{l} \text{ruft_auf}(bss_i, bss_j, op) \\ \text{Synchronitätsmodus}(op) = \text{synchron} \end{array} \right. \end{array} \right. \wedge \quad (10.7)$$

Ende — Definition und Berechnungsvorschrift

Das in der Berechnungsvorschrift benutzte Prädikat `ruft_auf` gibt für zwei Bausteinschnittstellen bss_i und bss_j sowie eine Operation op an, ob eine Kommunikationsbeziehung existiert, bei der bss_i aufrufende Bausteinschnittstelle ist, bss_j aufgerufene Bausteinschnittstelle ist und op die aufgerufene Operation ist. Auf eine formale Definition des Prädikates wird hier verzichtet.

Die in der Berechnungsvorschrift benutzte Funktion *Synchronitätsmodus* gibt für eine Operation op den Wert des Attributes *Synchronitätsmodus* an. Auch dafür wird hier auf eine formale Definition verzichtet.

10.3.2 Die transaktionale Abhängigkeit

Eine weitere Abhängigkeit zwischen Anwendungsbausteinen ist die transaktionale Abhängigkeit. Sie beschreibt die Tatsache, dass Anwendungsbausteine eine Transaktion u. U. zurücknehmen müssen, wenn andere an der betreffenden Transaktion beteiligte Anwendungsbausteine die Durchführung nicht mit einem Commit bestätigen.

Auf der Basis der in Abschnitt 7.2.2 beschriebenen Überarbeitung des 3LGM² kann die transaktionale Abhängigkeit von Anwendungsbausteinen unter Nutzung des Attributs *Transaktionsmodus* der Assoziationsklasse *Kommunikationsbeziehung* bestimmt werden. Zur Formalisierung wird hier der Transaktionsabhängigkeitsgrad eingeführt.

Definition und Berechnungsvorschrift

Definition 10.2 Der **Transaktionsabhängigkeitsgrad** *TAG* gibt für einen Anwendungsbaustein awb an, wieviele Anwendungsbausteine die Rücknahme einer Transaktion durch awb verursachen können.

Zur Berechnung des Transaktionsabhängigkeitsgrades werden die *Kommunikationsbeziehungen* gezählt, die der Anwendungsbaustein awb nutzt und deren Attribut *Transaktionsmodus* den Wert *Commit erforderlich* hat:

$$TAG(awb \in \underline{AWB}) := |\{bss_i \in \underline{BSS}\}| \left| \begin{array}{l} \text{besitzt}(awb, bss_i) \\ \exists \left(\begin{array}{l} bss_j \in \underline{BSS} \\ op \in \underline{OPER} \end{array} \right) \left| \begin{array}{l} \text{ruft_auf}(bss_i, bss_j, op) \\ \text{Transaktionsmodus}(bss_i, bss_j, op) = \text{Commit} \\ \text{erforderlich} \end{array} \right. \end{array} \right. \wedge \quad (10.8)$$

Ende — Definition und Berechnungsvorschrift

Die in der Berechnungsvorschrift benutzte Funktion *Transaktionsmodus* gibt für zwei Bausteinschnittstellen bss_i und bss_j sowie eine Operation op den Wert des Attributes *Transaktionsmodus* der *Kommunikationsbeziehung* an, über die op von bss_i bei bss_j aufgerufen wird. Auf eine formale Definition der Funktion wird hier verzichtet.

10.3.3 Die Transaktionsstärke einer Transaktionsausführung

Sowohl informationale als auch funktionale Abhängigkeit setzen eine besondere Qualität der Kommunikation voraus, damit die Integrationsanforderungen tatsächlich erfüllt werden können. Um diese Kommunikationsqualität beschreiben zu können, wird in diesem Abschnitt der Begriff der *Transaktionsstärke* eingeführt. Die Stärke einer Transaktion ist dabei nicht mit der Definition einer Transaktion festgelegt. Sie beschreibt einen Zustand der Integration, genauer: Sie beschreibt die Güte der Integration hinsichtlich der Gewährleistung von Datenintegrität. Aus der Definition einer Transaktion ergeben sich jedoch möglicherweise Vorgaben für die Transaktionsstärke der Transaktionsausführung.

Definition und Berechnungsvorschrift

Definition 10.3 Die **einfache Transaktionsstärke** *ase* gibt an, wie viele der beteiligten **Anwendungsbausteine** ihren Teil der Transaktion bestätigen müssen, damit die Transaktion im auslösenden **Anwendungsbaustein** als ausgeführt gilt und folglich nicht zurückgenommen werden muss.

Die einfache Transaktionsstärke *ase* einer Transaktionsausführung *ta* wird folgendermaßen berechnet:

1. Der Anfangswert der Berechnung ist 1.
2. Für jeden beteiligten **Nicht-Master-Anwendungsbaustein**, der seinen Teil der Transaktion NICHT bestätigen muss, wird der Wert $1/(1+\text{Anzahl der beteiligten Nicht-Master-Anwendungsbausteine})$ abgezogen.
3. Für jeden **Master-Anwendungsbaustein** für einen der betroffenen **Objekttypen**, der seinen Teil der Transaktion NICHT bestätigen muss, wird der Wert 1 abgezogen.

Ende — Definition und Berechnungsvorschrift

Für bestimmte Aussagen über die Integrationsqualität ist eine Standardisierung der Transaktionsstärke auf einen Wertebereich zwischen -1 und $+1$ hilfreich.

Definition und Berechnungsvorschrift

Definition 10.4 Die *standardisierte Transaktionsstärke* *ass* ist die einfache Transaktionsstärke einer Transaktionsausführung geteilt durch die Anzahl der **Master-Anwendungsbausteine** der betroffenen **Objekttypen**.

Ende — Definition und Berechnungsvorschrift

10.3.4 Kategorien von Transaktionsausführungen

Zur Erleichterung der Beschreibung von Transaktionsausführungen werden hier die folgenden Kategorien unterschieden:

vollständige Ausführungen: Transaktionsausführungen mit einer standardisierten Transaktionsstärke gleich 1 werden hier als *vollständig* bezeichnet.

zulässige Ausführungen: Transaktionsausführungen mit einer standardisierten Transaktionsstärke größer 0 werden hier als *zulässig* bezeichnet.

unzulässige Ausführungen: Transaktionsausführungen mit einer standardisierten Transaktionsstärke kleiner oder gleich 0 werden hier als *unzulässig* bezeichnet.

Vollständige Ausführungen sind also Spezialfälle der zulässigen Ausführungen.

10.4 Ein Anwendungsszenario aus dem Universitätsklinikum Leipzig

Die Anwendung der in diesem Kapitel vorgestellten Kennzahlberechnungen wird hier an einem Beispielszenario aus dem Universitätsklinikum Leipzig (UKL) demonstriert.

10.4.1 Vorstellung des Anwendungsszenarios

Das hier verwendete Anwendungsszenario ist eine Erweiterung des Anwendungsszenarios aus Kapitel 9. Abbildung 10.2 zeigt einen Auszug aus dem Informationssystem des UKL, modelliert auf der Basis des 3LGM². Wichtige **Anwendungsbausteine** sind beispielsweise das Patientenverwaltungssystem, das OP-Planungssystem, das OP-Dokumentationssystem, das Radiologieinformationssystem und das Laborinformationssystem. Der zentrale **Anwendungsbaustein** für die Unterstützung der Integration ist der Kommunikationsserver (vgl. Beschreibung des Anwendungsszenarios in Abschnitt 9.2).

Es wird angenommen, dass eine grundlegende Überarbeitung der Integrationsinfrastruktur geplant ist. Die Kommunikation zwischen den **Anwendungsbausteinen** soll zukünftig dienstorientiert gestaltet werden. Mit der Umgestaltung soll bei der Infrastruktur für die Kommunikation von Falldaten begonnen werden. Dabei sollen Falldaten von einem Master-Patient-Index auf der Basis des Person Identification Service der OMG bereitgestellt werden (Abbildung 10.3).

Alte Architektur

Bisher wurden Falldaten im **Master-Anwendungsbaustein** für den Objekttyp *Fall*, dem Patientenverwaltungssystem, erfasst und über den Kommunikationsserver an andere **Anwendungsbausteine** verteilt. Die Übermittlung der Daten erfolgte zum größten Teil asynchron über Nachrichtendateien und TCP-Sockets. Abbildung 10.4a hebt

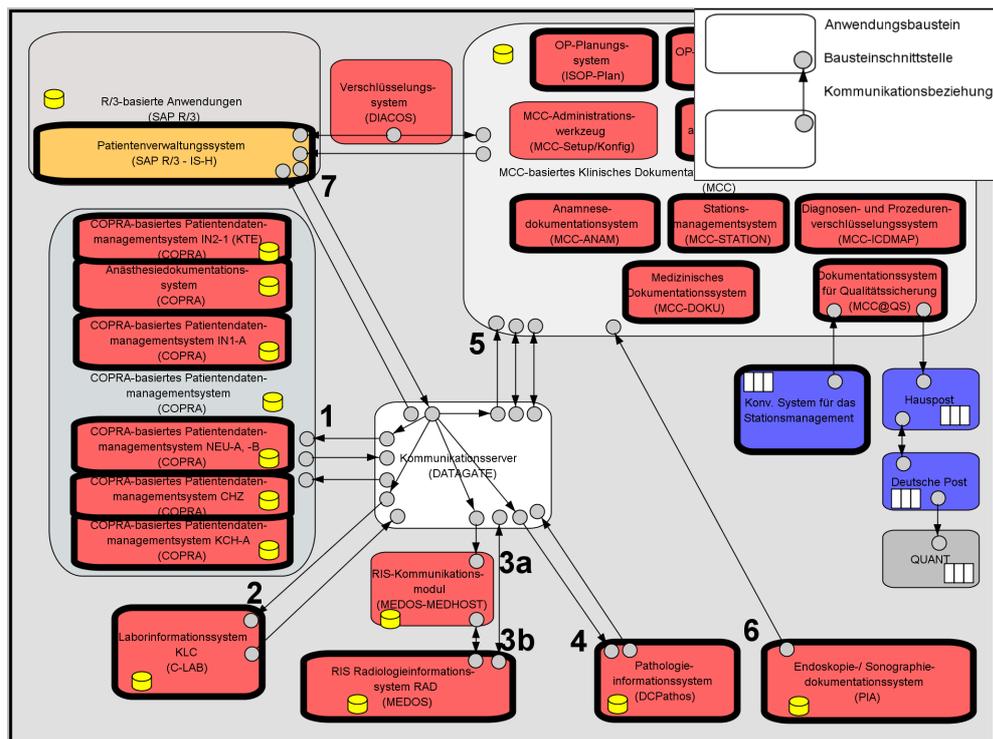
- die Kommunikationsverbindungen der Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *NP1110*² und des Anwendungsbausteines *Patientenverwaltungssystem* (Kommunikationsverbindungen 1 bis 5 ohne 3b) und
- die Kommunikationsverbindungen der Übermittlungsdomänen des Objekttyps *Fall*, des Ereignistyps *A19*³ sowie der Anwendungsbausteine *RIS-Kommunikationsmodul* und *KDMS-MCC* (Kommunikationsverbindungen 3b und 6)

² Der Ereignistyp *NP1110* ist im Kommunikationsstandard *SAP-HCM* definiert und wird im Zusammenhang mit stationären Patientenaufnahmen ausgelöst. Er entspricht dem Ereignistyp *A01* des Kommunikationsstandards *HL7*.

³ Der Ereignistyp *A19* ist im Kommunikationsstandard *HL7* definiert und wird im Zusammenhang mit einer Abfrage von Falldaten ausgelöst.

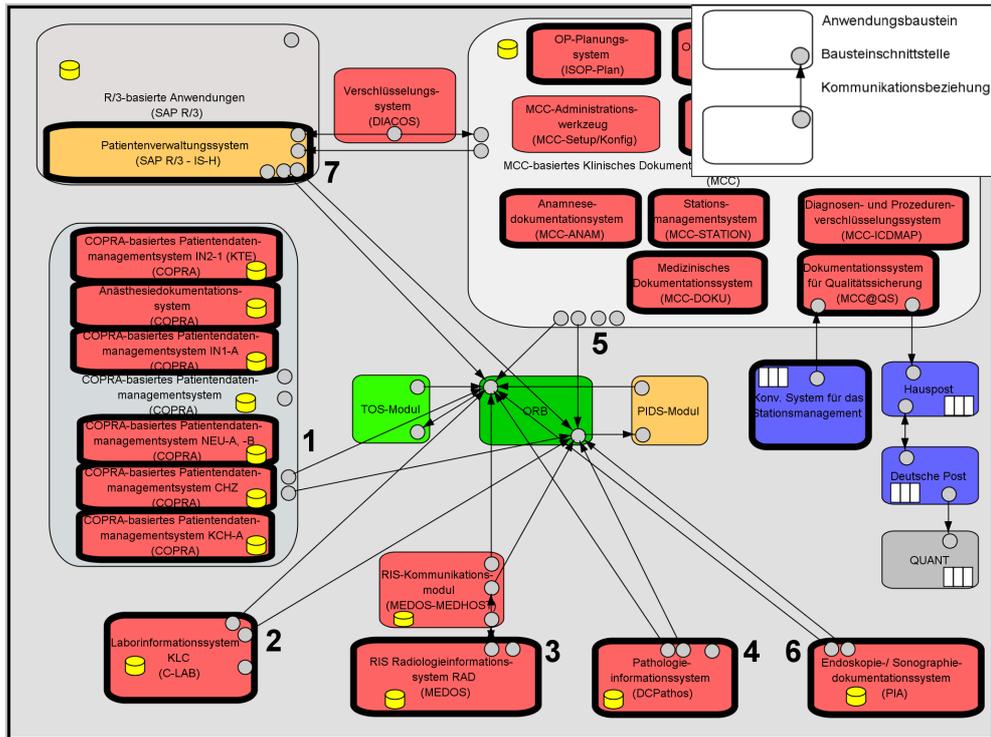
hervor. Die Kommunikationsverbindungen 1 bis 5 ohne 3b gehen vom Patientenverwaltungssystem aus und führen zu den Anwendungsbausteinen, die Änderungen von Daten zum Objekttyp *Fall* auch in ihren Datenbanksystemen durchführen müssen. Zur Vereinfachung wird hier angenommen, dass über die Kommunikationsbeziehungen der Kommunikationsverbindungen 1 bis 5 ohne 3b jeweils nur eine Operation aufgerufen wird, die die zu übermittelnden Falldaten als Parameter entgegennimmt. Die Kommunikationsverbindungen 3b und 6 übermitteln Anfragedatensätze an die Anwendungsbausteine *RIS-Kommunikationsmodul* und *KDMS-MCC* als Parameter der aufgerufenen Operationen und Anfrageergebnisse als Ergebnisse der aufgerufenen Operationen.

In der alten Architektur besteht eine große Unabhängigkeit der betrachteten Anwendungsbausteine hinsichtlich der Kommunikation von Falldaten: Bei den Kommunikationsverbindungen 1 bis 5 (ohne 3b) hat das Attribut *Synchronitätsmodus* aller aufgerufenen Operationen jeweils den Wert *asynchron* und das Attribut *Transaktionsmodus* jeweils den Wert *Commit nicht erforderlich*. Bei den Kommunikationsverbindungen 3b und 6 hat das Attribut *Synchronitätsmodus* der aufgerufenen Operationen jeweils den Wert *synchron* und das Attribut *Transaktionsmodus* jeweils den Wert *Commit nicht erforderlich* (Abbildung 10.4a).



Der Startpunkt (7) und die Endpunkte (1-6) der elektronischen Kommunikationsverbindungen für Falldaten sind nummeriert.

Abbildung 10.2: Alte Architektur im Anwendungsszenario aus dem UKL; hervorgehoben sind die Anwendungsbausteine der Datendomäne des Objekttyps *Fall*, des Ereignistyps *NP1110* und des Anwendungsbausteines *Patientenverwaltungssystem*



Die Startpunkte der elektronischen Kommunikationsverbindungen zum PIDS-Modul sind nummeriert.

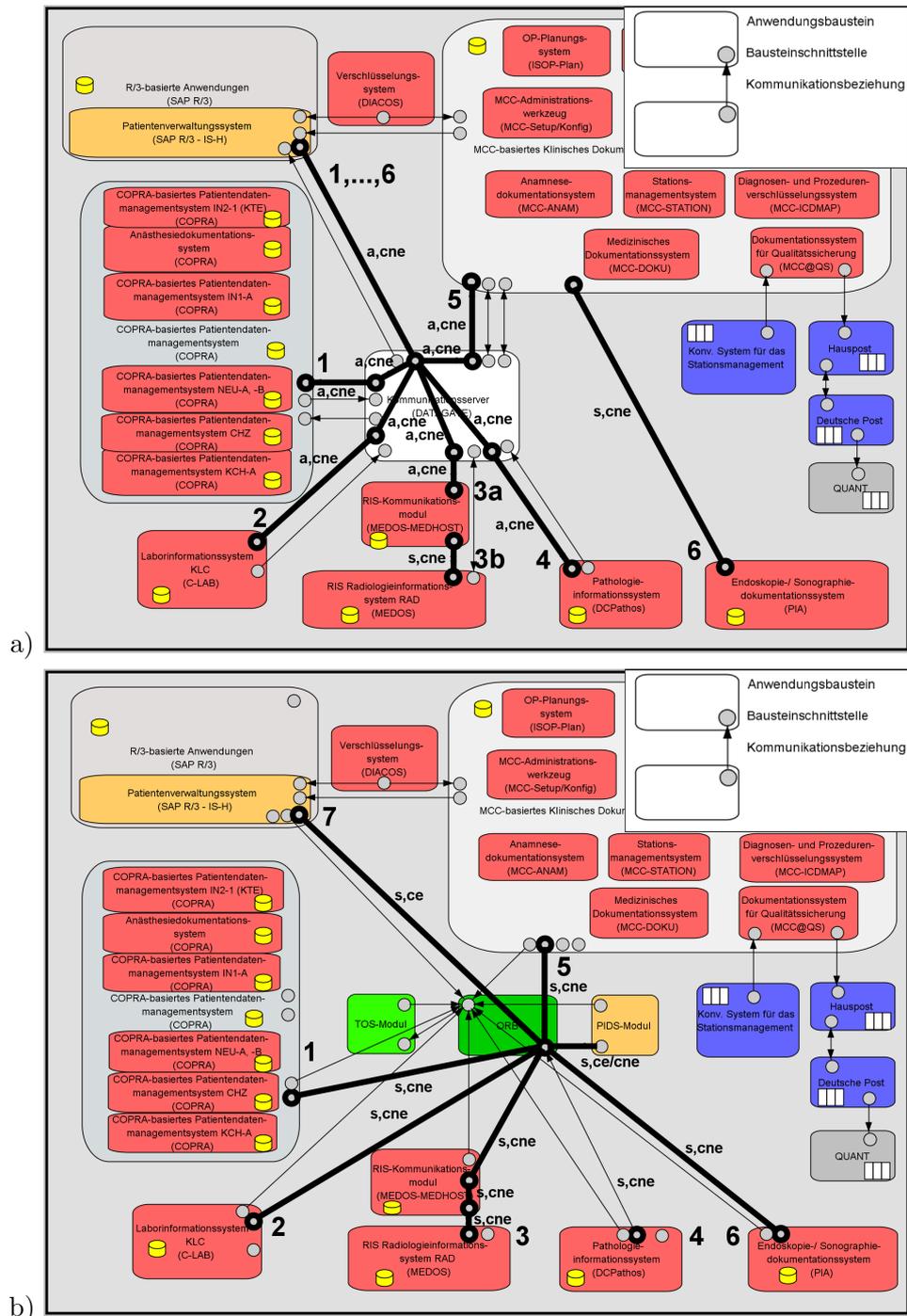
Abbildung 10.3: Geplante Architektur im Anwendungsszenario aus dem UKL; hervorgehoben sind die **Anwendungsbausteine** der Datendomäne des Objekttyps *Fall*, des Ereignistyps *NP1110* und **Anwendungsbausteines** *Patientenverwaltungssystem*

Geplante Architektur

Zukünftig sollen Falldaten von einem zentralen Falldatenverzeichnis, einem Master-Patient-Index (MPI), abgerufen werden. Der MPI sei hier eine Implementierung des Person Identification Service (PIDS) der Object Management Group (OMG) (vgl. Abschnitt 4.2.3). **Anwendungsbausteine**, die die beiden Objekttypen interpretieren, sollen die aktuellen Daten zu einem Fall immer dann vom PIDS abrufen, wenn sie benötigt werden. Es soll weiterhin möglich sein, Falldaten im Patientenverwaltungssystem zu erfassen. Abbildung 10.4b zeigt einen Entwurf für die neue Integrationsinfrastruktur, zunächst für die Falldaten. Er enthält u. a. einen **Anwendungsbaustein**, der den PIDS zur Verfügung stellt, und einen Object Request Broker (ORB) als zentralen **Anwendungsbaustein** für die Integration. In der Abbildung sind

- die Kommunikationsverbindung der Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *NP1110* und des **Anwendungsbausteines** *Patientenverwaltungssystem* (Kommunikationsverbindung 7) und
- die Kommunikationsverbindungen der Übermittlungsdomäne des Objekttyps *Fall*, des Ereignistyps *A19* und des **Anwendungsbausteines** *PIDS* (Kommunikationsverbindungen 1 bis 6)

hervorgehoben. Zu letztgenannten **Anwendungsbausteinen** gehören u. a. das Laborinformationssystem und Radiologieinformationssystem.



Bei a) sind die Endpunkte der Kommunikationsverbindungen vom Patientenverwaltungssystem zu den „Ziel“-Anwendungsbausteinen der Falldaten nummeriert. Bei b) sind die Startpunkte der Kommunikationsverbindungen zum PIDS-Modul nummeriert. Die Kommunikationsbeziehungen sind mit Abkürzungen für das Attribut *Transaktionsmodus* und für das Attribut *Synchronitätsmodus* versehen: synchron, asynchron, Commit erforderlich, Commit nicht erforderlich

Abbildung 10.4: Anwendungsszenario aus dem UKL mit Kommunikationsserver (a) und mit ORB (b); hervorgehoben sind die Kommunikationsverbindungen zur Verteilung der Falldaten

Die Kommunikationsverbindung 7 geht vom Patientenverwaltungssystem aus und führt zum PIDS-Modul, der die Änderungen von Daten zum Objekttyp *Fall* auch in seinem Datenbanksystem durchführen muss. Zur Vereinfachung wird hier angenommen, dass über die Kommunikationsverbindung 7 nur die Operation *find_or_register_ids* des PIDS aufgerufen wird (vgl. [OMG 2001c], S. 2-28 - 2-32). Die Kommunikationsverbindungen 1 bis 6 übermitteln Anfragen an den Anwendungsbaustein *PIDS-Modul* als Parameter der aufgerufenen Operationen und Anfrageergebnisse als Ergebnisse der aufgerufenen Operationen. Zur Vereinfachung wird hier angenommen, dass über die Kommunikationsverbindungen 1 bis 6 nur die Operation *find_candidates* des PIDS aufgerufen wird (vgl. [OMG 2001c], S. 2-19 - 2-21). Ebenfalls zur Vereinfachung wurden, im Gegensatz zum Beispiel in Abschnitt 7.6.2, am ORB die Schnittstellen-Elemente für die IDL-Stubs und die Skeletons zusammengefasst.

Über die für die geplante Architektur betrachteten vereinigten Übermittlungsdomänen soll dieselbe Funktionalität bereit gestellt werden wie über die für die alte Architektur betrachtete Übermittlungsdomäne: die Verteilung der Falldaten⁴.

In der geplanten Architektur sind sowohl das Patientenverwaltungssystem als auch der PIDS-Modul *Master-Anwendungsbaustein* für den Objekttyp *Fall*. Um die Bestimmung nicht zulässiger Transaktionen zu demonstrieren, wurde die in der Praxis notwendige Kommunikationsverbindung zur Übermittlung von aktualisierten Falldaten vom PIDS-Modul zum Patientenverwaltungssystem weggelassen.

In der geplanten Architektur besteht eine wesentlich geringere Unabhängigkeit der betrachteten Anwendungsbausteine hinsichtlich der Kommunikation von Falldaten: Bei den Kommunikationsverbindungen 1 bis 6 hat das Attribut *Synchronitätsmodus* aller aufgerufenen Operationen jeweils den Wert *synchron* und das Attribut *Transaktionsmodus* jeweils den Wert *Commit nicht erforderlich*. Bei der Kommunikationsverbindungen 7 hat das Attribut *Synchronitätsmodus* der aufgerufenen Operationen jeweils den Wert *synchron* und das Attribut *Transaktionsmodus* jeweils den Wert *Commit erforderlich* (Abbildung 10.4b).

10.4.2 Kennzahlen zur Abhängigkeit für das Anwendungsszenario

Im Folgenden werden nur die Anwendungsbausteine berücksichtigt, die zu den bei der Vorstellung des Anwendungsszenarios in Abschnitt 10.4 beschriebenen Übermittlungsdomänen gehören.

Für die vorgestellte alte Architektur haben die in diesem Kapitel vorgestellten Kennzahlen folgende Werte:

1. $AGI(\text{Patientenverwaltungssystem}) = 0$.
2. $AGI(awb_i) = 1$ für
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Radiologieinformationssystem RAD,
 - das Pathologieinformationssystem,
 - das MCC-basierte Klinische Dokumentations- und Managementsystem und
 - das Endoskopie-/Sonographiedokumentationssystem.

⁴ Ereignistypen zu Verlegungen, Entlassungen, Patientenzusammenführungen oder weiteren fallbezogenen Ereignissen werden hier zur Vereinfachung nicht berücksichtigt.

Diese **Anwendungsbausteine** sind vom Patientenverwaltungssystem informational abhängig.

3. $AGF(awb_i) = 0$ für
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Pathologieinformationssystem und
 - das MCC-basierte Klinische Dokumentations- und Managementsystem.
4. $AGF(awb_i) = 1$ für
 - das Radiologieinformationssystem RAD und
 - das Endoskopie-/Sonographiedokumentationssystem.

Das Radiologieinformationssystem ist vom RIS-Kommunikationsmodul funktional abhängig, das Endoskopie-/Sonographiedokumentationssystem vom MCC-basierten Klinischen Dokumentations- und Managementsystem. Die funktionale Abhängigkeit ist auf die **Aufgabe Suche von Falldaten** bezogen.

5. $AAG(awb_i) = 0$ für
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Pathologieinformationssystem und
 - das MCC-basierte Klinische Dokumentations- und Managementsystem.
6. $AAG(awb_i) = 1$ für
 - das Radiologieinformationssystem RAD und
 - das Endoskopie-/Sonographiedokumentationssystem.

Das Radiologieinformationssystem hängt bei der Ausführung vom RIS-Kommunikationsmodul ab, das Endoskopie-/Sonographiedokumentationssystem vom MCC-basierten Klinischen Dokumentations- und Managementsystem.

7. $TAG(awb_i) = 0$ für alle **Anwendungsbausteine**.
8. $ase = 1/6$ für die Transaktion, die durch den bearbeitenden Zugriff auf den **Objekttyp Fall** im Patientenverwaltungssystem ausgelöst wird.
9. $ass = 1/6$ für die beschriebene Transaktion; die Transaktion ist folglich zulässig.

Für die vorgestellte geplante Architektur haben die in diesem Kapitel vorgestellten Kennzahlen folgende Werte:

1. $AGI(awb_i) = 0$ für
 - das Patientenverwaltungssystem und
 - den PIDS-Modul.
2. $AGI(awb_i) = 2$ für
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Radiologieinformationssystem RAD,
 - das Pathologieinformationssystem,
 - das MCC-basierte Klinische Dokumentations- und Managementsystem und
 - das Endoskopie-/Sonographiedokumentationssystem.

Diese **Anwendungsbausteine** sind vom Patientenverwaltungssystem und vom PIDS-

Modul informational abhängig.

3. $AGF(awb_1) = 1$ für
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Radiologieinformationssystem RAD,
 - das Pathologieinformationssystem,
 - das MCC-basierte Klinische Dokumentations- und Managementsystem und
 - das Endoskopie-/Sonographiedokumentationssystem.

Diese **Anwendungsbausteine** sind vom PIDS-Modul funktional abhängig. Die funktionale Abhängigkeit ist auf die **Aufgabe Suche von Falldaten** bezogen.

4. $AAG(awb_i) = 0$ für
 - den PIDS-Modul.
5. $AAG(awb_i) = 1$ für
 - das Patientenverwaltungssystem,
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Radiologieinformationssystem RAD,
 - das Pathologieinformationssystem,
 - das MCC-basierte Klinische Dokumentations- und Managementsystem und
 - das Endoskopie-/Sonographiedokumentationssystem.

Diese **Anwendungsbausteine** hängen bei der Ausführung vom PIDS-Modul ab.

6. $TAG(awb_i) = 0$ für
 - den PIDS-Modul,
 - das COPRA-basierte Patientendatenmanagementsystem,
 - das Laborinformationssystem,
 - das Radiologieinformationssystem RAD,
 - das Pathologieinformationssystem,
 - das MCC-basierte Klinische Dokumentations- und Managementsystem und
 - das Endoskopie-/Sonographiedokumentationssystem.
7. $TAG(awb_i) = 1$ für das Patientenverwaltungssystem.
8. $ase = 1$ für die Transaktion, die durch den bearbeitenden Zugriff auf den Objekttyp *Fall* im Patientenverwaltungssystem ausgelöst wird.
9. $ass = 1$ für die beschriebene Transaktion; die Transaktion ist folglich zulässig.
10. $ase = 0$ für die Transaktion, die durch den bearbeitenden Zugriff auf den Objekttyp *Fall* im PIDS-Modul ausgelöst wird.
11. $ass = 0$ für die beschriebene Transaktion; die Transaktion ist folglich unzulässig.

In der geplanten Architektur besteht eine höhere Abhängigkeit der **Anwendungsbausteine**. Sie muss außerdem um eine Kommunikationsverbindung zur Übermittlung von aktualisierten Falldaten vom PIDS-Modul zum Patientenverwaltungssystem ergänzt werden.

11 Heterogenitätsbewertung von Kommunikationsverbindungen

11.1 Vorbemerkungen

In vielen Informationssystemen werden viele unterschiedliche Integrationstechniken verwendet. Diese Heterogenität wird oft dadurch gefördert, dass die Auswirkungen der Einführung einer neuen Integrationstechnik und das Einsparungspotential bei der Nutzung vorhandener Integrationstechniken nicht analysiert wird.

Die in diesem Kapitel erarbeiteten Kennzahlen unterstützen die Analyse der Heterogenität, bis hin zu einem Versuch der Abschätzung von Einsparungspotential bei Integrationskosten.

11.2 Der absolute Heterogenitätsgrad von Kommunikationsverbindungen

11.2.1 Der absolute Heterogenitätsgrad von zwei Kommunikationsverbindungen

Der absolute Heterogenitätsgrad H erlaubt eine quantitative Einschätzung der Wiederverwendung von Integrationstechniken. Er gibt an, wie stark sich Kommunikationsverbindungen hinsichtlich der beteiligten Kommunikationsbeziehungen und der zugehörigen Vermittlungsbeziehungen unterscheiden.

Für die Bestimmung des Heterogenitätsgrades werden Kommunikationsverbindungen hinsichtlich der zugehörigen Bausteinschnittstellen und der aufgerufenen Operationen miteinander verglichen. Im Abschnitt 8.6.1 wurden dazu Kommunikationsverbindungen als Folgen von Kommunikationsbeziehungen formalisiert.

Zur Bestimmung des absoluten Heterogenitätsgrades H wird zunächst ein Gleichheitswert (similarity score) p bestimmt. Der Gleichheitswert von zwei Kommunikationsverbindungen wird ähnlich dem Gleichheitswert von zwei Nukleotidsequenzen bei der Genomanalyse [PEARSON 2001] bestimmt. Dort wird beim Ausrichten der Sequenzen aneinander für jede Abweichung eine „Strafe“ vergeben, für jede Übereinstimmung eine „Gutschrift“. Beim Ausrichten werden die Symbolsequenzen so gegenübergestellt, dass die Summe der „Strafen“ und „Gutschriften“ maximal wird. Dabei werden „Strafen“ i. d. R. mit einem negativen Wert versehen.

Hier wird der Needleman-Wunsch-Algorithmus (NW-Algorithmus) zur globalen Ausrichtung von Symbolsequenzen angewendet (Kasten „Exkurs: Ausrichtung von Symbolsequenzen“; [PEARSON 2001], S. 22-27, [NEEDLEMAN and WUNSCH 1970]). Der Algorithmus liefert für zwei Sequenzen eine Ausrichtung mit der geringsten Abweichung.

Für die Berechnung der Gleichheit von zwei Kommunikationsverbindungen werden die Sequenzen der Tupel der Kommunikationsbeziehungen aneinander ausgerichtet. Dabei werden die Abweichungen und Übereinstimmungen nach der folgenden Vorschrift bewertet. Die Abbildungen 11.1 und 11.2 können das Verstehen der Vorschrift unterstützen.

Kasten 11.1: Exkurs: Ausrichtung von Symbolsequenzen

Needleman-Wunsch-Algorithmus

Der Algorithmus von Needleman und Wunsch ([PEARSON 2001], S. 22-27, [NEEDLEMAN and WUNSCH 1970]) ist prinzipiell nicht auf die Ausrichtung von Nukleotidsequenzen beschränkt. Er berechnet rekursiv die besten Ausrichtungen für zwei Symbolsequenzen. In einer Matrix der zwei Sequenzen bestimmt er, oben links beginnend, für jedes Feld der Matrix einen Gleichheitswert (similarity score). Die Berechnungsvorschrift für den Gleichheitswert $p_{i,j}$ für das Feld an Position (i, j) ist:

$$\begin{aligned}
 p_{i,j} \mid i = 0 \vee j = 0 & := 0 \\
 p_{i,j} \mid i, j > 0 & := \max \begin{cases} p_{i-1,j} + s((i-1, j), (i, j)) \\ p_{i-1,j-1} + s((i-1, j-1), (i, j)) \\ p_{i,j-1} + s((i, j-1), (i, j)) \end{cases}
 \end{aligned}$$

Dabei ist $s((k, l), (i, j))$ der Wert für den Übergang vom Feld (k, l) zum benachbarten Feld (i, j) . Üblicherweise wird ein diagonalen Übergang von einem Feld $(i-1, j-1)$ zum Feld (i, j) mit einem positiven Wert bewertet („Gutschrift“), wenn die zugehörigen Symbole der beiden Sequenzen gleich sind. Jeder waagerechte oder senkrechte Übergang wird mit einem negativen Wert („Strafe“) bewertet. Waagerechte und senkrechte Übergänge, d. h. Übergänge, bei denen sich nur einer der beiden Indizes ändert, führen zum Einfügen einer Lücke (engl: gap) in derjenigen Sequenz, deren Index sich nicht ändert.

In der Beispielmatrix beträgt die „Gutschrift“ für einen diagonalen Übergang mit gleichen Symbolen 1. Die „Strafe“ für einen diagonalen Übergang mit ungleichen Symbolen beträgt -1 , für jeden anderen Übergang -2 .

Um die besten Ausrichtungen herauszufinden, werden, ausgehend von der unteren rechten Ecke der Matrix, die Pfade mit den größten Gleichheitswerten gesucht.

	A	B	D	D	E	F	G	H	I
A	\	\	\	\	\	\	\	\	\
B	\	!	\	\	\	\	\	\	\
D	\	\	!	\	\	\	\	\	\
E	\	\	!	!	\	\	\	\	\
G	\	\	!	!	\	\	\	\	\
K	\	\	\	!	!	!	!	\	\
H	\	\	\	\	!	!	!	!	\
I	\	\	\	\	\	!	!	!	!

Optimal global alignments (score 2):

A B D D E F G H I (top)
 A B D - E G K H I (side)
 or A B - D E G K H I
 (Bildquelle: [PEARSON 2001], S. 25)

Definition und Berechnungsvorschrift

Definition 11.1 Der absolute Gleichheitswert p von zwei Kommunikationsverbindungen ist ein Maß für die Abweichung der Kommunikationsverbindungen hinsichtlich der zugehörigen Bausteinschnittstellen und der aufgerufenen Operationen.

1. Für die Berechnung von p werden die Kommunikationsverbindungen aneinander ausgerichtet. Die Ausrichtung der Kommunikationsverbindungen erfolgt dazu nach einer Anpassung des Needleman-Wunsch-Algorithmus (NW-Algorithmus) zur globalen Ausrichtung von Symbolsequenzen.
2. Für die Berechnung des Gleichheitswertes (similarity score) werden die betrachteten Kommunikationsverbindungen in einer Matrix gegenübergestellt. Der Gleichheitswert wird darüber bestimmt, dass der Weg durch die Matrix gesucht wird, der den höchsten Gleichheitswert in der letzten (meist unteren rechten) Zelle ergibt.
3. Die Anpassung des NW-Algorithmus erfolgt durch folgende Regeln:

- 3.1. Die **Kommunikationsbeziehungen** werden entsprechend der Definitionsgleichung 8.49 in Abschnitt 8.6.2 als Tripelfolgen notiert¹.
- 3.2. Die **Kommunikationsverbindungen** werden zur Vorbereitung der Anwendung des NW-Algorithmus in Matrixform notiert. Jedes Tripel bildet ein Symbol im Sinne des NW-Algorithmus.
- 3.3. Der Gleichheitswert $d_{i,j}$ von zwei einzelnen **Kommunikationsbeziehungen** wird nach folgenden Regeln bestimmt:
 - 3.3.1. Der Startwert für die Berechnung ist 0.
 - 3.3.2. Es werden jeweils die aufgerufenen **Bausteinschnittstellen** miteinander und die **Operationen** miteinander verglichen.
 - 3.3.3. Ist eine der beiden verglichenen **Kommunikationsbeziehungen** das erste Element einer der betrachteten **Kommunikationsverbindungen**, werden auch die aufrufenden **Bausteinschnittstellen** miteinander verglichen.
 - 3.3.4. Für jede Übereinstimmung der **Bausteinschnittstellen** oder der **Operationen** wird eine „Gutschrift“ von 0 addiert.
 - 3.3.5. Für jede Abweichung einer **Bausteinschnittstelle** oder einer **Operation** wird eine „Strafe“ von -1 addiert.
- 3.4. Die im NW-Algorithmus verrechneten Werte für die Ausgangsfelder und für die Übergänge zwischen den Matrixfeldern werden nach den folgenden Regeln bestimmt:
 - 3.4.1. Für jedes Feld mit $i = 0$ gilt: $p_{i,j} := j * (-1)$.
 - 3.4.2. Für jedes Feld mit $j = 0$ gilt: $p_{i,j} := i * (-1)$.
 - 3.4.3. Für jeden diagonalen Übergang $(i-1, j-1) \rightarrow (i, j)$ zu einem Feld mit gleichen **Kommunikationsbeziehungen** wird eine „Gutschrift“ $s((i-1, j-1), (i, j))$ in Höhe des Gleichheitswertes $d_{i,j} = 0$ der beiden **Kommunikationsbeziehungen** vergeben (vgl. Regel 3.3).
 - 3.4.4. Für jeden diagonalen Übergang $(i-1, j-1) \rightarrow (i, j)$ zu einem Feld mit abweichenden **Kommunikationsbeziehungen** wird eine „Strafe“ $s((i-1, j-1), (i, j))$ in Höhe des Gleichheitswertes $d_{i,j}$ der beiden **Kommunikationsbeziehungen** vergeben (vgl. Regel 3.3).
 - 3.4.5. Für jeden waagerechten Übergang $(i, j-1) \rightarrow (i, j)$ mit $i + j > 1$ und jeden senkrechten Übergang $(i-1, j) \rightarrow (i, j)$ mit $i + j > 1$, d. h. beim Einfügen einer Lücke, wird eine „Strafe“ von -1 vergeben.
 - 3.4.6. Für jeden waagerechten Übergang $(i, j-1) \rightarrow (i, j)$ mit $i = 0 \wedge j = 1$ und jeden senkrechten Übergang $(i-1, j) \rightarrow (i, j)$ mit $i = 1 \wedge j = 0$, d. h. beim Einfügen der ersten Lücke vor der ersten **Kommunikationsbeziehung** einer der betrachteten **Kommunikationsverbindungen**, wird eine „Strafe“ von -2 vergeben.
- 3.5. Für jedes mehrfache Auftreten eines bestimmten Vergleichspaars von aufgerufenen **Bausteinschnittstellen** und für jedes mehrfache Auftreten eines Vergleichspaars von **Operationen** wird keine „Strafe“ von -1 beim Ausrichten der **Kommunikationsverbindungen** und Berechnen des absoluten Gleichheitswertes vergeben².

¹ Da die Abweichungen für Vermittlungsverbindungen separat berechnet werden, genügt zunächst die Tripelnotation. Um die Vermittlungsverbindungen nicht zu vergessen, kann die Verwendung der Quadrupelnotation sinnvoll sein.

² Durch diese und die folgende Regel werden Abweichungen bestimmter **Bausteinschnittstellen** bzw. **Operationen** nur einmal und nicht mehrfach gezählt (siehe die Erläuterungen zu diesen Regeln am Ende dieses

Auszurichtende Kommunikationsverbindungen:

$$\frac{(bss_1, bss_2, op_1, \langle \rangle), (bss_2, bss_3, op_3, \langle \rangle), (bss_3, bss_4, op_4, \langle \rangle), (bss_4, bss_5, op_5, \langle \rangle)}{(bss_1, bss_2, op_2, \langle \rangle), (bss_2, bss_3, op_3, \langle \rangle), (bss_3, bss_6, op_6, \langle \rangle), (bss_6, bss_4, op_7, \langle \rangle), (bss_4, bss_5, op_5, \langle \rangle)}$$

Anwendung des Needleman-Wunsch-Algorithmus:

	$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 0$		(bss_1, bss_2, op_1)	(bss_2, bss_3, op_3)	(bss_3, bss_4, op_4)	(bss_4, bss_5, op_5)
$i = 1$	(bss_1, bss_2, op_2)	-1	-2	-3	-4
$i = 2$	(bss_2, bss_3, op_3)	-2	-1	-2	-3
$i = 3$	(bss_3, bss_6, op_6)	-3	-2	-3	-4
$i = 4$	(bss_6, bss_4, op_7)	-4	-3	-3	-4
$i = 5$	(bss_4, bss_5, op_5)	-5	-4	-4	-3

$$d_{1,1} = -1 = 0 + 0 - 1 = s((0,0), (1,1)) ; p_{1,1} = \boxed{-1} = \max \begin{cases} p_{0,1} + s((0,1), (1,1)) = -1 - 1 = -2 \\ p_{0,0} + s((0,0), (1,1)) = 0 - 1 = -1 \\ p_{1,0} + s((1,0), (1,1)) = -1 - 1 = -2 \end{cases}$$

$$d_{1,2} = -2 = -1 - 1 = s((0,1), (1,2)) ; p_{1,2} = \boxed{-2} = \max \begin{cases} p_{0,2} + s((0,2), (1,2)) = -2 - 1 = -3 \\ p_{0,1} + s((0,1), (1,2)) = -1 - 2 = -3 \\ p_{1,1} + s((1,1), (1,2)) = -1 - 1 = -2 \end{cases}$$

$$d_{1,3} = -2 = -1 - 1 = s((0,2), (1,3)) ; p_{1,3} = \boxed{-3} = \max \begin{cases} p_{0,3} + s((0,3), (1,3)) = -3 - 1 = -4 \\ p_{0,2} + s((0,2), (1,3)) = -3 - 2 = -5 \\ p_{1,2} + s((1,2), (1,3)) = -2 - 1 = -3 \end{cases}$$

$$d_{2,2} = 0 = 0 + 0 = s((1,1), (2,2)) ; p_{2,2} = \boxed{-1} = \max \begin{cases} p_{1,2} + s((1,2), (2,2)) = -2 - 1 = -3 \\ p_{1,1} + s((1,1), (2,2)) = -1 + 0 = -1 \\ p_{2,1} + s((2,1), (2,2)) = -2 - 1 = -3 \end{cases}$$

Ausgerichtete Kommunikationsverbindungen:

$$\frac{(bss_1, bss_2, op_1, \langle \rangle), (bss_2, bss_3, op_3, \langle \rangle), (bss_3, bss_4, op_4, \langle \rangle), (bss_4, bss_5, op_5, \langle \rangle)}{+0 + 0 - 1 \quad +0 + 0 \quad -1 \quad +0 - 1 \quad +0 + 0}$$

$$\frac{(bss_1, bss_2, op_2, \langle \rangle), (bss_2, bss_3, op_3, \langle \rangle), (bss_3, bss_6, op_6, \langle \rangle), (bss_6, bss_4, op_7, \langle \rangle), (bss_4, bss_5, op_5, \langle \rangle)}$$

Der Gleichheitswert der beiden Kommunikationsverbindungen ist -3.

Abbildung 11.1: Ausrichtung von zwei Kommunikationsverbindungen

- 3.6. Existieren beim Ausrichten von Kommunikationsverbindungen Alternativen, d. h. mehrere mögliche Wege durch die Matrix, dann werden die Alternativen bevorzugt, bei denen bestimmte Vergleichspaare mehrfach auftreten.
- 3.7. Für alle Vermittlungsverbindungen der betrachteten Kommunikationsverbindungen wird, getrennt nach der Vermittlungsstufe, der Gleichheitswert separat bestimmt und zum Gleichheitswert der ursprünglich betrachteten Kommunikationsverbindungen addiert³.
- 3.8. Für Vermittlungsverbindungen wird für jede der möglichen Abweichungen die Hälfte des Wertes addiert, der auf der übergeordneten Kommunikationsstufe für eine solche Abweichung addiert würde.

Ende — Definition und Berechnungsvorschrift

Auf der Basis des absoluten Gleichheitswertes wird der absolute Heterogenitätsgrad definiert:

Abschnittes).

³ Sehr oft werden mehr als zwei Vermittlungsverbindungen ausgerichtet werden müssen. Eine allgemeine Vorschrift für die Ausrichtung beliebig vieler Kommunikationsverbindungen und die Bestimmung des zugehörigen Gleichheitswertes wird in Abschnitt 11.2.2 angegeben.

Zu vergleichende Kommunikationsverbindungen:

$$\frac{\begin{pmatrix} (bss_1, bss_2, op_1, \langle (bss_7, bss_8, op_4, \langle (bss_{11}, bss_{12}, op_7, \langle \rangle) \rangle) \rangle) \\ (bss_1, bss_2, op_2, \langle (bss_7, bss_8, op_6, \langle (bss_{11}, bss_{12}, op_7, \langle \rangle) \rangle) \rangle) \end{pmatrix}}{\begin{pmatrix} (bss_1, bss_2, op_1, \langle (bss_7, bss_8, op_4, \langle (bss_{11}, bss_{12}, op_7, \langle \rangle) \rangle) \rangle) \\ (bss_2, bss_3, op_3, \langle \rangle) \end{pmatrix}}$$

Ausrichtung und Vergleich der Kommunikationsbeziehungen:

$$\frac{\begin{pmatrix} (bss_1, bss_2, op_1, \langle \dots \rangle) \\ 0 \quad +0 \quad -1 \end{pmatrix}}{\begin{pmatrix} (bss_1, bss_2, op_2, \langle \dots \rangle), (bss_2, bss_3, op_3, \langle \dots \rangle) \\ -1 \end{pmatrix}} = -2$$

Ausrichtung und Vergleich der Vermittlungsbeziehungen (Stufe 1):

$$\frac{\begin{pmatrix} (bss_7, bss_8, op_4, \langle \dots \rangle) \\ 0 \quad +0 \quad -0,5 \end{pmatrix}}{\begin{pmatrix} (bss_7, bss_8, op_6, \langle \dots \rangle) \\ -0,5 \end{pmatrix}} = -0,5$$

Ausrichtung und Vergleich der Vermittlungsbeziehungen (Stufe 2):

$$\frac{\begin{pmatrix} (bss_{11}, bss_{12}, op_7, \langle \rangle) \\ 0 \quad +0 \quad +0 \end{pmatrix}}{\begin{pmatrix} (bss_{11}, bss_{12}, op_7, \langle \rangle) \\ 0 \end{pmatrix}} = 0$$

Der Gleichheitswert:

$$p(kv_1, kv_2) = -2 - 0,5 + 0 = -2,5$$

Abbildung 11.2: Berechnung des Gleichheitswertes p von zwei ausgerichteten Kommunikationsverbindungen mit Vermittlungsverbindungen

Definition 11.2 Der absolute Heterogenitätsgrad H von zwei Kommunikationsverbindungen ist gleich dem Betrag ihres absoluten Gleichheitswertes:

$$H(kv_1, kv_2) := |p(kv_1, kv_2)| \quad \begin{array}{l} H: \text{ absoluter Heterogenitätsgrad} \\ p: \text{ absoluter Gleichheitswert} \end{array} \quad (11.1)$$

Ende — Definition und Berechnungsvorschrift

Die Festlegung der Bewertung von Übereinstimmungen und Abweichungen

Der Heterogenitätsgrad soll Abweichungen von Kommunikationsverbindungen angeben. Deshalb werden Übereinstimmungen hier nur mit dem Wert 0 angerechnet (Regel 3.3.4). Abweichungen können folglich nicht durch Übereinstimmungen „aufgehoben“ werden. Damit ist der Gleichheitswert stets kleiner oder gleich 0. Der größtmögliche Betrag für die Abweichung von zwei Kommunikationsbeziehungen ist $2 = |-2|$: Abweichung der aufgerufenen Bausteinschnittstellen plus Abweichung der Operationen (Regeln 3.3.2 und Regel 3.3.5). Eine Ausnahme bilden die jeweils ersten Kommunikationsbeziehungen der betrachteten Kommunikationsverbindungen. Der größtmögliche Betrag für die Abweichung dieser Kommunikationsbeziehungen ist $3 = |-3|$: Abweichung der aufrufenden Bausteinschnittstellen plus Abweichung

der aufgerufenen Bausteinschnittstellen plus Abweichung der Operationen (Regeln 3.3.3 und 3.3.5).

Die Festlegung des Wertes -1 als Basiswert für eine Abweichung einer Bausteinschnittstelle oder einer Operation ist zunächst willkürlich (Regel 3.3.5). Der Basiswert könnte durch einen beliebigen anderen (negativen) Wert ersetzt werden. Die Festlegung der übrigen Werte für Abweichungen muss dann im selben Verhältnis erfolgen. Dann entstehen andere absolute Beträge für die Gleichheitswerte und die Heterogenitätsgrade, aber die Ordnungsrelationen „größer als“ ($>$) und „kleiner als“ ($<$) bleiben gleich.

Der Wert -1 für das Einfügen einer Lücke ist gleich der Hälfte der größtmöglichen Abweichung von zwei Kommunikationsbeziehungen (Regel 3.4.5). Damit ist es für die Berechnung des Heterogenitätsgrades unerheblich, ob zwei vollständig unterschiedliche Kommunikationsbeziehungen bei der Ausrichtung übereinandergelegt oder durch Einfügen von Lücken gegeneinander verschoben werden:

$$\frac{(\dots)(bss_2, bss_3, op_2)(\dots)}{-1 - 1} = \frac{(\dots)(bss_2, bss_3, op_2)(\dots)}{-1} \frac{(\dots)}{(bss_5, bss_6, op_4)(\dots)}$$

Diese Flexibilität ist insbesondere beim Ausrichten von mehr als zwei Kommunikationsverbindungen nützlich (siehe Abschnitt 11.2.2). Dabei können u. U. günstigere Ausrichtungen entstehen, wenn bestimmte vollständig abweichende Kommunikationsbeziehungen gegeneinander verschoben werden. Dabei ändert sich der Gleichheitswert der Kommunikationsverbindungen mit den gegeneinander verschobenen Kommunikationsbeziehungen nicht.

Durch die zusätzlichen Regeln für die Vergleiche der jeweils ersten Kommunikationsbeziehungen der betrachteten Kommunikationsverbindungen wird erreicht, dass auch die aufrufenen Bausteinschnittstellen dieser Kommunikationsbeziehungen in den Vergleich einbezogen werden (Regel 3.3.3). Bei den darauf folgenden Kommunikationsbeziehungen ist die aufrufende Bausteinschnittstelle zugleich die aufgerufene Bausteinschnittstelle der vorhergehenden Kommunikationsbeziehung und sollte folglich nicht „doppelt“ verrechnet werden. Deshalb werden, mit Ausnahme der Vergleiche mit Beteiligung einer ersten Kommunikationsbeziehung, immer nur die aufgerufenen Bausteinschnittstellen und die Operationen miteinander verglichen.

Durch die besondere „Strafe“ für die erste Lücke vor der jeweils ersten Kommunikationsbeziehung einer Kommunikationsverbindung (Regel 3.4.6) wird auch für die ersten Kommunikationsbeziehungen erreicht, dass sie bei vollständiger Abweichung bei der Ausrichtung übereinandergelegt oder durch Einfügen von Lücken gegeneinander verschoben werden können:

$$\frac{(bss_1, bss_2, op_1)(\dots)}{-1 - 1 - 1} = \frac{(bss_1, bss_2, op_1)(\dots)}{-2} \frac{(\dots)}{(bss_3, bss_4, op_2)(\dots)}$$

Die Regeln für die Berücksichtigung von Zyklen in Kommunikationsverbindungen

Die vorgestellte Berechnungsvorschrift für den absoluten Gleichheitswert kann zu wenig sinnvollen Ergebnissen führen, wenn die Regeln 3.5 und 3.6 weggelassen werden. Problematisch sind dann solche Szenarios, bei denen innerhalb einer Kommunikationsverbindung bestimmte

(bss_1, bss_2, op_1)	(bss_2, bss_3, op_2)	(bss_3, bss_4, op_3)	(bss_4, bss_5, op_4)	
	(bss_1, bss_3, op_5)	(bss_3, bss_4, op_6)	(bss_4, bss_6, op_7)	
(bss_7, bss_2, op_8)	(bss_2, bss_3, op_9)	(bss_3, bss_6, op_7)		
-2	-1 + 0 - 1	+ 0 - 1	- 1 - 1	
-1 + 0 - 1	+ 0 - 1	- 1	- 1 - 1	
-2	-1 + 0 - 1	- 1	+ 0 + 0	
-6	-5	-3	-4	= -18

Abbildung 11.3: Berechnung des Gleichheitswertes p von drei Kommunikationsverbindungen

Operationsaufrufe mehrfach auftreten, d. h. wenn Kommunikationsverbindungen Zyklen enthalten. Zyklen entstehen, wenn eine Bausteinschnittstelle mehrfach als aufgerufene Bausteinschnittstelle auftritt, u. U. mit derselben aufgerufenen Operation.

Diese Form der Wiederverwendung wird beim Ausrichten und Berechnen des Gleichheitswertes durch die beiden genannten Regeln berücksichtigt. Ohne sie könnten ungeeignet hohe Abweichungen, d. h. ungeeignet niedrige Gleichheitswerte, berechnet werden.

11.2.2 Der absolute Heterogenitätsgrad von n Kommunikationsverbindungen

Der Heterogenitätsgrad kann auch für mehr als zwei Kommunikationsverbindungen bestimmt werden. Die Definition und Berechnungsvorschrift für den Gleichheitswert p aus Abschnitt 11.2.1 wird dazu folgendermaßen erweitert:

Definition und Berechnungsvorschrift

Definition 11.3 Der absolute Gleichheitswert p von $n > 1$ Kommunikationsverbindungen ist ein Maß für die Abweichung der Kommunikationsverbindungen hinsichtlich der zugehörigen Bausteinschnittstellen und der aufgerufenen Operationen.

Für die Berechnung von p für n Kommunikationsverbindungen werden die Regeln aus Abschnitt 11.2.1 durch folgende Ergänzungen allgemeiner gefasst:

- 1*. Gleichheitswerte werden jeweils für Vektoren von n Kommunikationsbeziehungen berechnet. Der Gleichheitswert von n Kommunikationsbeziehungen ist gleich der Summe der $n - 1$ paarweisen Gleichheitswerte.
- 2*. Der Gleichheitswert von zwei Lücken ist 0.
- 3*. Die Ausrichtung der Kommunikationsverbindungen erfolgt auf der Basis eines n -dimensionalen Würfels. Die Bewertung der Übergänge zwischen den benachbarten Würfелеlementen erfolgt, wie beim Verfahren für zwei Kommunikationsverbindungen, auf der Basis der berechneten Abweichungen.
- 4*. Für die abschließende Ausrichtung entsprechend des günstigsten Pfades durch den n -dimensionalen Würfel gilt: Bei allen Kommunikationsverbindungen, deren Elemente bei einem Übergang zwischen benachbarten Würfелеlementen beibehalten werden, wird eine Lücke eingefügt.

Ende — Definition und Berechnungsvorschrift

Abbildung 11.3 zeigt ein Beispiel für die Ausrichtung von drei Kommunikationsverbindungen und die Berechnung des zugehörigen Gleichheitswertes.

Auch die Definition des Heterogenitätsgrades H für n Kommunikationsverbindungen erfolgt auf der Basis des absoluten Gleichheitswertes:

Definition 11.4 Der **absolute Heterogenitätsgrad** H von n Kommunikationsverbindungen ist gleich dem Betrag ihres absoluten Gleichheitswertes:

$$H(kv_1, \dots, kv_n) := |p(kv_1, \dots, kv_n)| \quad \begin{array}{l} H: \text{ absoluter Heterogenitätsgrad} \\ p: \text{ absoluter Gleichheitswert} \end{array} \quad (11.2)$$

Ende — Definition und Berechnungsvorschrift

11.2.3 Interpretation des absoluten Heterogenitätsgrades

Der absolute Heterogenitätsgrad H ist stets eine ganze Zahl⁴ größer oder gleich 0. Sein Betrag kann zunächst nicht als intuitiv verständliche Kenngröße interpretiert werden. Er kann jedoch verwendet werden, um verschiedene alternative Architekturen hinsichtlich ihrer Heterogenität zu vergleichen. Entsprechend der Grundannahme 2 aus Kapitel 1 ist ein geringerer Heterogenitätsgrad zu bevorzugen.

Der in Abschnitt 11.3 definierte relative Heterogenitätsgrad und der in Abschnitt 11.4 definierte kostenbewertete Heterogenitätsgrad können intuitiver interpretiert werden.

11.2.4 Problematik der Anwendung des Heterogenitätsgrades

Die Bewertung von Abweichungen von Operationen

Nach Regel 3.3.5 wird jede Abweichung von Bausteinschnittstellen und jede Abweichung von Operationen mit einer „Strafe“ von -1 belegt. Nach der in Abschnitt 7.2.2 definierten Erweiterung des 3LGM² hinsichtlich der Operationen kann eine Operation immer nur einer einzigen Bausteinschnittstelle zugeordnet werden. Jede Abweichung von Bausteinschnittstellen ist also immer mit einer Abweichung der Operationen verbunden.

Eine Abweichung von Operationen ohne Abweichung der Bausteinschnittstellen wird durch die genannte Regel mit der Hälfte der „Strafe“ für eine vollständige Abweichung der Bausteinschnittstellen und der Operationen bewertet. In manchen Szenarios kann eine geringere „Strafe“ für Abweichungen der Operationen sinnvoll sein. Diese Problem wird hier nicht weiter behandelt.

Die Berücksichtigung gleicher Bausteinschnittstellen

Mit der vorgestellten Berechnungsvorschrift für den Heterogenitätsgrad kann zunächst nicht berücksichtigt werden, ob bestimmte verschiedene Bausteinschnittstellen prinzipiell gleich sind, z. B. durch mehrfache Installation. Dieser Sachverhalt kann auch als Wiederverwendung betrachtet werden. Er ist insbesondere bei Informationssystemmodellen, die viele gleiche Komponenten beinhalten, relevant.

⁴ Bei anderen Beträgen für die „Strafen“ bei Abweichungen von Bausteinschnittstellen oder Operationen kann er zu einer reellen Zahl (größer oder gleich 0) werden.

11.3 Der relative Heterogenitätsgrad von Kommunikationsverbindungen

Wenn das 3LGM² um eine Gleichheitsrelation für Bausteinschnittstellen erweitert wird, können gleiche Installationen von Bausteinschnittstellen beim Bestimmen des Heterogenitätsgrades so behandelt werden, als wäre nur eine einzige Installation vorhanden.

Die Abhängigkeit des Heterogenitätsgrades vom Detaillierungsgrad

Bei der Anwendung des Heterogenitätsgrades können für ein Informationssystem leicht mehrere unterschiedliche Bewertungen entstehen. Der Wert des Heterogenitätsgrades hängt sehr stark davon ab, wie fein Kommunikationsverbindungen modelliert wurden. Die detaillierten Kenntnisse einer bestimmten Kommunikationsverbindung im Vergleich zu möglicherweise nur groben Kenntnissen einer anderen Kommunikationsverbindung kann zu unterschiedlich detaillierter Modellierung führen. Der Heterogenitätsgrad für eine Architektur mit der detaillierter modellierten Kommunikationsverbindung kann dann allein aufgrund der höheren Detaillierung höher sein als der Heterogenitätsgrad für eine Architektur mit Nutzung der nur grob modellierten Kommunikationsverbindung.

Der in den „Grundsätzen ordnungsgemäßer Modellierung“ beschriebene Grundsatz der Vergleichbarkeit ([BECKER et al. 2000], S. 16) impliziert die Berücksichtigung des Detaillierungsgrades bei der Modellierung. Ein allgemeines Vorgehen zur Einhaltung eines bestimmten Detaillierungsgrades wurde bisher jedoch nicht formuliert. Die Angabe einer entsprechenden Vorschrift erfolgt auch in dieser Arbeit nicht.

Der Abschnitt 11.4 beschreibt, wie die Berücksichtigung von Kosten u. a. das Problem der unterschiedlichen Detaillierung beseitigen oder abschwächen kann.

11.3 Der relative Heterogenitätsgrad von Kommunikationsverbindungen

11.3.1 Der relative Heterogenitätsgrad

Der relative Heterogenitätsgrad ist eine Grundlage für weiterführende Berechnungen auf der Basis der bisherigen Definitionen für den Heterogenitätsgrad. Er wird in Abschnitt 11.4 für die Definition des kostenbewerteten Heterogenitätsgrades verwendet.

Zur Berechnung des relativen Heterogenitätsgrades wird hier zunächst der relative Gleichheitswert definiert. Für die Berechnung des relativen Gleichheitswertes werden die bisher definierten Regeln zur Berechnung des absoluten Gleichheitswertes angepasst.

Definition und Berechnungsvorschrift

Definition 11.5 Der **relative Gleichheitswert** pr von $n > 1$ Kommunikationsverbindungen ist die Summe der gemittelten Abweichungen der einzelnen Vektoren von Kommunikationsbeziehungen nach Ausrichtung der Kommunikationsverbindungen.

Für die Berechnung von pr für n Kommunikationsverbindungen werden die Regeln aus den Abschnitten 11.2.1 und 11.2.2 durch folgende Ergänzungen angepasst:

- 1#. Für die Berechnung des relativen Gleichheitswertes eines einzelnen Vektors von Kommunikationsbeziehungen werden die $\binom{n}{2}$ paarweisen Gleichheitswerte addiert; die Sum-

me wird durch die Anzahl der Paare dividiert und mit $n - 1$ multipliziert:

$$pr(kb_1, \dots, kb_n) := \frac{(p_1, \dots, p_{\binom{n}{2}})}{\binom{n}{2}} * (n - 1) \quad (11.3)$$

2[#]. Die Summe der relativen Gleichheitswerte der einzelnen Vektoren ergibt den relativen Gleichheitswert der betrachteten Kommunikationsverbindungen.

Ende — Definition und Berechnungsvorschrift

Die Definition des relativen Heterogenitätsgrades erfolgt analog zur Definition des absoluten Heterogenitätsgrades auf der Basis des relativen Gleichheitswertes:

Definition 11.6 Der **relative Heterogenitätsgrad** HR von n Kommunikationsverbindungen ist gleich dem Betrag ihres relativen Gleichheitswertes:

$$HR(kv_1, \dots, kv_n) := |pr(kv_1, \dots, kv_n)| \quad \begin{array}{l} H: \text{ relativer Heterogenitätsgrad} \\ p: \text{ relativer Gleichheitswert} \end{array} \quad (11.4)$$

Ende — Definition und Berechnungsvorschrift

11.3.2 Interpretation des relativen Heterogenitätsgrades

Der relative Heterogenitätsgrad HR ist stets eine ganze Zahl⁵ größer oder gleich 0. Mit der gewählten „Strafe“ von -1 für jede Abweichung von **Bausteinschnittstellen** oder **Operationen** kann er als Schätzfunktion für die Anzahl der **Bausteinschnittstellen** und der Implementierungen von **Operationen** verwendet werden, die möglicherweise durch andere bereits im Informationssystem vorhandene **Bausteinschnittstellen** und zugehörige Implementierungen von **Operationen** ersetzt werden können.

Die Bedeutung der Berechnung für einen einzelnen Vektor von **Kommunikationsbeziehungen** kann folgendermaßen erklärt werden:

- 1 Der Wert $\binom{n}{2}$ ist die Anzahl der Paare von **Kommunikationsbeziehungen**. Die Division der einzelnen absoluten Gleichheitswerte der Paare ergibt einen durchschnittlichen absoluten Gleichheitswert.
- 2 Ausgehend von der Annahme, dass die bei der Heterogenitätsgradberechnung betrachteten Kommunikationsverbindungen demselben Zweck dienen, kann formuliert werden: „Alle **Kommunikationsbeziehungen** können durch eine einzige ersetzt werden.“
- 3 Wenn sich alle **Kommunikationsbeziehungen** paarweise vollständig unterscheiden, können theoretisch $(n - 1)$ **Bausteinschnittstellen** und $(n - 1)$ **Operationen** weggelassen werden. Das entspricht einem absoluten Gleichheitswert von $-2 * (n - 1)$; dabei ist -2 der minimale absolute Gleichheitswert von zwei **Kommunikationsbeziehungen**⁶.
- 4 Durch die Multiplikation von $n - 1$ mit dem durchschnittlichen absoluten Gleichheitswert (anstelle des maximalen) werden Wiederverwendungen von **Operationen** und **Bausteinschnittstellen** berücksichtigt.

⁵ Bei anderen Beträgen für die „Strafen“ bei Abweichungen von **Bausteinschnittstellen** oder **Operationen** kann er zu einer reellen Zahl (größer oder gleich 0) werden.

⁶ Bei den ersten **Kommunikationsbeziehungen** der Kommunikationsverbindungen ist entsprechend -3 als minimaler paarweiser absoluter Gleichheitswert zu verrechnen.

11.4 Der kostenbewertete Heterogenitätsgrad von Kommunikationsverbindungen

11.4.1 Kosten von Kommunikationsverbindungen

Kosten sind das wohl am meisten verwendete Bewertungskriterium für viele Sachverhalte. Auch bei der Bewertung von Informationssystemen werden Kosten oft in den Mittelpunkt gestellt.

Um Kommunikationsverbindungen hinsichtlich ihrer Kosten vergleichen zu können, wird hier eine weitere Ergänzung der in den Abschnitten 8.6.1 bis 8.6.2 definierten Tupelnotation von **Kommunikationsbeziehungen** vorgenommen. Eine **Kommunikationsbeziehung** kann also als Quintupel $(bss_1, bss_2, op, \{vb_1, \dots, vb_n\}, k)$ aufgefasst werden:

$$kb = (bss_1, bss_2, op, \langle vb_1, \dots, vb_n \rangle, k(kb)) \quad \left| \begin{array}{l} bss_1 \quad := \text{aufrufende Bausteinschnittstelle} \\ bss_2 \quad := \text{aufgerufene Bausteinschnittstelle} \\ op \quad \quad := \text{aufgerufene Operation} \\ \langle vb_1, \dots, vb_n \rangle := \text{Vermittlungsverbindung als Folge der zugehörigen Kommunikationsbeziehungen} \\ k(kb) \quad := \text{Kosten von kb} \end{array} \right. \quad (11.5)$$

Für eine bessere Übersichtlichkeit kann, falls die Vermittlungsverbindungen nicht betrachtet werden sollen, eine verkürzte Notation ohne die Vermittlungsverbindung gewählt werden:

$$kb = (bss_1, bss_2, op, k(kb)) \quad (11.6)$$

Die Kosten einer Kommunikationsverbindung sind gleich der Summe der Kosten der zugehörigen **Kommunikationsbeziehungen**:

$$K(kv) := \sum_{i=0 \dots L(kv)} k(kb_i) \quad (11.7)$$

11.4.2 Der kostenbewertete Heterogenitätsgrad

Der in diesem Abschnitt vorgestellte kostenbewertete Heterogenitätsgrad für Kommunikationsverbindungen kann verwendet werden, um die Wiederverwendung von Integrationstechniken unter Berücksichtigung der mit ihnen verbundenen Kosten zu bewerten.

Für die Definition des kostenbewerteten Heterogenitätsgrades wird zunächst der kostenbewertete Gleichheitswert eingeführt.

Definition und Berechnungsvorschrift

Definition 11.7 Der **kostenbewertete Gleichheitswert** pk von $n > 1$ Kommunikationsverbindungen ist ein Maß für die Abweichung der Kommunikationsverbindungen hinsichtlich der zugehörigen **Bausteinschnittstellen** und der aufgerufenen **Operationen** unter Berücksichtigung der Kosten der einzelnen **Kommunikationsbeziehungen**.

Für die Berechnung von pk für n Kommunikationsverbindungen werden die Regeln aus den Abschnitten 11.2.1, 11.2.2 und 11.3 durch folgende Ergänzungen erweitert und geändert:

- 1⁺. Der kostenbewertete Gleichheitswert wird prinzipiell wie der relative Gleichheitswert bestimmt (vgl. Abschnitt 11.3). Für die Berechnung der kostenbewerteten Gleichheits-

11 Heterogenitätsbewertung von Kommunikationsverbindungen

werte der einzelnen Vektoren von **Kommunikationsbeziehungen** werden die in den folgenden Regeln angegebenen Anpassungen angewendet.

- 2⁺. Jeder absolute Gleichheitswert $p(kb_i, kb_j)$ für ein Paar von **Kommunikationsbeziehungen** wird durch Division durch den theoretischen maximalen absoluten Gleichheitswert p_{tmax} für ein Paar von **Kommunikationsbeziehungen** standardisiert:

$$ps(kb_i, kb_j) := \frac{p(kb_i, kb_j)}{p_{tmax}} \quad (11.8)$$

Mit den standardisierten Gleichheitswerten für die einzelnen Paare wird der standardbasierte relative Gleichheitswert für jeden Vektor von **Kommunikationsbeziehungen** berechnet:

$$psr(kb_1, \dots, kb_n) := \frac{(ps_1, \dots, ps_{\binom{n}{2}})}{\binom{n}{2}} * (n - 1) \quad (11.9)$$

- 3⁺. Jeder der standardbasierten relativen Gleichheitswerte für einen Vektor von **Kommunikationsbeziehungen** wird mit den durchschnittlichen Kosten der zugehörigen **Kommunikationsbeziehungen** multipliziert. Das Ergebnis ist der kostenbewertete Gleichheitswert für den betreffenden Vektor von **Kommunikationsbeziehungen**:

$$pk(kb_1, \dots, kb_n) := psr(kb_1, \dots, kb_n) * \frac{\sum_{i=1 \dots n} k(kb_i)}{n} \quad (11.10)$$

- 4⁺. Die Summe der kostenbewerteten Gleichheitswerte der einzelnen Vektoren ergibt den kostenbewerteten Gleichheitswert der betrachteten **Kommunikationsverbindungen**.

Ende — Definition und Berechnungsvorschrift

Der kostenbewertete Heterogenitätsgrad von **Kommunikationsverbindungen** wird analog zum absoluten und zum relativen Heterogenitätsgrad definiert:

Definition 11.8 Der **kostenbewertete Heterogenitätsgrad HK** von n **Kommunikationsverbindungen** ist gleich dem Betrag ihres kostenbewerteten Gleichheitswertes:

$$HK_{kv1, \dots, kvn} := |pk_{kv1, \dots, kvn}| \quad \begin{array}{l} HK: \text{ kostenbewerteter Heterogenitätsgrad} \\ pk : \text{ kostenbewerteter Gleichheitswert} \end{array} \quad (11.11)$$

Ende — Definition und Berechnungsvorschrift

11.4.3 Interpretation des kostenbewerteten Heterogenitätsgrades

Der kostenbewertete Heterogenitätsgrad HK ist stets eine Kostenangabe. Mit der in Regel 2⁺ vorgeschriebenen Standardisierung der Gleichheitswerte für Paare von **Kommunikationsbeziehungen** kann er als Schätzfunktion für die Summe der Kosten der **Bausteinschnittstellen** und der Implementierungen von **Operationen** verwendet werden, die möglicherweise durch andere bereits im Informationssystem vorhandene **Bausteinschnittstellen** und zugehörige Implementierungen von **Operationen** ersetzt werden können.

Die Bedeutung der Berechnung für einen einzelnen Vektor von **Kommunikationsbeziehungen** kann, ähnlich der Berechnung des relativen Heterogenitätsgrades, folgendermaßen erklärt werden:

- 1 Die Standardisierung der absoluten Gleichheitswerte für die einzelnen Paare von **Kommunikationsbeziehungen** wird wegen der Annahme vorgenommen, dass die Kosten i. d. R. zwar für einzelne **Bausteinschnittstellen**, aber nicht für einzelne **Operationen** bekannt sind. Eine vollständige Abweichung von zwei **Kommunikationsbeziehungen** (standardisierter absoluter Gleichheitswert $psr = 2$) entspricht also einem einmaligen zusätzlichen Aufkommen der Schnittstellenkosten für die zweite **Bausteinschnittstelle**. Eine Abweichung nur bezüglich der **Operationen** entspricht einem nur anteiligen Aufkommen der Schnittstellenkosten, die für die zusätzliche Implementierung der zweiten **Operation** entstehen.
- 2 Der Wert $\binom{n}{2}$ ist die Anzahl der Paare von **Kommunikationsbeziehungen**. Die Division der einzelnen standardisierten absoluten Gleichheitswerte durch die Anzahl der Paare ergibt einen durchschnittlichen standardisierten absoluten Gleichheitswert.
- 3 Ausgehend von der Annahme, dass die bei der Heterogenitätsgradberechnung betrachteten Kommunikationsverbindungen demselben Zweck dienen, kann formuliert werden: „Alle **Kommunikationsbeziehungen** können durch eine einzige ersetzt werden.“
- 4 Wenn sich alle **Kommunikationsbeziehungen** paarweise vollständig unterscheiden, können theoretisch $(n - 1)$ **Bausteinschnittstellen** und $(n - 1)$ **Operationen** weggelassen werden. Das entspricht einem Gleichheitswert von $-1 * (n - 1)$; dabei ist -1 der minimale standardisierte absolute Gleichheitswert von zwei **Kommunikationsbeziehungen**⁷.
- 5 Durch die Multiplikation von $n - 1$ mit dem durchschnittlichen Gleichheitswert (anstelle des maximalen) werden Wiederverwendungen von **Operationen** und **Bausteinschnittstellen** berücksichtigt.
- 6 Die jeweilige abschließende Multiplikation der durchschnittlichen Kosten für eine **Kommunikationsbeziehung** des betrachteten Vektors führt zum kostenbewerteten Heterogenitätsgrad. Jede vollständige Abweichung von zwei **Kommunikationsbeziehungen** führt dabei zu einem vollständigen Anrechnen (mit Faktor 1) der durchschnittlichen Kosten. Jede Abweichung nur bezüglich der **Operation** führt zu einem anteiligen Anrechnen.

11.4.4 Bemerkungen zum kostenbewerteten Heterogenitätsgrad

Bis auf das Problem der Abhängigkeit von der Detaillierung der Modellierung gelten alle in Abschnitt 11.2.4 zum absoluten Heterogenitätsgrad genannten Probleme auch für den kostenbewerteten Heterogenitätsgrad. Insbesondere die Bewertung der Abweichung bezüglich der **Operation** mit der Hälfte der Abweichung bezüglich der **Bausteinschnittstelle** und der **Operation** kann bei **Bausteinschnittstellen** mit vielen **Operationen** zu schlecht interpretierbaren Ergebnissen führen.

Die Unabhängigkeit von der Detaillierung der Modellierung

Der kostenbewertete Heterogenitätsgrad ist, mit der folgenden Voraussetzung, weniger „anfällig“ für unterschiedliche Detaillierung bei den verglichenen Kommunikationsverbindungen:

⁷ Bei den ersten **Kommunikationsbeziehungen** der Kommunikationsverbindungen ist entsprechend -3 als minimaler paarweiser Gleichheitswert zu verrechnen.

Hier wird vorausgesetzt, dass eine detailliertere Modellierung zu einer vergrößerten Anzahl von Bausteinschnittstellen führt, die jedoch jeweils geringere Kosten verursachen. Die Summe der Kosten bei detaillierterer Modellierung sollte prinzipiell gleich der Summe der Kosten bei weniger detaillierter Modellierung sein. Damit wird eine vergrößerte Anzahl von Bausteinschnittstellen durch geringere Einzelkosten ausgeglichen.

11.5 Ein Anwendungsszenario aus dem Universitätsklinikum Leipzig

Die Anwendung der in diesem Kapitel vorgestellten Kennzahlberechnungen wird hier an dem bereits in Abschnitt 10.4 vorgestellten Beispielszenario aus dem Universitätsklinikum Leipzig (UKL) demonstriert. Auch hier werden die alte und die geplante Architektur hinsichtlich der Kommunikation von Falldaten analysiert und verglichen.

11.5.1 Ermittlung der Kommunikationsverbindungen als Folgen von Kommunikationsbeziehungen

In Abbildung 11.4 wurden die Kommunikationsbeziehungen der bereits in Abschnitt 10.4 betrachteten Kommunikationsverbindungen mit Tripelnotationen versehen. Diese Notierungen werden bei der Ausrichtung der Kommunikationsverbindungen verwendet.

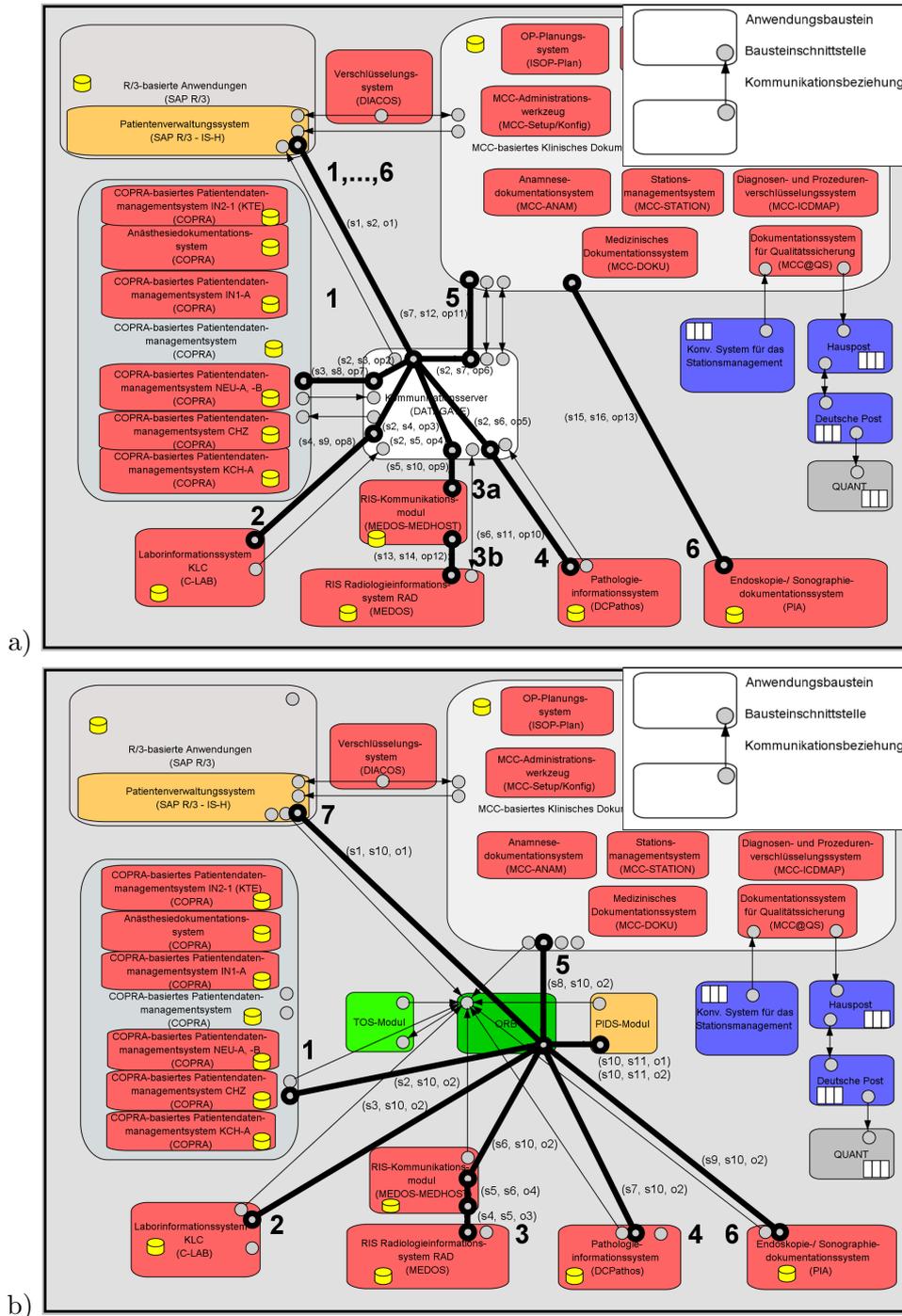
Die Bausteinschnittstellen des Kommunikationsservers (Abbildung 11.4a) wurden detaillierter modelliert als die des ORB (Abbildung 11.4b). Wie bereits in Abschnitt 10.4 beschrieben, wurden die IDL-Stubs und Skeletons, die für die Kommunikation über den ORB verwendet werden, für den TOS-Modul und auch für den PIDS-Modul zu jeweils einer Bausteinschnittstelle zusammengefasst.

In Abschnitt 11.2.4 wurde erläutert, dass ein unterschiedlicher Detaillierungsgrad u. U. zu schwer interpretierbaren Ergebnissen beim Vergleich von Architekturen führen kann. Die im hier betrachteten Beispiel gegebenen Unterschiede bei der Modellierung können jedoch durch folgende Argumentation gerechtfertigt werden: Bei einer CORBA-basierten Architektur werden IDL-Stubs und Skeletons i. d. R. gleichzeitig bei der Implementierung generiert. Sie sind durch die Schnittstellenspezifikation des über sie bereitgestellten Dienstes unmittelbar miteinander verknüpft bzw. voneinander abhängig. Dadurch ist es auch schwer, Schnittstellenkosten getrennt für IDL-Stubs und Skeletons anzugeben. Im Gegensatz dazu werden bei einer Kommunikationsserver-basierten Architektur die einzelnen Bausteinschnittstellen des Kommunikationsservers oft getrennt konfiguriert und verursachen dadurch jeweils eigene Konfigurations- und Wartungskosten. Die getrennten Kosten entstehen z. B. bei der speziellen Einrichtung des Nachrichtentransfers von und zu den einzelnen Partnersystemen aber auch bei der internen Konfiguration. Sehr oft werden Nachrichten, die am Kommunikationsserver eingehen, vor der Weiterversendung gefiltert und transformiert. Die Filter und die Transformationen hängen dabei von den einzelnen Anwendungsbausteinen ab, die die Nachrichten empfangen sollen.

11.5.2 Kennzahlen zur Heterogenität für das Anwendungsszenario

Die Tripelnotationen für die einzelnen Kommunikationsbeziehungen wurden um zusätzliche vierte Elemente mit Kostenangaben ergänzt und zu Kommunikationsverbindungen zusammengesetzt.

11.5 Ein Anwendungsszenario aus dem Universitätsklinikum Leipzig



Bei a) sind die Endpunkte der Kommunikationsverbindungen vom Patientenverwaltungssystem zu den „Ziel“-Anwendungsbausteinen der Falldaten nummeriert. Bei b) sind die Startpunkte der Kommunikationsverbindungen zum PIDS-Modul nummeriert. Die Kommunikationsbeziehungen sind mit Tripelnotationen entsprechend den Ausführungen in Abschnitt 8.6.1 versehen.

Abbildung 11.4: Anwendungsszenario aus dem UKL mit Kommunikationsserver (a) und mit ORB (b); hervorgehoben sind die Kommunikationsverbindungen zur Verteilung der Falldaten

11 Heterogenitätsbewertung von Kommunikationsverbindungen

$(s_1, s_2, o_1, 1000€)$	$(s_2, s_3, o_2, 200€)$	$(s_3, s_8, o_7, 2500€)$		
$(s_1, s_2, o_1, 1000€)$	$(s_2, s_4, o_3, 200€)$	$(s_4, s_9, o_8, 1800€)$		
$(s_1, s_2, o_1, 1000€)$	$(s_2, s_5, o_4, 250€)$	$(s_5, s_{10}, o_9, 1900€)$		
$(s_1, s_2, o_1, 1000€)$	$(s_2, s_6, o_5, 300€)$	$(s_6, s_{11}, o_{10}, 3000€)$		
$(s_1, s_2, o_1, 1000€)$	$(s_2, s_7, o_6, 200€)$	$(s_7, s_{12}, o_{11}, 2500€)$		
$(s_{13}, s_{14}, o_{12}, 800€)$				
$(s_{15}, s_{16}, o_{13}, 900€)$				
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
-33	-30	-30		= -93 = p
- 9,43	- 8,57	- 8,57		= -26,57 = pr
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
-3005,43€	-938,07€	-9543,86€		= -13487,36€ = pk

a)

		$(s_1, s_{10}, o_1, 3000€)$	$(s_{10}, s_{11}, o_1, 3000€)$	
		$(s_2, s_{10}, o_2, 2000€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
		$(s_3, s_{10}, o_2, 1500€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
$(s_4, s_5, o_3, 800€)$	$(s_5, s_6, o_4, 1000€)$	$(s_6, s_{10}, o_2, 1000€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
		$(s_7, s_{10}, o_2, 2000€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
		$(s_8, s_{10}, o_2, 1800€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
		$(s_9, s_{10}, o_2, 2000€)$	$(s_{10}, s_{11}, o_2, 3000€)$	
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
-18	-12	-27	- 6	= -63 = p
- 5,14	- 3,43	- 7,71	- 1,71	= -17,99 = pr
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>
-195,43€	-244,29€	-4883,00€	-2580,00€	= -7902,72€ = pk

b)

Abbildung 11.5: Ausgerichtete Kommunikationsverbindungen des Anwendungsszenarios; Berechnung der absoluten und der relativen Gleichheitswerte

Abbildung 11.5 zeigt die Ausrichtungen der Kommunikationsverbindungen und die Ergebnisse der Berechnungen der Gleichheitswerte. Aus den Gleichheitswerten können die Heterogenitätsgrade unmittelbar ermittelt werden. Für die alte Architektur mit Kommunikationsserver gilt bezüglich der hier betrachteten Kommunikationsverbindungen:

- $H = 93$,
- $HR = 26,57$,
- $HK = 13.487,36€$

Für die neue Architektur mit ORB gilt bezüglich der hier betrachteten Kommunikationsverbindungen:

- $H = 63$,
- $HR = 17,99$,
- $HK = 7.902,72€$

Z Abschluss

Z.1 Vorbemerkungen

In den Kapiteln zwischen der Einleitung und dem hier beginnenden Abschluss wurde das Thema der Informationssystemarchitektur aus verschiedenen, jedoch nicht unabhängigen, Blickwinkeln betrachtet. Im Teil I „Grundlagen“ wurden Grundlagen zu Integrationsanforderungen, Architekturstandards und Methoden der Geschäftsprozessmodellierung erarbeitet. In Teil II „Architekturmodellierung“ wurde auf der Basis der Grundlagen ein Metamodell für die Modellierung von Informationssystemen, das 3LGM² zum 3LGM_A² überarbeitet. Für das 3LGM_A² wurden in Teil III „Architekturbewertung“ Ansätze zur Bewertung von Informationssystemarchitekturen hinsichtlich der Integration entworfen.

Dieses abschließende Kapitel fasst die Ergebnisse der vorliegenden Arbeit mit Bezug zu den in der Einleitung genannten Zielen und Fragen zusammen.

Z.2 Beantwortung der Fragen

In der Einleitung wurden zu den folgenden Zielen Fragen gestellt:

- Ziel Z1** Es soll dargestellt werden, welche Anforderungen bzgl. der Integration von Informationssystemkomponenten typischerweise in Krankenhäusern gestellt werden und welche Techniken zu ihrer Erfüllung verwendet werden können.
- Ziel Z2** Es soll eine vergleichende Übersicht über Architekturstandards erstellt werden.
- Ziel Z3** Es soll eine Überarbeitung des 3LGM² entworfen werden, welche die Modellierung der Erfüllung von Integrationsanforderungen und die Bewertung der Verwendung unterschiedlicher Integrationstechniken erlaubt. Dabei sollen vorhandene Architekturstandards berücksichtigt werden.
- Ziel Z4** Es soll mit Hilfe des überarbeiteten 3LGM² untersucht werden, ob verschiedene Integrationstechniken ausgetauscht werden können, ohne dass dabei Einbußen an Funktionalität entstehen.

Z.2.1 Fragen zu Ziel Z1

F1.1

Welche Anforderungen werden an die Integration von Informationssystemkomponenten gestellt?

Im Anwendungsbereich des Gesundheitswesens können sehr viele einzelne Integrationsanforderungen gestellt werden. Sie hängen davon ab, welche Aufgaben unter Nutzung und Bearbeitung welcher Informationen in welchem Zusammenhang zu erledigen sind und welche Werkzeuge dafür zur Verfügung stehen.

In Kapitel 2 wurden zur Einteilung der Anforderungen die Anforderungskategorien

- physische Integration,
- Datenintegration,
- funktionale Integration,
- semantische Integration,
- Kontextintegration,
- Präsentationsintegration und
- Zugriffsintegration.

erarbeitet.

Kapitel 9 beschreibt mit dem Domänenkonzept einen Ansatz zur Überprüfung der Erfüllung von Integrationsanforderungen dieser Kategorien mit Hilfe des 3LGM_A² (siehe Antwort zu Frage F4.2).

F1.2

Welche Techniken können zur Erfüllung der Integrationsanforderungen verwendet werden und welche Vor- bzw. Nachteile haben sie?

Für Integrationstechniken existieren sehr viele Standards, die oft nur mit Hilfe von Rahmenwerken für die Informationssystemarchitektur verstanden und angewendet werden können. In Kapitel 4 wurden die Rahmenwerke

- Referenzmodell für die Kopplung offener Systeme (OSI-Referenzmodell),
- Referenzmodell für offene verteilte Informationsverarbeitung (RM-ODP),
- Object Management Architecture (OMA),
- Healthcare Information Framework (HIF),
- The Open Group Architectural Framework, Version 7, (TOGAF 7),
- Zachman-Rahmenwerk,
- Architektur integrierter Informationssysteme (ARIS),
- Enterprise Application Integration (EAI),
- The Open Group Architectural Framework, Version 8, (TOGAF 8) und
- Enterprise Application Planning (EAP)

vorgestellt.

Sehr viele Standards für Integrationstechniken wurden auf der Basis des OSI-Referenzmodells entwickelt oder können leicht den Ebenen des OSI-Referenzmodells zugeordnet werden.

Die folgenden Absätze geben einen beispielhaften Überblick.

Physische Integration Für die physische Integration können u. a. die in der Standardreihe 802.X der IEEE spezifizierten Techniken verwendet werden. Die Standards 802.3x spezifizieren beispielsweise das sogenannte Ethernet, die Standards 802.11x die als Wireless LAN bekannten Techniken (vgl. Abschnitt 4.2.1).

Datenintegration Für die Datenintegration können alle Techniken verwendet werden, die allgemein für den (logischen) Datenaustausch vorgesehen sind. Dazu existieren u. a. anwendungsbereichsunabhängige Standards für die Dokumentenübertragung, z. B. HTTP, aber auch

anwendungsbereichsbezogene Standards für Nachrichten- bzw. Dokumentenformate und ggf. zugehörige Ereignistypen, z. B. HL7. Auch die Kommunikation von Komponenten in einer komponentenorientierten Architektur dient oft der Herstellung von Datenintegration zwischen den Komponenten. Deshalb können z. B. auch CORBA und DCOM als Standards für Datenintegration genannt werden (vgl. Abschnitte 4.2.1, 4.2.3 und 4.3).

Funktionale Integration Für die funktionale Integration werden Techniken verwendet, die das Aufrufen von Prozeduren, Funktionen oder, in komponentenorientierter Terminologie, Operationen ermöglichen. Neben bereits älteren Techniken wie RPC gehören dazu u. a. ebenfalls die CORBA und das DCOM (vgl. Abschnitte 4.2.3 und 4.3).

Semantische Integration Wenn semantische Integration einfach als Abgleich von Daten zu Begriffssystemen oder, allgemeiner, Katalogen verstanden wird, können alle Techniken für die Datenintegration auch für semantische Integration verwendet werden. Ein Beispiel für „höherwertige“ Techniken für semantische Integration sind die Dienstgruppe „Concept Healthcare Common Services“ der HISA und Terminologieserver (vgl. Abschnitte 4.2.4 und 4.3).

Kontextintegration Für Kontextintegration existieren nur wenige bekannte Techniken. Beispiele sind die mit dem Microsoft Internet Explorer verfügbare Integrierte Windows-Authentifizierung sowie, für den Anwendungsbereich des Gesundheitswesens, die Techniken des CCOW-Standards (vgl. Abschnitt 4.3).

Präsentationsintegration Die Herstellung von Präsentationsintegration ist nicht notwendigerweise mit der Bereitstellung und ggf. Übermittlung von Daten verbunden. Zunächst wird die einheitliche Gestaltung von Benutzungsschnittstellen gefordert. Beispiele für entsprechende Standards sind CUA, Motif, die Apple Human Interface Guidelines oder die Grundsätze ergonomischer Dialoggestaltung für Bildschirmarbeitsplätze (DIN 66234) (vgl. Abschnitt 4.3).

Wenn über funktionale Integration Elemente von Benutzungsschnittstellen wiederverwendet werden, dann trägt funktionale Integration zur Präsentationsintegration bei.

Zugriffsintegration Für die Zugriffsintegration können alle Techniken zur zentralen Benutzerverwaltung verwendet werden. Dazu werden viele Produkte angeboten, z. B. der Active Directory Server oder der RSA Authentication Service.

Ein Beispiel für einen Standard ist die Spezifikation des Kerberos Network Authentication Service (vgl. Abschnitt 4.2.1).

Bewertungsansätze für den Einsatz von Integrationstechniken zur Bestimmung von Vor- und Nachteilen bestimmter Techniken werden in der Antwort zur Frage F4.2 beschrieben.

Z.2.2 Fragen zu Ziel Z2

F2.1

Welche bekannten Architekturstandards gibt es und welche Charakteristika haben sie?

Architekturstandards wurden in dieser Arbeit in Kapitel 4 mit Bezug zu verschiedenen Rahmenwerken für die Informationssystemarchitektur vorgestellt. Die meisten behandelten Rahmenwerke sind selbst Standards.

Abgrenzung: Mit Architekturstandards sind hier NICHT Standards für spezielle Integrationstechniken gemeint.

Die Rahmenwerke wurden in die Kategorien *Technische Rahmenwerke* und *Unternehmensbezogene Rahmenwerke* eingeteilt. Die vorgestellten technischen Rahmenwerke sind

- das OSI-Referenzmodell,
- das Referenzmodell für offene verteilte Informationsverarbeitung (RM-ODP),
- die Object Management Architecture (OMA),
- das Healthcare Information Framework (HIF) und
- (der) The Open Group Architectural Framework, Version 7, (TOGAF 7).

Die vorgestellten unternehmensbezogenen Rahmenwerke sind

- das Zachman-Rahmenwerk,
- die Architektur integrierter Informationssysteme (ARIS),
- die Enterprise Application Integration (EAI),
- (der) The Open Group Architectural Framework, Version 8, (TOGAF 8) und
- das Enterprise Application Planning.

Die Rahmenwerke wurden hauptsächlich hinsichtlich der Bereitstellung von Begriffsdefinitionen und Metamodellen sowie der Bereitstellung von Informations-, Architektur- und Vorgehens-Referenzmodellen betrachtet. Am Ende von Kapitel 4 wurde dazu eine vergleichende Übersicht erstellt.

F2.2

Wie stehen die Integrationsanforderungen mit den Architekturstandards in Beziehung?

Architekturstandards werden i. d. R. entworfen, um komplexe Informationssysteme durch Aufteilung in Komponenten besser beherrschen zu können. Die Tatsache, dass die betreffenden Komponenten zusammenwirken müssen, d. h. dass Integration gewährleistet werden muss, wird dabei beim Entwurf des Standards durch die Festlegung der Beziehungen zwischen den Komponenten unmittelbar deutlich.

Vor allem die technischen Rahmenwerke und die zugehörigen Standards für einzelne Elemente der Rahmenwerke beschreiben Ansätze zur Realisierung von Kommunikationsbeziehungen zwischen Komponenten, z. B. durch Spezifikation von Schnittstellen oder von Protokollen für die Kommunikation. Damit werden die konzeptionellen Voraussetzungen für die Herstellung von Integration, d. h. die Erfüllung von Integrationsanforderungen, geschaffen. In der Antwort zu Frage 1.2 wurden den einzelnen Anforderungskategorien beispielhaft Standards zu Integrationstechniken zugeordnet.

Die in Abschnitt 4.2 vorgestellten technischen Rahmenwerke wurden, mit Ausnahme von TOGAF, als Grundlage für spezielle Architekturstandards für Komponenten und Kommunikationsprotokolle entwickelt. Im Gegensatz dazu helfen TOGAF in der Version 7 und die vorgestellten unternehmensbezogenen Rahmenwerke, mit der Vielzahl der Integrationsanforderungen und den existierenden technischen Lösungen umzugehen.

F2.3

Welchen Integrationstechniken liegen die Architekturstandards zugrunde?

Wie bereits in der Antwort zu Frage F2.2 beschrieben, wurden die meisten der technischen Rahmenwerke entwickelt, um auf ihrer Basis spezielle Architekturstandards für Komponenten und Kommunikationsprotokolle, d. h. für spezielle Integrationstechniken, zu erstellen.

OSI-Referenzmodell Das OSI-Referenzmodell ist die Grundlage für viele spezielle Standards zur physischen Integration, z. B. die IEEE-Standardreihe 802.X, und für viele eng mit der physischen Integration verknüpfte Protokolle für den Datenaustausch (vgl. Abschnitt 4.2.1).

RM-ODP Das RM-ODP beschreibt zunächst Grundlagen für den Entwurf verteilter Informationssysteme und damit für viele Integrationstechniken. Die im RM-ODP beschriebenen speziellen Funktionen, die in einem verteilten System implementiert sein sollten, können als spezielle Techniken zur Verwaltung der Verteilung betrachtet werden (vgl. Abschnitt 4.2.2).

OMA Die OMA wurde ebenfalls für den Entwurf verteilter Informationssysteme entwickelt. Sie stellt im Gegensatz zum allgemeiner ausgerichteten RM-ODP den Aspekt der Bereitstellung von Diensten durch Komponenten und das Aufrufen der Dienste in den Vordergrund. Sie ist damit zunächst Grundlage für funktionale Integration (vgl. Abschnitt 4.2.3).

Funktionale Integration steht in einer Dualitätsbeziehung zu Datenintegration. Gleichzeitig kann funktionale Integration, wie Datenintegration, zur Erfüllung von Integrationsanforderungen anderer Anforderungskategorien, z. B. semantischer Integration oder Zugriffsintegration, dienen. Damit können die OMA und die ergänzenden Standards zur Erfüllung von allen Integrationsanforderungen, mit Ausnahme der physischen Integration, beitragen.

Der vermutlich wichtigste ergänzende Standard zur OMA ist die Common Object Request Broker Architecture (CORBA). Darin wird die zentrale Komponente der OMA, der Object Request Broker (ORB), spezifiziert. Der ORB ist eine Technik zur Vermittlung der Dienstanforderungen an Komponenten.

Zusätzlich wurden viele ergänzende Standards zu den in der OMA beschriebenen Dienstkategorien erstellt, deren Implementierungen zur Erfüllung vieler Integrationsanforderungen beitragen können.

HIF Das HIF wurde als Grundlage für spezielle Dienstspezifikationen für den Anwendungsbereich des Gesundheitswesens entwickelt. Implementierungen der auf der Basis des HIF in ergänzenden Standards spezifizierten Dienstgruppen können zur Erfüllung mehrerer Integrationsanforderungen beitragen (vgl. Abschnitt 4.2.4).

TOGAF TOGAF ist ein Rahmenwerk für den Entwurf bzw. die Weiterentwicklung von Informationssystemarchitekturen und damit auch für die Anwendung von Integrationstechniken. Die zuvor genannten Rahmenwerke sind Grundlage für die Entwicklung weiterer Standards für Integrationstechniken. Im Gegensatz dazu werden in der von der Open Group für TOGAF bereitgestellten Standards Information Base (TOGAF SIB) viele von anderen Organisationen erstellte Standards zusammengefasst und den Dienstkategorien des TOGAF Technical Reference Model (TOGAF TRM) zugeordnet (vgl. Abschnitt 4.2.5).

Z.2.3 Fragen zu Ziel Z3

F3.1

Wie stehen die in Frage F2.1 genannten Architekturstandards mit dem 3LGM² in Beziehung?

Ein Ziel der Entwicklung des 3LGM² ist die Unterstützung des strategischen Managements bei der Auswahl von Integrationstechniken auf der Basis der zugrunde liegenden Standards. Dazu wurde das 3LGM² in Kapitel 7 auf der Basis des Kapitel 4 zu Architekturstandards und des Kapitel 5 zur Geschäftsprozessmodellierung überarbeitet (für eine ausführlichere Zusammenfassung der Überarbeitung siehe Antwort zur Frage F3.3).

Für das überarbeitete 3LGM² wurden die Ansätze zur Architekturbewertung in den Kapiteln 9 bis 11 entwickelt. Damit können Vergleiche der Anwendung verschiedener Integrationstechniken durchgeführt werden.

F3.2

Wie muss das 3LGM² überarbeitet werden, um die in Frage F1.1 genannten Integrationsanforderungen modellieren zu können?

Für die Modellierung der Integrationsanforderung *semantische Integration* wurde in Kapitel 7 die Klasse *Begriffssystem* in das 3LGM² eingeführt. Mit den zum in Kapitel 9 entwickelten Domänenkonzept gehörenden Anforderungsdomänen kann das Bestehen von Integrationsanforderungen zwischen *Anwendungsbausteinen* ermittelt werden (für eine ausführlichere Zusammenfassung zum Domänenkonzept siehe Beantwortung der Frage F4.2).

Bei der Erstellung des Domänenkonzeptes wurden die für die Architekturmodellierung vorgenommenen Überarbeitungen des 3LGM² berücksichtigt. Sie werden in der folgenden Antwort zusammengefasst.

F3.3

Wie muss das 3LGM² überarbeitet werden, um die in Frage F1.2 genannten Integrationstechniken modellieren sowie Vor- und Nachteile bestimmen zu können?

In Kapitel 7 wurde das 3LGM² zum 3LGM_A² überarbeitet, um die Modellierung der Anwendung verbreiteter Architekturkonzepte zu ermöglichen. Dazu wurde auf der logischen Werkzeugebene die Klasse *Operation* eingeführt, und die Assoziationsbeziehungen der Klassen für die Integrationsmodellierung wurden überarbeitet.

Die Überarbeitung reflektiert das Objekt- bzw. Komponentenkonzept, das hauptsächlich im RM-ODP und in den mit der OMA eng verbundenen Metamodellen Object Model und Component Model beschrieben wird. Über das Modellieren der Bereitstellung von Operationen kann auch die Implementierung von Diensten modelliert werden. Dienste bzw. Dienstkategorien werden u. a. in der OMA und ergänzenden Standards, in der auf dem HIF basierenden Healthcare Information Systems Architecture, Part I, (HISA) und im TOGAF TRM spezifiziert.

Die Überarbeitung stellt eine starke Verallgemeinerung der in den genannten Werken definierten Begriffe für die Architekturbeschreibung dar, mit der die Möglichkeit der Beschreibung bestimmter Details verloren geht. Diese Verallgemeinerung wurde hier akzeptiert, um die Komplexität des 3LGM_A² im Vergleich zum 3LGM² nicht wesentlich zu erhöhen (vgl. Abschnitte 6.1 und 7.1).

Z.2.4 Fragen zu Ziel Z4

F4.1

Welche Gemeinsamkeiten und welche Unterschiede bestehen zwischen den in Frage F2.1 genannten Architekturstandards?

In Abschnitt 4.6 wurden die in Kapitel 4 vorgestellten Rahmenwerke unter Berücksichtigung einiger weiterer zugehöriger Standards gegenübergestellt. Die Gegenüberstellung wurde auf der Basis der zu den Rahmenwerken gehörenden Metamodelle, der Unterscheidungen von Sichtweisen auf die Architektur und der Bereitstellung von Referenzmodellen vorgenommen.

Eine vollständige Überführungsvorschrift zwischen einzelnen Metamodellen oder Referenzmodellen im Sinne einer bijektiven Abbildung von Elementen eines Rahmenwerkes oder ergänzender Standards auf Elemente eines anderen Rahmenwerkes oder ergänzenden Standards kann aufgrund der verschiedenen Schwerpunkte und der damit verbundenen unterschiedlichen Terminologie nicht angegeben werden.

Gemeinsamkeiten und eine damit verbundene prinzipielle Gleichwertigkeit lassen sich für

- zentrale Begriffe des RM-ODP, z. B. Objekt, Schnittstelle oder Ereignis und
- zentrale Begriffe des OMG Object Models und des OMG Component Models

feststellen.

Ähnlichkeiten können ebenfalls für

- die auf der Basis der OMA entwickelten Dienstspezifikationen,
- die auf der Basis des HIF entwickelten HISA-Dienstspezifikationen und
- die Dienstkategorien des TOGAF TRM¹

festgestellt werden.

¹ Das TOGAF TRM gibt eine textliche funktionale Spezifikation für Dienstkategorien an, jedoch keine formale informationale oder funktionale Spezifikation. Insofern ist die Aussage zur Gleichwertigkeit zu den anderen Spezifikationen nur bei entsprechender Abstraktion von den Spezifikationsdetails auf allgemeine Angaben zur Funktionalität gerechtfertigt.

F4.2

Wie können die in Frage F2.3 genannten Integrationstechniken ausgetauscht werden und welche Einbußen oder Gewinne an Funktionalität entstehen dabei?

In Teil III dieser Arbeit wurden drei Ansätze zur Bewertung der Integration entwickelt. Mit diesen Ansätzen können konkrete Anwendungen von Integrationstechniken in bestimmten Informationssystemen bewertet und verglichen werden.

Für die Entwicklung der Bewertungsansätze wurden folgende Grundannahmen getroffen (vgl. Kapitel 1):

GA1: Bei der Bewertung der Architektur stehen folgende Fragen im Vordergrund:

- Welche Integrationsanforderungen bestehen?
- Sind die Integrationsanforderungen erfüllt?
- Wie groß ist die Abhängigkeit zwischen Anwendungssystemen und wo besteht Potential zur Verringerung der Abhängigkeit?
- Wie komplex ist die Architektur des Informationssystems und wo besteht Vereinfachungspotential?

GA2: Mit der Anzahl unterschiedlicher verwendeter Integrationstechniken steigt die Komplexität und sinkt die Beherrschbarkeit eines Informationssystems. Das gilt insbesondere dann, wenn mehrere Integrationstechniken zur Erfüllung derselben Integrationsanforderung(en) angewendet werden.

Die Bewertungsansätze basieren auf dem in Kapitel 6 vorgestellten Metamodell $3LGM^2$ und seiner in Kapitel 7 entwickelten Erweiterung zum $3LGM_A^2$. Ihre Ausdrucksfähigkeit ist daher durch die Vorgaben des $3LGM_A^2$ beschränkt. Diese Beschränkung wird hier ausdrücklich akzeptiert, da die Bewertungsansätze, wie auch das $3LGM_A^2$, ohne umfassende theoretische Kenntnisse einer komplexen Modellierungssprache angewendet werden sollen (vgl. Abschnitte 6.1 und 7.1).

1. Mit dem Domänenkonzept (Kapitel 9) kann in einem $3LGM_A^2$ -Modell überprüft werden, ob Integrationsanforderungen erfüllt sind. Es können Mengen von **Anwendungsbausteinen** (Domänen) ermittelt werden, für die bestimmte Integrationsanforderungen bestehen (Anforderungsdomänen), und Mengen von **Anwendungsbausteinen**, für die bestimmte Integrationsanforderungen erfüllt sind (Kommunikationsdomänen).

Anforderungsdomänen sind auf der Basis der erarbeiteten Anforderungskategorien (vgl. Antwort zu Frage F1.1) in

- Datendomänen,
- funktionale Domänen,
- semantische Domänen,
- Kontextdomänen,
- Präsentationsdomänen und
- Zugriffsdomänen

unterteilt. Kommunikationsdomänen sind in

- Übermittlungsdomänen für **Objekttypen** und
- Aufrufdomänen für **Aufgaben**

unterteilt. Die Übermittlung von **Objekttypen** steht dabei für das Übermitteln bestimmter Daten zwischen **Anwendungsbausteinen**, das Aufrufen von **Aufgaben** steht für das Bereitstellen von Funktionalität von **Anwendungsbausteinen** für andere **Anwendungsbausteine** und das Nutzen dieser Funktionalität durch andere **Anwendungsbausteine** (vgl. auch Ende von Abschnitt 7.2.2).

Durch Vergleich der Anforderungsdomänen mit den Kommunikationsdomänen kann die Erfüllung der Integrationsanforderungen überprüft werden. Abschnitt 9.5 enthält Beispiele für die Durchführung der Vergleiche und Hinweise zur Interpretation.

2. In Kapitel 10 wurden Kennzahlen zur Abhängigkeit von **Anwendungsbausteinen** und zur Bewertung der Qualität von verteilten Transaktionen erarbeitet. Über die vier Kennzahlen

- informationaler Abhängigkeitsgrad,
- funktionaler Abhängigkeitsgrad,
- Ausführungsabhängigkeitsgrad und
- Transaktionsabhängigkeitsgrad

kann beschrieben werden, wie stark ein **Anwendungsbaustein** von anderen **Anwendungsbausteinen** abhängt. Informationale Abhängigkeit ist Abhängigkeit hinsichtlich der von anderen **Anwendungsbausteinen** bereitgestellten Daten; funktionale Abhängigkeit ist Abhängigkeit hinsichtlich der bereitgestellten **Operationen**. Ausführungsabhängigkeit beschreibt den Sachverhalt, dass ein **Anwendungsbaustein** bei seiner Ausführung durch andere **Anwendungsbausteine** blockiert werden kann; Transaktionsabhängigkeit beschreibt den Sachverhalt, dass **Anwendungsbausteine**, die verteilte Transaktionen auslösen, von der Bestätigung der Transaktionsausführung in anderen **Anwendungsbausteinen** abhängig sind.

Mit der

- Transaktionsstärke

wird bestimmt, wie bei redundanter Speicherung von Informationen die Verteilung von Datenänderungen abgesichert ist.

3. In Kapitel 11 wurden Kennzahlen für die Bewertung der Komplexität der Integrationsinfrastruktur erarbeitet. Mit Hilfe der drei Kennzahlen

- absoluter Heterogenitätsgrad,
- relativer Heterogenitätsgrad und
- kostenbewerteter Heterogenitätsgrad

wird beschrieben, wie stark Integrationstechniken in den Kommunikationsverbindungen eines Informationssystems wiederverwendet werden. Dabei wird vorausgesetzt, dass Integrationstechniken in angemessener Weise über **Bausteinschnittstellen**, **Operationen**, **Nachrichtentypen**, **Dokumententypen** und **Kommunikationsbeziehungen** modelliert werden können.

Der relative Heterogenitätsgrad kann als Schätzfunktion für die Anzahl von potentiell ersetzbaren **Bausteinschnittstellen** und implementierten **Operationen** interpretiert werden. Der kostenbewertete Heterogenitätsgrad kann als Schätzfunktion für die potentiell wegen Heterogenität unnötig entstehenden Kosten interpretiert werden.

Beim Vergleich von Architekturen mit unterschiedlichen Integrationstechniken kann über die

Anwendung des Domänenkonzeptes festgestellt werden, welche Integrationsanforderungen in den Architekturen erfüllt sind. Die Kennzahlen ermöglichen den Vergleich hinsichtlich der „Güte“ der Integration. Wenn beispielsweise in einem Integrationsszenario die Autonomie, d. h. Unabhängigkeit, von Anwendungsbausteinen wesentliches Merkmal ist, können die genannten Abhängigkeitsgrade als wesentliche Vergleichskriterien verwendet werden.

Mit Hilfe von hinreichend detaillierten Architektur-Referenzmodellen, in die die Anwendung verschiedener Integrationstechniken „hineinmodelliert“ wird, können Integrationstechniken auch unabhängig von konkreten existierenden oder geplanten Informationssystemarchitekturen verglichen werden. Entsprechende Referenzmodelle müssten alle typischerweise in einem festzulegenden Anwendungsbereich genutzten Komponenten beinhalten. Weiterhin müssen die zugehörigen Integrationsanforderungen und die im Vordergrund stehenden Bewertungskriterien festgelegt werden (vgl. Abschnitt 4.6).

Z.3 Diskussion

Z.3.1 Kommentierung der Ergebnisse

In dieser Arbeit wurde ein Ansatz zur Modellierung von Architekturen verschiedener Architekturstile entwickelt. Grundlage dafür ist das Metamodell $3LGM^2$, das für die Architekturmodellierung zum $3LGM_A^2$ überarbeitet wurde.

Beschränkung der Komplexität der Modellierung

Für die Nutzung des $3LGM_A^2$ im Informationsmanagement sollten keine ausführlichen Kenntnisse von Architekturstandards vorausgesetzt werden. Gleichzeitig sollte die Komplexität des $3LGM_A^2$ im Vergleich zum $3LGM^2$ nicht wesentlich erhöht werden. Bei der Überarbeitung des $3LGM^2$ wurden deshalb nur wenige grundlegende Begriffe berücksichtigt. Die Auswahl kann dazu führen, dass für die Modellierung bestimmter Sachverhalte keine ausdrücklich dafür definierten Begriffe zur Verfügung stehen. Die konsequente Nutzung der zur Verfügung stehenden Begriffe, die eine Herausforderung bei der Modellierung darstellt, kann jedoch das Vergleichen von Architekturen unterschiedlicher Architekturstile oder von speziellen, auf ein bestimmtes Problem zugeschnittenen Architekturen, erleichtern.

Kategorisierung der Integrationsanforderungen

Für Informationssystemmodelle auf der Basis des $3LGM_A^2$ wurden Ansätze zur Bewertung von Informationssystemen entwickelt. Das Domänenkonzept kann zur Überprüfung der Erfüllung von Integrationsanforderungen genutzt werden. Dazu müssen die bei einem bestimmten Integrationsszenario zu berücksichtigenden Anforderungen in die im Grundlagenkapitel zu Integrationsanforderungen erarbeiteten Anforderungskategorien eingeteilt werden, was u. U. zunächst einen Mehraufwand bedeutet.

Die Kategorien können aber gleichzeitig als Hilfestellung zur systematischen Erarbeitung der im konkreten Fall bestehenden Integrationsanforderungen genutzt werden. Mit einem Modellierungswerkzeug, das die Überprüfung der Erfüllung der Anforderungen nach dem Domänenkonzept umsetzt, kann die Überprüfung dann leicht erfolgen. Voraussetzung ist eine

angemessene Modellierung der Integrationsinfrastruktur (Bausteinschnittstellen, Operationen usw.).

Verallgemeinerung durch Kennzahlberechnung

Die ebenfalls für die Bewertung erarbeiteten Kennzahlen wurden konzipiert, um Architekturen hinsichtlich der Abhängigkeit von Anwendungsbausteinen und der Komplexität der Integrationsinfrastruktur zu vergleichen. Der Vergleich kann dabei auch zwischen einer vorhandenen Architektur (IST-Zustand) und einem geplanten Zustand nach Überarbeitung (SOLL-Zustand) erfolgen. Einige der Kennzahlen können helfen, den Umfang der möglicherweise ersetzbaren Elemente und die durch sie verursachten Kosten abzuschätzen. Bei der Nutzung der Kennzahlen können u. U. spezielle Bedingungen hinsichtlich der Verwendung bestimmter Komponenten nicht berücksichtigt werden. Die Nutzung der Berechnungsvorschriften ist implizit mit einer verallgemeinerten Betrachtung der betrachteten konkreten Architektur verbunden. Das Ausmaß der Verallgemeinerung wird durch die Ausdrucksfähigkeit der für die Berechnungen verwendeten Modellierungskonstrukte und zusätzlich durch die beschränkte Flexibilität der Berechnungsvorschriften bestimmt.

Spezifische Probleme der Kennzahlenanwendung

Zusätzlich zur Beachtung der mit der Berechnung von Kennzahlen verbundenen Verallgemeinerung sind die für die Heterogenitätsgrade angegebenen spezifischen Probleme zu berücksichtigen. Dazu gehören die Abhängigkeit vom Detaillierungsgrad der Modellierung und die möglicherweise für bestimmte Architekturen ungeeignete Festlegung der Bewertungen von Abweichungen der Bausteinschnittstellen und der Operationen.

Z.3.2 Erfüllung der Ziele

Für jedes der in der Einleitung der Arbeit gestellten Ziele wurde eine Lösung entworfen. Die Lösungen sind

- zu Ziel Z1:
 - die Anforderungskategorien für Integrationsanforderungen,
 - die im Zusammenhang mit Rahmenwerken vorgestellten spezifischen Standards für Integrationstechniken und
 - das zur Überprüfung der Integrationsanforderungen entworfene Domänenkonzept;
- zu Ziel Z2:
 - die Vorstellung der Architekturstandards und die erstellte vergleichende Übersicht;
- zu Ziel Z3
 - die unternommene Überarbeitung des $3LGM^2$ zum $3LGM_A^2$;
- zu Ziel Z4
 - das Domänenkonzept zur Überprüfung der Erfüllung von Integrationsanforderungen und
 - die Kennzahlen *informationaler Abhängigkeitsgrad*, *funktionaler Abhängigkeitsgrad*, *Ausführungsabhängigkeitsgrad* und *Transaktionsabhängigkeitsgrad*,

- die Kennzahl *Transaktionsstärke* und
- die Kennzahlen *absoluter Heterogenitätsgrad*, *relativer Heterogenitätsgrad* und *kostenbewerteter Heterogenitätsgrad*.

Die Lösungen verallgemeinern, wie im vorhergehenden Abschnitt kommentiert, die Integrationsproblematik durch die Kategorisierung der Integrationsanforderungen, die Anwendung des $3LGM_A^2$ und den verallgemeinernden Charakter der Berechnungsvorschriften. Mit dem Verallgemeinern wird die Möglichkeit zum Vergleich verschiedener Architekturen geschaffen; mit der Akzeptanz der Verallgemeinerung und der Anwendung der hier vorgestellten Bewertungsansätze entfällt die Notwendigkeit, spezifische Bewertungsansätze für konkrete Bewertungen zu entwerfen.

Z.4 Potential für Evaluationen und für weiterführende Forschung

Die vorgestellten Bewertungsansätze wurden auf Beispiele aus dem Universitätsklinikum Leipzig angewendet. Mit der Anwendung auf andere Informationssysteme muss die allgemeine Praktikabilität untersucht werden. Dabei sollten insbesondere auch Informationssysteme, die moderne Techniken nutzen (z. B. auf der Basis der Grid-Technologie), einbezogen werden.

Es wurde mehrfach auf das Problem der Festlegung der Bewertungen von Abweichungen zwischen Bausteinschnittstellen und Operationen bei der Berechnung von Heterogenitätsgraden hingewiesen. Gezielte Studien können wesentlich zur sicheren Festlegung der Bewertungen, u. U. abhängig von betrachteten Architekturstilen, beitragen.

Die entworfenen Bewertungsansätze können zur Analyse von Informationssystemen mit vielen einzelnen Komponenten und komplexen Integrationsinfrastrukturen beitragen. „Große“ Informationssysteme, z. B. in regionalen Informationsverbänden, werden oft von mehreren verantwortlichen Einrichtungen betreut und weiterentwickelt. Die für die Bewertung notwendigen Modelle können daher u. U. nur durch paralleles Modellieren an unterschiedlichen Standorten erstellt werden. Damit entstehen hohe Anforderungen an eine Infrastruktur zur verteilten Modellierung, für die geeignete Lösungen geschaffen werden müssen.

Anhang A Anforderungskatalog für die Informationsverarbeitung im Krankenhaus (Auszug)

Der Anhang enthält einen Auszug aus dem Anforderungskatalog für die Informationsverarbeitung im Krankenhaus [IMBI/MI 2001], S. 28 ff, der die Integration von Informationssystemkomponenten betrifft.

3.3 **Gruppe III: Integration**

- ✓ Gewährleistung, dass die unterschiedlichen (rechnerbasierten und konventionellen) Informationssystemkomponenten reibungslos und effizient zusammenarbeiten.
- ☞ Der Anforderungskatalog kann nur allgemeine Vorgehensweisen beschreiben. Im konkreten Anwendungsfall muss ggf. eine Kommunikationsmatrix, in der alle betroffenen Schnittstellen zwischen Anwendungssystemen betrachtet werden können, erstellt werden.
- ☞ Die verschiedenen Integrationsaspekte erfordern manchmal Abstriche von der maximal möglichen

A.x.x.x: Anforderung ✓: Ziel ☞: zu beachten

chen Funktionalität. Hier müssen die Vor- und Nachteile einer Spezialisierung gegenüber einem gut integrierten System im Einzelfall abgewogen werden.

Aspekt III.1: Prozessintegration

- ✓ Optimale Einbettung der Nutzung von Daten und informationsverarbeitenden Funktionen in Arbeitsabläufe, unabhängig davon welches Anwendungssystem diese Daten oder Funktionen zur Verfügung stellt.

Teilaspekt III.1.1: Datenintegration

- ✓ Ermöglichung der multiplen Verwendung von einmal aufgezeichneten Daten und damit Vermeidung von Aufzeichnungsfehlern und Dateninkonsistenzen.

A.III.1.1.a	Daten sollten möglichst am Ort der Entstehung und zum Zeitpunkt der Entstehung aufgezeichnet werden.
A.III.1.1.b	Gleiche Daten (wie z.B. Patientenstammdaten, Hauptdiagnosen) sollten nur einmal erfasst werden.
A.III.1.1.c	Alle Daten sollten möglichst für verschiedene Zwecke verwendet werden können.
A.III.1.1.d	Medienbrüche bei der Weitergabe von Daten sind zu vermeiden.
A.III.1.1.e	Es existiert ein globales Datenmodell mit klinikumswweit eindeutigen, semantikfreien Objektidentifikationen für die zentralen Objekte wie z.B. Einrichtungen, Patienten, Fälle, Maßnahmen, Dokumente.
A.III.1.1.f	Änderungen an bereits vorhandenen Daten, die redundant in weiteren Anwendungssystemen gespeichert werden, werden über Standardschnittstellen kommuniziert und nachvollzogen (Synchronisation).
A.III.1.1.g	Die logische Zusammenführung aller Daten zu einem Patienten ist möglich.

Teilaspekt III.1.2: Funktionsintegration

- ✓ Reduktion von sich überlappender Funktionalität in unterschiedlichen Anwendungssystemen.
- ✓ Vermeidung von redundanter Ausführung von Funktionen, die zu dem gleichen Ergebnis führen.

A.III.1.2.a	Mehrfach benötigte Funktionen (z.B. Patientenidentifizierung) sind nur in einem Anwendungssystem verfügbar ("funktionale Schlankheit").
A.III.1.2.b	Ergebnisse, die durch das Ausführen einer Funktion eines Anwendungssystems entstehen, stehen danach anderen Anwendungssystemen zur Verfügung.

Teilaspekt III.1.3: Ablaufintegration

- ✓ Alle Anwendungssysteme sollen sich reibungslos in die Abläufe der Benutzer einfügen.

A.III.1.3.a	Die Anwendungssysteme können an die Arbeitsabläufe der Benutzer angepasst werden.
A.III.1.3.b	Die von den verschiedenen Benutzergruppen benötigten Funktionen stehen jeweils unkompliziert und schnell am Arbeitsplatz zur Verfügung.
A.III.1.3.c	Die von den verschiedenen Benutzergruppen benötigten Daten sollen jeweils am richtigen Ort zur richtigen Zeit in der richtigen Form zur Verfügung stehen.

Aspekt III.2: Werkzeugintegration

- ✓ Schaffung der technischen Voraussetzungen für eine integrierte Informationsverarbeitung.
- ☞ Werkzeugintegration ist Voraussetzung für alle anderen Integrationsaspekte.

A.x.x.x: Anforderung ✓: Ziel ☞: zu beachten

- ☞ Die Heterogenität der Werkzeuge der Informationsverarbeitung (z.B. Hardware, Netzwerk-komponenten, Netzprotokolle, Formulare, Telefone etc.) ist möglichst gering zu gehalten, um Aufwände für Betrieb und Wartung zu minimieren.

Teilaspekt III.2.1: Zugangs- und Zugriffsintegration

- ✓ Vereinfachung der Verwendung von Anwendungssystemen durch einen einheitlichen Benutzerzugang.

A.III.2.1.a	Alle für einen Benutzer relevanten Funktionen sind in einfacher, einheitlicher Weise am klinischen Arbeitsplatz verfügbar.
A.III.2.1.b	Der Anwender soll sich gegenüber dem System nur einmal authentifizieren müssen ("Single-Sign-On"). Dies erfordert auch ein personenbezogenes Berechtigungskonzept sowie schnelle, einfache Benutzerwechsel (siehe auch Gruppe V: Datenschutz).

Teilaspekt III.2.2: Präsentationsintegration

- ✓ Vereinfachung der Benutzung von Anwendungssystemen und Reduktion des Schulungsaufwands.

A.III.2.2.a	Die Benutzungsschnittstellen unterschiedlicher Anwendungssysteme sind ähnlich gestaltet und einheitlich zu bedienen.
A.III.2.2.b	Daten unterschiedlicher Anwendungssysteme werden adäquat und konsistent präsentiert (siehe auch Gruppe VI: Benutzungsschnittstelle).

Teilaspekt III.2.3: Kommunikationsintegration

- ✓ Unterstützung der effizienten Kommunikation zwischen Anwendungssystemen.
- ✓ Reduktion der Aufwände für Schnittstellenentwicklung.
- ☞ Im konventionellen Bereich sind Standardschnittstellen z.B. standardisierte Formulare und Absprachen, wer diese wann und wie ausfüllt bzw. wann wo abzugeben hat.
- ☞ In der Regel lassen Standards Interpretationsspielraum. Daher sind detaillierte Spezifikationen und Absprachen über die Inhalte auch bei der Verwendung von Standards unbedingt erforderlich.

A.III.2.3.a	Es werden Standard-Schnittstellenprotokolle für den Austausch von Nachrichten zwischen rechnerbasierten Anwendungssystemen verwendet.
A.III.2.3.b	Wegen der besonderen Bedeutung der Kommunikation im rechnergestützten Teil eines Informationssystems sind Schnittstellendefinitionen schriftlich festzuhalten und ausführliche Funktionstests durchzuführen und zu dokumentieren.
A.III.2.3.c	Neben rechnergestützten Werkzeugen sollten auch die medizintechnischen Geräte (z.B. bildgebende Modalitäten, Laborgeräte etc.) Schnittstellen anbieten.
A.III.2.3.d	Die Zahl der Schnittstellen zwischen Anwendungssystemen sowie die Zahl verwendeter Schnittstellenprotokolle ist möglichst gering zu halten.
A.III.2.3.e	Die Verantwortlichkeiten für Zeitpunkt, Inhalt und Qualität der Kommunikation sind klar geregelt.
A.III.2.3.f	Der korrekte Empfang übermittelter Daten kann überprüft werden.

Teilaspekt III.2.4: Technische Integration

- ✓ Schaffung der technischen Voraussetzungen für integrierte Informationsverarbeitung.

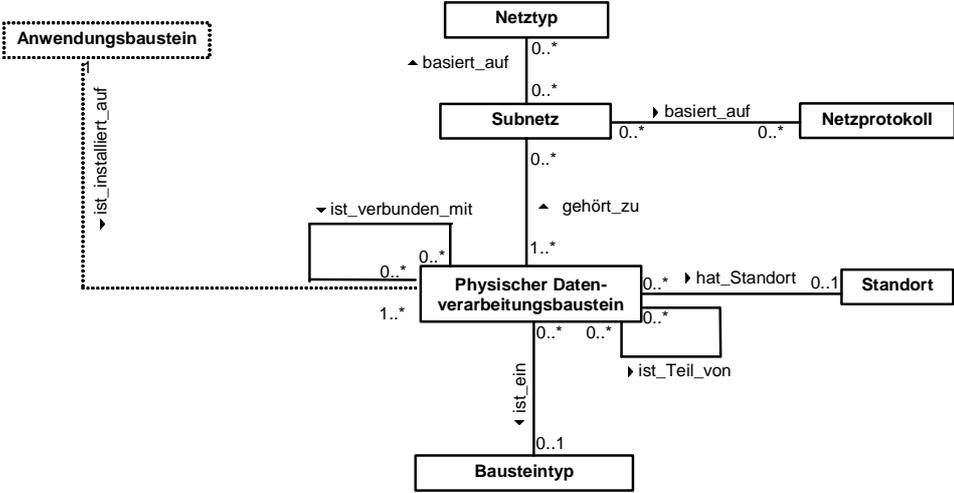
A.III.2.4.a	Leistungsfähige Rechner stehen in ausreichender Zahl an geeigneten Orten zur Verfügung. Zu diesen Orten gehören je nach den zu unterstützenden Aufgaben z.B. Stationszimmer, Arztzim-
-------------	---

A.x.x.x: Anforderung ✓: Ziel ☞: zu beachten

Anhang A Anforderungskatalog für die Informationsverarbeitung im Krankenhaus (Auszug)

	mer, Pflegeleitungszimmer, OP-Bereiche, Verwaltungsarbeitsplätze, Sekretariate.
A.III.2.4.b	Alle Rechnersysteme sind vernetzt. Die Netzbasisinfrastruktur muss hinreichend und durchgängig leistungsfähig sein.

Physische Werkzeugebene



Literaturverzeichnis

- [ABOWD et al. 1995] ABOWD, G., R. ALLEN and D. GARLAN (1995): Formalizing style to understand descriptions of software architecture. *ACM Transactions on Software Engineering and Methodology*, 4(4):319–364.
- [ALEXANDER 1977] ALEXANDER, C. (1977): *A pattern language : towns, buildings, construction*. New York: Oxford University Press.
- [ALLEN 2001] ALLEN, D. W. (2001): Establishing an EAI Architecture. *eAI Journal*, pp. 49–50.
- [APPLE COMPUTER INC. 1987] APPLE COMPUTER INC. (1987): *Apple Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley.
- [BACHMANN et al. 2000] BACHMANN, F., L. BASS, J. CARRIERE, P. CLEMENTS, D. GARLAN, J. IVERS, R. NORD and R. LITTLE (2000): *Software Architecture Documentation in Practice: Documenting Architectural Layers*. Special Report: CMU/SEI-2000-SR-004.
- [BALZERT 1996] BALZERT, H. (1996): *Lehrbuch der Software-Technik, vol. 1: Software-Entwicklung*. Heidelberg: Spektrum Akademischer Verlag.
- [BECKER and SCHÜTTE 1996] BECKER, J. and R. SCHÜTTE (1996): *Handelsinformationssysteme*. Landsberg: mi Wissenschaft.
- [BECKER et al. 2000] BECKER, J., R. SCHÜTTE, T. GEIB and H. IBERSHOFF (2000): Grundsätze ordnungsmäßiger Modellierung (GoM). Sachbericht des Forschungsprojektes „Grundsätze ordnungsmäßiger Modellierung“ mit dem Förderkennzeichen 01 IS 604 # des Bundesministeriums für Bildung und Forschung. ISBN: 3-540-65993-5.
- [BERRY and REEVES 1992] BERRY, R. E. and C. J. REEVES (1992): The evolution of the Common User Access Workplace Model. *IBM Systems Journal*, 31(3):414–428.
- [BLOCH and BRUCE 2004] BLOCH, B. and I. BRUCE (2004): Working With Web Services and MOM: The Best of Both Worlds. *Business Integration Journal*, pp. 76–79.
- [BPMI BPML WG 2002] BPMI BPML WG (2002): *Business Process Modeling Language*. Last Call Draft of the BPML specification.
- [BPMI NOTATION WG 2004] BPMI NOTATION WG (2004): *Business Process Modeling Notation (BPMN)*. Version 1.0.
- [BRIGL et al. 2004] BRIGL, B., A. HÄBER, T. WENDT and A. WINTER (2004): Ein 3LGM² Modell des Krankenhausinformationssystems des Universitätsklinikums Leipzig und seine Verwertbarkeit für das Informationsmanagement. In REBSTOCK, M., ed.: *Modellierung betrieblicher Informationssysteme - MobIS 2004*, vol. P-45 of *GI-Edition: Lecture Notes in Informatics*, pp. 21–41.
- [BRIGL et al. 2003] BRIGL, B., T. WENDT and A. WINTER (2003): Modeling interdependencies between business and communication processes in hospitals. In BAUD, R., M. FIESCHI, P. L. BEUX and P. RUCH, eds.: *Proceedings of MIE2003*, vol. 95 of *Studies in Health Technology and Informatics*. Amsterdam: IOS.
- [CCOW 2000a] CCOW (2000a): *Health Level-Seven Standard Context Management Specification, Data Definition: Patient Subject*. Version CM-1.2.
- [CCOW 2000b] CCOW (2000b): *Health Level-Seven Standard Context Management Specification, Data Definition: User Subject*. Version CM-1.2.

- [CCOW 2000c] CCOW (2000c): Health Level-Seven Standard Context Management Specification, Technology- and Subject-Independent Component Architecture. Version CM-1.2.
- [CEN TC251 1997] CEN TC251 (1997): Healthcare Information System Architecture Part 1 (HISA). CEN-Standard ENV 12967-1.
- [CLEGG 2000] CLEGG, C. W. (2000): Sociotechnical principles for sytem design. *Applied ergonomics*, 31:463–477.
- [DIN 1988] DIN (1988): Bildschirmarbeitsplätze: Grundsätze ergonomischer Dialoggestaltung. Berlin. Beuth. DIN 66 234, Teil 8.
- [DoD DISA 1996] DoD DISA (1996): Department of Defense Technical Architecture Framework for Information Management, Volume 1: Overview. Version 3.0.
- [DROSDOWSKI 1994] DROSDOWSKI, G., ed. (1994): Duden, Das große Fremdwörterbuch. Mannheim: Dudenverlag.
- [EMMERICH 2003] EMMERICH, W. (2003): Konstruktion von verteilten Objekten. dpunkt.verlag.
- [FERSTL and SINZ 1995] FERSTL, O. K. and E. J. SINZ (1995): Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. *Wirtschaftsinformatik*, 37(3):209–220.
- [FERSTL and SINZ 2001] FERSTL, O. K. and E. J. SINZ (2001): Grundlagen der Wirtschaftsinformatik. München: Oldenbourg, 4. Auflage ed. Band 1.
- [GAMMA et al. 1997] GAMMA, E., R. HELM, R. JOHNSON and J. VLISSIDES (1997): Entwurfsmuster. München: Addison-Wesley.
- [GARLAN 1995] GARLAN, D. (1995): What is style? In *Proceedings of Dagshtul Workshop on Software Architecture*, pp. 169–183.
- [GARLAN et al. 1997] GARLAN, D., R. T. MONROE and D. WILE (1997): ACME: An Architecture Description Interchange Language. In *Proceedings of CASCON '97*, pp. 169–183.
- [GARLAN and SHAW 1994] GARLAN, D. and M. SHAW (1994): Advances in Software Engineering and Knowledge Engineering, chap. An Introduction to Software Architecture, pp. 1–39. Series on Software Engineering and Knowledge Engineering (2). Singapore: World Scientific Publishing Company.
- [HASSELBRING 2000] HASSELBRING, W. (2000): Information System Integration. *Communications of the ACM*, 43(6):32–36.
- [HAUX et al. 1998] HAUX, R., A. LAGEMANN, P. KNAUP, P. SCHMÜCKER and A. WINTER (1998): Management von Informationssystemen. Stuttgart: Teubner.
- [HAUX et al. 2004] HAUX, R., A. WINTER, E. AMMENWERTH and B. BRIGL (2004): Strategic Information Management in Hospitals: An Introduction to Hospital Information Systems. Springer.
- [HAYNE and POLLARD 2000] HAYNE, S. C. and C. E. POLLARD (2000): A comparative analysis of critical issues facing Canadian information systems personnel: a national and global perspective. *Information & Management*, 38:73–86.
- [HEIMBIGNER and MCLEOD 1985] HEIMBIGNER, D. and D. MCLEOD (1985): A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems*, 3(3):253–278.
- [HILLIARD 2000] HILLIARD, R. (2000): IEEE-Std-1471-2000 IEEE-Std-1471-2000: Recommended Practice for Architectural Description of Software-Intensive Systems. Internet: <http://www.enterprise-architecture.info/Images/Documents/IEEE%201471-2000.pdf>.
- [HIMSS/RSNA 2003a] HIMSS/RSNA (2003a): IHE IT Infrastructure Technical Framework, Volume 1 (ITI TF-1): Integration Profiles. Revision 1.0.
- [HIMSS/RSNA 2003b] HIMSS/RSNA (2003b): IHE Technical Framework, Volume I: Integration Profiles. Revision 5.5.

- [HL7 1999] HL7 (1999): Health Level Seven (HL7). Version 2.3.1; Standard der HL7-Organisation; erhältlich für Mitglieder u. a. über die HL7-Benutzergruppe Deutschland.
- [IDS SCHEER AG 2004] IDS SCHEER AG (2004): ARIS Rahmenkonzept. Präsentationsfoliensatz.
- [IEEE 802.11 WG 2003] IEEE 802.11 WG (2003): 802.11. Local and metropolitan area networks—Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE-Standard 802.11, 1999 Edition (R2003).
- [IEEE 802.3 WG 2002] IEEE 802.3 WG (2002): 802.3. Local and metropolitan area networks—Specific requirements. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE-Standard 802.3-2002.
- [IMBI/MI 2001] IMBI/MI (2001): Anforderungskatalog für die Informationsverarbeitung im Krankenhaus. Version 1.0b.
- [INFORMATION SCIENCES INSTITUTE 1981a] INFORMATION SCIENCES INSTITUTE (1981a): Internet Protocol. RFC 791; erstellt für die Defense Advanced Research Projects Agency durch das Information Sciences Institute, University of Southern California.
- [INFORMATION SCIENCES INSTITUTE 1981b] INFORMATION SCIENCES INSTITUTE (1981b): Transmission Control Protocol. RFC 793; erstellt für die Defense Advanced Research Projects Agency durch das Information Sciences Institute, University of Southern California.
- [INGENERF and DIEDRICH 1997] INGENERF, J. and T. DIEDRICH (1997): Notwendigkeit und Funktionalität eines Terminologieservers in der Medizin. Künstliche Intelligenz, 97(3):6–14. Sonderheft „Wissensbasierte Systeme in der Medizin“.
- [ISO JTC 1 1993] ISO JTC 1 (1993): Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 1: Systems. ISO-Standard ISO/IEC 11172-1:1993(E).
- [ISO JTC 1 1998] ISO JTC 1 (1998): Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1. ISO-Standard ISO/IEC 8859-1:1998(E).
- [ISO TC 130 2004] ISO TC 130 (2004): Graphic technology – Prepress digital data exchange – Tag image file format for image technology (TIFF/IT). ISO-Standard ISO 12639:2004(E).
- [ISO/IEC JTC 1 1995] ISO/IEC JTC 1 (1995): ISO/IEC DIS 13235 - ODP Trading Function. ISO-Standard ISO/IEC DIS 13235 (Final Draft: 20 June, 1995).
- [ISO/IEC JTC 1 1996a] ISO/IEC JTC 1 (1996a): Information technology — Open Distributed Processing — Reference model: Architecture. ISO-Standard ISO/IEC 10746-3: 1996(E).
- [ISO/IEC JTC 1 1996b] ISO/IEC JTC 1 (1996b): Information technology — Open Distributed Processing — Reference model: Foundations. ISO-Standard ISO/IEC 10746-2: 1996(E).
- [ISO/IEC JTC 1 1996c] ISO/IEC JTC 1 (1996c): Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model. Standard ISO/IEC 7498-1: 1994(E).
- [ISO/IEC JTC 1 1998a] ISO/IEC JTC 1 (1998a): Information technology — Open Distributed Processing — Reference model: Architectural semantics. ISO-Standard ISO/IEC 10746-4: 1998(E).
- [ISO/IEC JTC 1 1998b] ISO/IEC JTC 1 (1998b): Information technology — Open Distributed Processing — Reference model: Overview. ISO-Standard ISO/IEC 10746-1: 1998(E).
- [JURIC 2002] JURIC, M. B. (2002): EAI and Web Services. eAI Journal, pp. 31–35.
- [KELLER et al. 1992] KELLER, G., M. NÜTTGENS and A.-W. SCHEER (1992): Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes. Heft 89.

- [LANGNER et al. 1997] LANGNER, P., C. SCHNEIDER and J. WEHLER (1997): Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen. *Wirtschaftsinformatik*, 39(5):479–489.
- [LEHMANN and MEYER ZU BEXTEN 2002] LEHMANN, T. M. and E. MEYER ZU BEXTEN (2002): *Handbuch der Medizinischen Informatik*. München: Hanser.
- [LEINER et al. 1997] LEINER, F., W. GAUS and R. HAUX (1997): *Medizinische Dokumentation*. Stuttgart: Schattauer. 2. Auflage.
- [LINTHICUM 2000a] LINTHICUM, D. S. (2000a): Application Servers and EAI. *eAI Journal*.
- [LINTHICUM 2000b] LINTHICUM, D. S. (2000b): Enterprise Application Integration. Addison-Wesley. Internet: <http://dbs.uni-leipzig.de/buch/index.html>.
- [LONGO 2001] LONGO, J. R. (2001): The ABCs of Enterprise Application Integration. *eAI Journal*, pp. 56–58.
- [LUBLINSKY 2002] LUBLINSKY, B. (2002): Approaches to B2B Integration. *eAI Journal*, pp. 38–47.
- [LUBLINSKY and FARREL JR. 2001] LUBLINSKY, B. and M. FARREL JR. (2001): Enterprise Architecture and J2EE. *eAI Journal*, pp. 38–43.
- [LUTZ 2000] LUTZ, J. C. (2000): EAI Architecture Patterns. *eAI Journal*, pp. 64–73.
- [NEEDLEMAN and WUNSCH 1970] NEEDLEMAN, S. B. and C. D. WUNSCH (1970): A general method applicable to the search for similarities in the amino acid sequences of two proteins. *Journal of Molecular Biology*, 48(3):444–453.
- [NETWORK WORKING GROUP 1989] NETWORK WORKING GROUP (1989): Requirements for Internet Hosts – Communication Layers. RFC 1122.
- [NETWORK WORKING GROUP 1999] NETWORK WORKING GROUP (1999): Hypertext Transfer Protocol – HTTP/1.1. RFC 2616.
- [NIEMANN et al. 2002] NIEMANN, H., W. HASSELBRING, T. WENDT, A. WINTER and M. MEIERHOFER (2002): Kopplungsstrategien für Anwendungssysteme im Krankenhaus. *Wirtschaftsinformatik*, 44(75):425–434.
- [OLSEN 1995] OLSEN, P. S. (1995): Aspects of Integration in HIS. *International Journal of Bio-Medical Computing*, 39:53–57.
- [OMG 1997] OMG (1997): A Discussion of the Object Management Architecture.
- [OMG 2000a] OMG (2000a): Air Traffic Control Specification. Version 1.0.
- [OMG 2000b] OMG (2000b): Currency Specification. Version 1.0.
- [OMG 2000c] OMG (2000c): Internationalization, Time Operations, and Related Facilities. Version 1.0.
- [OMG 2000d] OMG (2000d): Meta Object Facility (MOF) Specification. Version 1.3.
- [OMG 2000e] OMG (2000e): Mobile Agent Facility Specification. Version 1.0.
- [OMG 2000f] OMG (2000f): Trading Object Service Specification. Version 1.0.
- [OMG 2001a] OMG (2001a): Biomolecular Sequence Analysis Specification. Version 1.0.
- [OMG 2001b] OMG (2001b): Clinical Observations Access Service Specification. Version 1.0.
- [OMG 2001c] OMG (2001c): Person Identification Service (PIDS) Specification. Version 1.1.
- [OMG 2001d] OMG (2001d): Resource Access Decision Facility Specification. Version 1.0.
- [OMG 2002a] OMG (2002a): CORBA Components. Version 3.0.

- [OMG 2002b] OMG (2002b): Genomic Maps Specification. Version 1.0.
- [OMG 2002c] OMG (2002c): Persistent State Service Specification. Version 2.0.
- [OMG 2002d] OMG (2002d): Security Service Specification. Version 1.8.
- [OMG 2003a] OMG (2003a): Gene Expression Specification. Version 1.1.
- [OMG 2003b] OMG (2003b): MDA Guide. Version 1.0.1.
- [OMG 2003c] OMG (2003c): OMG Unified Modeling Language Specification. Version 1.5 (formal/03-03-01).
- [OMG 2004] OMG (2004): CORBA. Version 3.0.3.
- [PEARSON 2001] PEARSON, W. R. (2001): Protein sequence comparison and Protein evolution (Tutorial - ISMB2000). Internet: <http://www.people.virginia.edu/~wrp/papers/ismb2000.pdf>.
- [RAHM 1994] RAHM, E. (1994): Mehrrechner-Datenbanksysteme: Grundlagen der verteilten und parallelen Datenbankverarbeitung. Bonn: Oldenbourg. Internet: <http://dbs.uni-leipzig.de/buch/index.html>.
- [SAP 2005] SAP (2005): HCM. Systemdokumentation zu SAP R/3 IS-H.
- [SCHEER 1998a] SCHEER, A.-W. (1998a): ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. Berlin: Springer, 3. Auflage ed. Band 2.
- [SCHEER 1998b] SCHEER, A.-W. (1998b): ARIS - Vom Geschäftsprozeß zum Anwendungssystem. Berlin: Springer, 3. Auflage ed. Band 1.
- [SCHEER et al. 1995] SCHEER, A.-W., M. NÜTTGENS and V. ZIMMERMANN (1995): Rahmenkonzept für ein integriertes Geschäftsprozeßmanagement. *Wirtschaftsinformatik*, 37(5):426–434.
- [SCHMIDT 2002] SCHMIDT, J. (2002): EAI Principles, Part 3. *eAI Journal*, p. 48.
- [SCHÜTTE 1998] SCHÜTTE, R. (1998): Grundsätze ordnungsmässiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden: Gabler.
- [SHAW and GARLAN 1995] SHAW, M. and D. GARLAN (1995): Computer Science Today: Recent Trends and Developments, chap. Formulations and Formalisms in Software Architecture, pp. 307–323. *Lecture Notes in Computer Science* (1000). Springer.
- [SHAW and GARLAN 1996] SHAW, M. and D. GARLAN (1996): Software Architecture. Upper Saddle River, New Jersey: Prentice Hall.
- [SPEWAK and HILL 1992] SPEWAK, S. H. and S. C. HILL (1992): Enterprise Architecture Planning. New York: John Wiley & Sons.
- [STYLIANOU and KUMAR 2000] STYLIANOU, A. C. and R. L. KUMAR (2000): An Integrative Framework for IS Quality Management. *Communications of the ACM*, 43(9):99–104.
- [THE OPEN GROUP 2001] THE OPEN GROUP (2001): The Open Group Architectural Framework. Internet: <http://www.opengroup.org/architecture/togaf7/index7.htm>. Version 7.
- [THE OPEN GROUP 2003] THE OPEN GROUP (2003): The Open Group Architectural Framework. Internet: <http://www.opengroup.org/architecture/togaf8/index8.htm>. Version 8.
- [VAN BEMMEL and MUSEN 1997] VAN BEMMEL, J. H. and M. A. MUSEN, eds. (1997): Handbook of Medical Informatics. Heidelberg: Springer.
- [VOSSEN 2000] VOSSEN, G. (2000): Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. München: Oldenbourg. 4. Auflage.
- [W3C 2004] W3C (2004): Web Services Description Language (WSDL). Version 2.0, Working Draft 3, August 2004.

- [WASHBURN and EVANS 1994] WASHBURN, K. and J. EVANS, eds. (1994): TCP/IP. Bonn: Addison-Wesley.
- [WENDT et al. 2004] WENDT, T., A. HÄBER, B. BRIGL and A. WINTER (2004): Modeling Hospital Information Systems (Part 2): Using the 3LGM² Tool for Modeling Patient Record Management. *Methods of Information in Medicine*, 43(3):256–267.
- [WG 2001] WG, IEEE 802.1 (2001): IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture. IEEE-Standard 802®-2001.
- [WINTER et al. 2001] WINTER, A., B. BRIGL and T. WENDT (2001): A UML-based Ontology for Describing Hospital Information System Architectures. In PATEL, V. L., R. ROGERS and R. HAUX, eds.: *Medinfo 2001*, pp. 778–782. Amsterdam: IOS.
- [WINTER et al. 2003] WINTER, A., B. BRIGL and T. WENDT (2003): Modeling Hospital Information Systems (Part 1): The Revised Three-layer Graph-based Meta Model 3LGM². *Methods of Information in Medicine*, 42(5):544–551.
- [WINTER and HAUX 1995] WINTER, A. and R. HAUX (1995): A Three-Level Graph-Based Model for the Management of Hospital Information Systems. *Methods of Information in Medicine*, 34(4):378–396.
- [WINTER et al. 1999] WINTER, A., A. WINTER, K. BECKER, O. BOTT, B. BRIGL, S. GRÄBER, W. HASSELBRING, R. HAUX, C. JOSTES, O.-S. PENGER, H.-U. PROKOSCH, J. RITTER, R. SCHÜTTE and A. TERSTAPPEN (1999): Referenzmodelle für die Unterstützung des Managements von Krankenhausinformationssystemen. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie*, 30(4):173–189.
- [WINTER et al. 1998] WINTER, A., R. ZIMMERLING, O. J. BOTT, S. GRÄBER, W. HASSELBRING, R. HAUX, A. HEINRICH, R. JAEGER, I. KOCK, D. P. F. MÖLLER, O.-S. PENGER, J. RITTER, A. TERSTAPPEN and A. WINTER (1998): Das Management von Krankenhausinformationssystemen: Eine Begriffsdefinition. *Informatik, Biometrie und Epidemiologie in Medizin und Biologie*, 29(2):93–105.
- [X/OPEN 1995] X/OPEN (1995): Motif Toolkit API. X/Open Document Number: C320.
- [ZACHMAN 1987] ZACHMAN, J. A. (1987): A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3).
- [ZACHMAN 1999] ZACHMAN, J. A. (1999): A Framework for Information Systems Architecture. *IBM Systems Journal*, 38(2&3):454–470. Reprint.

Stichwortverzeichnis

1 . . . 9

- 3LGM² 2, 3, 26, 28, 87–97, 192
 - Überarbeitung des ~ 99–105
 - fachliche Ebene 87, **88**, 100
 - Hauptklassen des ~ 87–92
 - Klassen für die Integrationsmodellierung 92–95
 - Klassen für die Prozessmodellierung 95
 - logische Werkzeugebene 87, **88**, 100
 - physische Werkzeugebene 87, **92**
- 3LGM_A² **99**, 99–114, 192–193
 - fachliche Ebene 100, 105, 121, 131
 - logische Werkzeugebene 103, 121, 131, 133
 - physische Werkzeugebene 121
- 3LGM 87
- 802.11x *siehe* IEEE
- 802.3 *siehe* IEEE
- 802.X *siehe* IEEE

A

- Abhängigkeit 157–169, 195
 - Ausführungs~ 160–161
 - funktionale ~ 159–160
 - ~sgrad
 - Ausführungs~ 160–161, 168, 169, 195
 - funktionaler ~ 159–160, 168, 169, 195
 - informationaler ~ 159, 167, 168, 195
 - Transaktions~ 161, 168, 169, 195
 - informationale ~ 159
 - transaktionale ~ 161
- ACME 27
- Aktion 22
- Altsystem 60
- American Standard Code for Information Interchange 34, 52
- Anforderung 2
 - Integrations~ 1, 3, 11–19, 35, 38, 46, 48, 53, 54, 129–156, 187–190, 192, 194
 - Erfüllung von ~en **143**, 129–156
 - ~skategorie 1, **12**
- Anforderungsdomäne *siehe* Domäne
- Anwendungsbaustein 12
- Anwendungsfall 100
- Anwendungsmanagement 22

- Anwendungsserver 61
- Anwendungssystem 12
- Anwendungsszenario 130–133, 163–169, 184–186
- Apple Human Interface Guidelines 54
- application server *siehe* Anwendungsserver
- Architektur 21, **24**, 99
 - ~bewertung
 - Grundannahmen der ~ 8
 - ~modell *siehe* Modell
 - ~qualität 7
 - ~referenzmodell *siehe* Modell
 - ~standard *siehe* Standard
 - ~stil 2, 22, 24, **27**, 28, 30, 95, 97, 99
- Architektur eines Informationssystems *siehe* Architektur
- Architektur integrierter Informationssysteme 28, 56–59, 75, 190
 - ARIS-Informationsmodell *siehe* ARIS-Metamodell
 - ARIS-Metamodell 59
 - Metamodell 72, 87
 - Sichten (Sichtweisen) 56
- Architektur-Referenzmodell *siehe* Modell
- Architekturmodell *siehe* Modell
- Architekturmodell im engeren Sinn *siehe* Modell
- Architekturmodell im weiteren Sinn *siehe* Modell
- Architekturmuster für EAI *siehe* Muster
- Architekturstandard *siehe* Standard
- Architekturstil *siehe* Architektur
- ARIS *siehe* Architektur integrierter Informationssysteme
- ASCII *siehe* American Standard Code for Information Interchange
- Aufrufdomäne *siehe* Domäne
- Ausrichtung 171–177, 179, 184, 186
- Aussage 22

B

- Begriffssystem **17**, 113–114, 136, 192
- Bewertungsziel 7
- Beziehung 23, 29, 30

BPML *siehe* Business Process Modeling Language
 Business Process Modeling Language 79–81

C

Carnegie Mellon University 22
 School of Computer Science 22
 CCOW *siehe* Clinical Context Object Workgroup
 CEN *siehe* Europäisches Komitee für Normung
 Clinical Context Object Workgroup *siehe* Health Level Seven
 COM *siehe* Component Object Model
 Comité Européen de Normalisation *siehe* Europäisches Komitee für Normung
 Common Object Request Broker Architecture *siehe* Object Management Architecture
 Common User Application 54
 Component Model 40, 72
 Component Object Model 53, 62
 Consistent Time *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Consistent Time Profile
 Continuum *siehe* The Open Group Architectural Framework
 CORBA *siehe* Common Object Request Broker Architecture
 CUA *siehe* Common User Application

D

Daten 11
 Datenbanksystem 157–159
 Datendomäne *siehe* Domäne
 Datenintegration *siehe* Integration
 Datenmodell
 konzeptionelles ~ 17
 logisches ~ 17
 objektorientiertes ~ 17, 62
 relationales ~ 17, 62
 Datenmodellierung
 semantische ~ 17
 DCOM *siehe* Distributed Component Object Model
 Detaillierung 179, 183
 Deutsches Institut für Normung 54
 Dienst 33, 40, 42, 46, **106**, 105–107
 Anwendungs~ 106, 107
 ~spezifikation 106
 Vermittlungs~ 106, **106**, 107
 Dienstkategorie *siehe* Kategorie
 DIN *siehe* Deutsches Institut für Normung

~ 66234 54
 Distributed Component Object Model 53, 62, 106
 Dokumentation 2
 Domäne 22, 129–156, 194
 Übermittlungs~ 140–142, 150, 152
 Anforderungs~ 129, 133–140, 194
 Aufruf~ 142–143, 147, 154
 Daten~ 133–134, 136, 143, 194
 funktionale ~ 134–136, 145, 194
 Kommunikations~ 129, 140–143, 145, 147, 194
 Kontext~ 136–138, 150, 194
 Präsentations~ 138–140, 194
 semantische ~ 136, 147, 194
 Zugriffs~ 140, 152, 154, 194

E

EAI *siehe* Enterprise Application Integration
 EAP *siehe* Enterprise Application Planning
 Ebene
 ~n des 3LGM² 87
 EAI-~n 60
 ~n des EAP 68, 69
 ~n des HIF 46
 Einarbeitungsaufwand 2
 EJB *siehe* Enterprise Java Beans
 Elementhierarchie 119
 EN
 ~ 12967-1 46
 ~ 12443 46
 Enterprise Application Integration 54, 60–64, 75, 190
 ~-Technologie 61
 Enterprise Application Planning 68–72
 Enterprise Architecture Planning 75
 Enterprise Continuum *siehe* The Open Group Architectural Framework
 Enterprise Java Beans 62
 Enterprise User Authentication *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Enterprise User Authentication
 Entität 22
 Entwurfsmuster 24, **24**, 27
 EPK *siehe* Ereignisgesteuerte Prozessketten
 Ereignis 72, 77, 79, 82
 ~gesteuerte Prozessketten 32, 77–79, 99
 ~typ 41, 79, 83, 99–101, 104, 143
 Ereignisgesteuerte Prozessketten *siehe* Ereignis
 EUA *siehe* Enterprise User Authentication

Europäisches Komitee für Normung 31, 46

F

Fehlertoleranz 11
 File Transfer Protocol 35, 52
 Framework for Information Systems Architecture *siehe* Zachman-Rahmenwerk
 FTP *siehe* File Transfer Protocol
 Funktion 38, 77
 funktionale Domäne *siehe* Domäne
 funktionale Integration *siehe* Integration

G

Genomanalyse 171
 Geschäftsprozess 59, 77–83, 100
 Gleichheitswert
 absoluter ~ **172, 177**, 171–178
 kostenbewerteter ~ **181**, 181–182
 relativer ~ **179**, 179–180
 Grundannahme 8
 Grundsätze
 ~ ergonomischer Dialoggestaltung für Bildschirmarbeitsplätze 54
 ~ ordnungsgemäßer Modellierung 179
 Gruppe 22

H

Hülle 63
 Health Level Seven 106
 Clinical Context Object Workgroup 53
 Health Level Seven 35, 100, 101
 ~ Standard Context Management Specification 53
 Healthcare Information Framework 46–49, 72, 74, 190, 191
 Ebenen des ~ 46
 Healthcare Information Systems Architecture 46, 48, 72, 106, 108–110
 Healthcare Information Systems Architecture *siehe* Healthcare Information Framework
 Heterogenität 171–186
 ~sgrad 171–184
 absoluter ~ **175, 178**, 171–178, 195
 kostenbewerteter ~ **182**, 181–184, 195
 relativer ~ **180**, 179–180, 195
 Heterogenitätsgrad *siehe* Heterogenität
 Hierarchie *siehe* Elementhierarchie
 HIF *siehe* Healthcare Information Framework
 HISA *siehe* Healthcare Information Systems Architecture

HL7 *siehe* Health Level Seven
 HTML 52
 HTTP *siehe* Hypertext Transfer Protocol
 Hypertext Transfer Protocol 35, 61

I

IDL *siehe* Interface Definition Language
 IEEE *siehe* Institute of Electrical and Electronics Engineers
 ~ 802.11x 34
 ~ 802.3 34
 ~ 802.X 14, 34, 52
 ~ 1471 21
 IETF *siehe* Internet Engineering Task Force
 IHE *siehe* Integrating the Healthcare Enterprise
 Information **11**, 26
 ~smodell *siehe* Modell
 ~s-Referenzmodell *siehe* Modell
 Informations-Referenzmodell *siehe* Modell
 Informationsmodell *siehe* Modell
 Informationssystem 12
 Informationssystemkomponente 2
 Institute of Electrical and Electronics Engineers 34
 Integrating the Healthcare Enterprise 12
 ~ Integrationsprofile 13
 Consistent Time Profile 14
 Enterprise User Authentication 19
 Patient Identifier Cross-referencing 16
 Patient Synchronized Applications 17
 Retrieve Information for Display 15
 Integration 1, 2, **12**
 ~sanforderung *siehe* Anforderung
 Daten~ 1, 12, **14**, 35, 46, 143–145, 188
 funktionale ~ 1, 12, **15**, 46, 53, 145, 188, 189
 horizontale ~ 60
 Kontext~ 2, 12, **16**, 53, 150–152, 188, 189
 physische ~ 12, **14**, 35, 129, 188
 Präsentations~ 12, **18**, 54, 152, 188, 189
 ~squalität 7
 semantische ~ 12, **16**, 53, 113, 147–150, 188, 189, 192
 ~technik 2
 ~technik 2, 3, 188–196
 Zugriffs~ 12, **19**, 152–156, 188, 189
 Integrationstechnik *siehe* Integration
 Interaktion 35, 40
 Interface Definition Language 41
 International Organization for Standardization 22

Internationale Organisation für Standardisierung 31, 32
Internet 61
Internet Engineering Task Force 34
Internet Protocol 34, 52
Interpretation 17
IP *siehe* Internet Protocol
ISO *siehe* Internationale Organisation für Standardisierung
~ 9241-10 54
ISO/IEC
~ 10746-1 35, 36
~ 10746-2 22, 35, 36, 72, 87
~ 10746-3 22, 35, 37
~ 10746-4 35, 37
~ 13235-1 38, 39
~ 7498-1 33
~ 7498-2 33
~ 7498-3 33
~ 7498-4 33
~ 10918 52

J

Java Database Connectivity 62
Java Server Pages 61
JDBC *siehe* Java Database Connectivity
JPEG 52
JSP *siehe* Java Server Pages

K

Kategorie
Dienst~n 47, 51, 52
Schnittstellen~n 41
~n von Transaktionsausführungen *siehe* Transaktion
Kategorien von Integrationsanforderungen *siehe* Anforderungskategorien
Kerberos Network Authentication Service 35
Kommunikation
~sdomäne *siehe* Domäne
nachrichtenbasierte ~ 2
~sverbinding **107**, 126–127, 171–186
Formalisierung von ~en 126–127
Kommunikationsdomäne *siehe* Domäne
Kommunikationsserver 61, 63, 111–112, 184, 186
Kommunikationsstandard *siehe* Standard
Komplexität 8
Komponente 23, 29, 30, 40, 41, 95, 99, 105
Komponententyp 27, **27**, 28, 96
Komposition 22

Konfiguration 22, 27, **27**, 29, 30, 96
Konnektor 95
Konnektorentyp 27, **27**, 28, 96
Kontextdomäne *siehe* Domäne
Kontextintegration *siehe* Integration
Kontextmanager 53
Kontextsynchronisierung 2
konzeptionelles Datenmodell *siehe* Datenmodell
Kosten
~ von Kommunikationsbeziehungen **181**, 182
~ von Kommunikationsverbindungen 181

L

legacy system *siehe* Altsystem
Logik **17**
logisches Datenmodell *siehe* Datenmodell

M

Managementinformation 22
Master Patient Index 63
MDA Guide *siehe* Model Driven Architecture Guide
Mengensymbole 120
message broker *siehe* Kommunikationsserver
message-oriented middleware *siehe* nachrichtenorientierte Middleware
Metamodell *siehe* Modell
Middleware
nachrichtenorientierte 61
Model Driven Architecture Guide 43, 45
Sichtweisen 43
Modell 26
Architektur~ **29**, 99
~ im engeren Sinn 29
~ im weiteren Sinn 29
Informations~ **29**, 47
Meta~ **26**, 28, 40, 59, 72, 87, 97
Referenz~ **26**, 28
Architektur~ 21, 29, **29**, 32, 33, 46, 51, 62, 72
Informations~ 28, **29**, 47
Prozess~ 100
Vorgehens~ 21, 28, **29**, 31, 43, 50, 51, 64, 65, 68
semantisches ~ 27, 28, 30, 95, 96
Transformation von ~en 44
MOM *siehe* message-oriented middleware
Motif 54
Moving Picture Experts Group 35, 52

MPEG *siehe* Moving Picture Experts Group
 Muster 24, **25**, 27, 29, 30
 Architektur~ für EAI 62

N

nachrichtenbasierte Kommunikation *siehe* Kommunikation
 nachrichtenorientierte Middleware *siehe* Middleware
 Nachrichtentyp 105
 Transformation von ~en 105
 Namensgraph 22
 Namensraum 22
 Needleman-Wunsch-Algorithmus 171, **172**, 174
 NW-Algorithmus *siehe* Needleman-Wunsch-Algorithmus

O

Object Management Architecture 31, 32, 40–46, 74, 106, 110–111, 190, 191
 Common Object Request Broker Architecture 40, **42**, 45, 52, 53, 62, 106, 184
 ~ Reference Model 40, **41**, 72
 Object Management Group 28, 40
 ~ Component Model 99, 100
 ~ Object Model 99, 102
 Object Model 28, 40, **40**, 72
 Object Request Broker 2, **41**, 62, 184, 186
 Objekt 22, 35, 40, 41, 72, 99
 initiiierendes ~ 22
 objektorientiertes Datenmodell *siehe* Datenmodell
 ODBC *siehe* Open DataBase Connectivity
 Offenheit 11
 OMA *siehe* Object Management Architecture
 OMG *siehe* Object Management Group
 OMG IDL *siehe* Interface Definition Language
 OMG OMA *siehe* Object Management Architecture
 Open DataBase Connectivity 62
 Open Group *siehe* The Open Group
 Operation 40, 99, 102–105
 ORB *siehe* Object Request Broker
 OSI-Referenzmodell 22, 32–35, 44, 74, 190, 191
 Schichten des ~s 33

P

Patient Identifier Cross-referencing *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Patient Identifier Cross-referencing
 Patient Synchronized Applications *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Patient Synchronized Applications
 Phase 31
 ~modell der ARIS 59
 ~n des EAP 68–72
 ~n der TOGAF ADM 50, 65
 physische Integration *siehe* Integration
 PIX *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Patient Identifier Cross-referencing
 Plattform 43–45
 Portabilität 22
 Prädikat 121–126
 Präsentationsdomäne *siehe* Domäne
 Präsentationsintegration *siehe* Integration
 Protokoll 34

R

Rahmenwerk 21, 31–73, 190
 Übersicht über ~e 73–75
 technische ~e 32–53, 190, 191
 unternehmensbezogene ~e 54–69, 190
 Reference Model for Open Distributed Processing *siehe* Referenzmodell für offene verteilte Informationsverarbeitung
 Referenzmodell *siehe* Modell
 Referenzmodell für die Kopplung offener Systeme *siehe* OSI-Referenzmodell
 Referenzmodell für offene verteilte Informationsverarbeitung 35–39, 44, 74, 87, 99, 100, 190, 191
 Trading Function 38, **39**
 relationales Datenmodell *siehe* Datenmodell
 Remote Function Call 2
 Request for Comments 34
 791 34
 793 34
 1122 34
 Retrieve Information for Display *siehe* Integrating the Healthcare Enterprise Integrationsprofile: Retrieve Information for Display
 RFC *siehe* Remote Function Call, *siehe* Request for Comments
 RM-ODP *siehe* Referenzmodell für offene verteilte Informationsverarbeitung

Rolle 31

S

Satz 22

Schicht 33, 44

~en des OSI-Referenzmodells *siehe* OSI-Referenzmodell

Schnittstelle 22, 35, 40, 72, 99, 102–105

Schnittstellenkategorie *siehe* Kategorie

Schnittstellentyp 27

School of Computer Science *siehe* Carnegie Mellon University

Semantik 27, 30, 96

semantische Datenmodellierung *siehe* Datenmodellierung

semantische Domäne *siehe* Domäne

semantische Integration *siehe* Integration

semantische Integritätsbedingungen 17

Semantisches Objektmodell 81

Sicht 21, 43

~en der ARIS 56

Sichtweise 31, 36, 43, 56

~n der ARIS *siehe* Sichten der ARIS

~n des MDA Guide 43, 44

~n des RM-ODP 37

~n des Zachman-Rahmenwerkes 56

Skalierbarkeit 11

SOM *siehe* Semantisches Objektmodell

Spezifikation

funktionale ~ 105

informationale ~ 105

Standard 32

Architektur~ 2, 3, 190–193

Kommunikations~ 35, 105

Standard Context Management Specification *siehe* Health Level Seven

Stufe 31

Subtyp 22

Synchronitätsmodus 104

Syntax 25, 27, 95

T

TAFIM *siehe* Technical Architecture Framework for Information Management

Tagged Image File Format 35, 52

TCP *siehe* Transmission Control Protocol

Technical Architecture Framework for Information Management 49

Telnet 35

Terminologieserver 53

The Open Group 31, 49

The Open Group Architectural Framework 21, 31, 49–53, 65, 66, 68, 74, 75, 192

~ Architecture Development Method 49, 50, 65

Phasen der ~ 50, 65

~ Enterprise Continuum 65, 66

~ Foundation Architecture 49, 68

~ Integrated Information Infrastructure Reference Model 68, 68

~ Resource Base 65

~ Standards Information Base 51, 52, 68

~ Technical Reference Model 46, 51, 68, 72

Dienstkategorien des ~ 51, 52

Einheiten des ~ 51

~ Version 7 49–53, 190

~ Version 8 65, 66, 68, 190

TIFF *siehe* Tagged Image File Format

TOGAF *siehe* The Open Group Architectural Framework

TOGAF ADM *siehe* TOGAF Architecture Development Method

TOGAF Architecture Development Method *siehe* The Open Group Architectural Framework

TOGAF EC *siehe* TOGAF Enterprise Continuum

TOGAF Enterprise Continuum *siehe* The Open Group Architectural Framework

TOGAF Foundation Architecture *siehe* The Open Group Architectural Framework

TOGAF III-RM *siehe* TOGAF Integrated Information Infrastructure Reference Model

TOGAF Integrated Information Infrastructure Reference Model *siehe* The Open Group Architectural Framework

TOGAF RB *siehe* TOGAF Resource Base

TOGAF Resource Base *siehe* The Open Group Architectural Framework

TOGAF SIB *siehe* TOGAF Standards Information Base

TOGAF Standards Information Base *siehe* The Open Group Architectural Framework

TOGAF Technical Reference Model *siehe* The Open Group Architectural Framework

TOGAF TRM *siehe* TOGAF Technical Reference Model

TP monitor *siehe* Transaction Processing Monitor

Trading Function *siehe* Referenzmodell für offene verteilte Informationsverarbeitung

Transaction Processing Monitor 62
 Transaktion 157–159, 195
 Abgleichs~ 158
 ~sausführung 162–163
 Kategorien von ~en 162–163, 168, 169
 Kooperations~ 158
 ~sstärke 162, 195
 einfache ~ 162, 168, 169
 standardisierte ~ 162, 168, 169
 Transaktionsausführung *siehe* Transaktion
 Transaktionsmodus 104
 Transaktionsstärke *siehe* Transaktion
 Transmission Control Protocol 34, 52
 Transparenz 22, 38
 Typ 22

U

Übermittlungsdomäne *siehe* Domäne
 UKL *siehe* Universitätsklinikum Leipzig
 UML *siehe* Unified Modeling Language
 Unified Modeling Language 21, 32, 45, 59
 Universitätsklinikum Leipzig 3, 130, 163
 use case *siehe* Anwendungsfall

V

Verfeinerung 22
 Verhalten 22
 Vermittlung 126–127
 ~stiefe 126–127
 ~sverbindung 126
 ~sverbindung 174, 175
 Versagen 22
 vertikale Fragmentierung 60
 Vertrag 22
 Vorgehens-Referenzmodell *siehe* Modell

W

Wartbarkeit 8
 Wireless LAN 34
 Wissen 11, 26
 Wrapper *siehe* Hülle

X

XNFS 52

Z

Zachman-Rahmenwerk 55, 56, 68, 75, 190
 Zugriffsdomäne *siehe* Domäne
 Zugriffsintegration *siehe* Integration

Stichwortverzeichnis